aws

Panduan Pengguna

Amazon Simple Storage Service

Versi API 2006-03-01

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Simple Storage Service: Panduan Pengguna

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

Apa itu Amazon S3? 1
Fitur-fitur Amazon S3 1
Kelas penyimpanan 1
Manajemen penyimpanan 2
Manajemen akses dan keamanan 3
Pemrosesan data 4
Pencatatan dan pemantauan penyimpanan 4
Analitik dan wawasan 5
Konsistensi kuat 5
Cara kerja Amazon S3 6
Bucket 7
Objek 9
Kunci 10
Penentuan Versi S3 10
ID Versi 10
Kebijakan bucket 10
Titik akses S3 11
Daftar kontrol akses (ACLs) 11
Wilayah 12
Model konsistensi data Amazon S3 13
Aplikasi yang bersamaan 14
Layanan terkait 15
Mengakses Amazon S3 16
AWS Management Console 16
AWS Command Line Interface 16
AWS SDKs 17
API REST Amazon S3 17
Membayar Amazon S3 18
Kepatuhan PCI DSS 18
Memulai 19
Pengaturan 19
Mendaftar untuk Akun AWS 20
Buat pengguna dengan akses administratif 20
Langkah 1: Buat bucket 22

Langkah 2: Unggah objek 28
Langkah 3: Mengunduh objek 29
Menggunakan konsol S3 30
Langkah 4: Salin objek 31
Langkah 5: Hapus objek dan bucket Anda 32
Menghapus objek 32
Mengosongkan bucket Anda 33
Menghapus bucket Anda 33
Langkah selanjutnya 34
Memahami kasus penggunaan umum 35
Mengontrol akses ke bucket dan objek Anda 35
Lindungi dan pantau penyimpanan Anda 36
Mengembangkan dengan Amazon S3 37
Belajar dari tutorial 38
Menjelajahi pelatihan dan dukungan 40
Menggunakan Amazon S3 dengan AWS CLI 40
Pengaturan 41
Langkah 1: Buat bucket Amazon S3 pertama Anda 41
Langkah 2: Unggah objek ke bucket Anda 48
Langkah 3: Mengunduh objek 49
Langkah 4: Salin objek ke folder 50
Langkah 5: Hapus objek dan bucket Anda 50
Langkah selanjutnya 53
Bekerja dengan ember tujuan umum 55
Ikhtisar ember tujuan umum 56
Ikhtisar ember tujuan umum 57
Pola ember tujuan umum umum 58
Izin 58
Mengelola akses publik ke ember tujuan umum 59
konfigurasi bucket tujuan umum 60
operasi bucket tujuan umum 63
pemantauan kinerja bucket tujuan umum 63
Pola ember umum 65
Pola bucket serba guna multi-penyewa 65
Bucket-per-use pola 66
Peraturan penamaan 67

Aturan penamaan bucket tujuan umum 68
Contoh nama bucket tujuan umum 69
Praktik terbaik 70
Membuat bucket yang menggunakan GUID dalam nama bucket 71
Kuota, batasan, dan batasan 73
Kuota ember 73
Objek dan batasan bucket 74
Peraturan penamaan bucket 74
Mengakses bucket 74
..... 74
Hosting virtual dari ember tujuan umum 77
Membuat ember tujuan umum 87
Pengaturan bucket tujuan umum 87
Melihat properti bucket 102
Mencantumkan bucket 104
Mengosongkan ember tujuan umum 110
Mengosongkan bucket tujuan umum dengan dikonfigurasi AWS CloudTrail 113
Menghapus bucket tujuan umum 114
Mountpoint untuk Amazon S3 120
Instalasi Mountpoint 121
Mengkonfigurasi dan menggunakan Mountpoint 126
Pemecahan Masalah Mountpoint 132
Browser Penyimpanan untuk Amazon S3 133
Menggunakan Browser Penyimpanan untuk S3 134
Instalasi Browser Penyimpanan untuk S3 134
Menyiapkan Browser Penyimpanan untuk S3 135
Mengkonfigurasi Browser Penyimpanan untuk S3 149
Pemecahan Masalah Browser Penyimpanan untuk S3 150
Mengonfigurasi Transfer Acceleration 151
Mengapa menggunakan Percepatan Transfer? 151
Persyaratan untuk menggunakan Transfer Acceleration 152
Memulai 153
Mengaktifkan Transfer Acceleration 155
Alat Perbandingan Kecepatan 163
Menggunakan Pembayaran Pemohon 163
Cara kerja Pembayaran Pemohon 165

Mengonfigurasi Pembayaran Pemohon ... 165
Mengambil konfigurasi requestPayment ... 167
Mengunduh objek dari ember Requester Pays ... 168
Bekerja dengan objek ... 170
Ikhtisar objek ... 171
Sub-sumber daya ... 172
Penamaan objek ... 173
Memilih nama kunci objek ... 174
Panduan penamaan kunci objek ... 175
Bekerja dengan metadata ... 178
Metadata objek yang ditentukan sistem ... 179
Metadata objek yang ditentukan pengguna ... 183
Mengedit metadata objek ... 185
Mempercepat penemuan data ... 189
Mengunggah Objek ... 229
Mengunggah objek ... 230
Mencegah mengunggah objek dengan nama kunci yang identik ... 243
Menggunakan unggahan multibagian ... 244
Membuat permintaan bersyarat ... 322
Cara mengambil atau menyalin objek berdasarkan metadata ... 322
Bagaimana mencegah penimpaan objek ... 325
Menyalin, memindahkan, dan mengganti nama objek ... 335
Untuk menyalin objek ... 339
Untuk memindahkan objek ... 349
Untuk mengganti nama objek ... 350
Mengunggah objek ... 352
Mengunduh objek ... 353
Mengunduh beberapa objek ... 354
Mengunduh bagian dari suatu objek ... 356
Mengunduh objek dari Akun AWS yang lain ... 357
Mengunduh objek yang diarsipkan ... 358
Mengunduh objek berdasarkan metadata ... 359
Memecahkan masalah pengunduhan objek ... 359
Memeriksa integritas objek di Amazon S3 ... 359
Menggunakan algoritma checksum yang didukung ... 360
Objek lengkap dan jenis checksum komposit ... 361

Menggunakan checksum objek lengkap untuk unggahan multipart 363
Menggunakan checksum tingkat suku cadang untuk unggahan multipart 364
Operasi Checksum 366
Menggunakan Konten- MD5 saat mengunggah objek 374
Menggunakan Konten- MD5 dan ETag untuk memverifikasi objek yang diunggah 374
Menggunakan checksum trailing 375
Menghapus objek 381
Praktik terbaik untuk dipertimbangkan sebelum menghapus objek 381
Menghapus objek dari bucket berkemampuan versi 382
Menghapus objek dari bucket yang ditangguhkan versi 383
Menghapus objek dari bucket yang tidak berversi 383
Menghapus objek dari bucket yang mengaktifkan MFA 383
Menghapus satu objek 384
Menghapus beberapa objek 396
Mengatur dan membuat daftar objek 399
Menggunakan prefiks 399
Membuat daftar objek 402
Menggunakan folder 405
Melihat properti objek 410
Mengkategorikan objek dengan tag 412
Menggunakan presigned URLs untuk mengunduh dan mengunggah objek 425
Siapa yang dapat membuat URL yang telah ditandatangani 426
Waktu kedaluwarsa untuk presigned URLs 427
Pembatasan kemampuan URL yang telah ditandatangani 428
Berbagi objek dengan presigned URLs 429
Mengunggah objek dengan presigned URLs 435
Mengubah objek 438
Membuat Titik Akses Objek Lambda 440
Menggunakan Titik Akses Lambda Objek Amazon S3 455
Pertimbangan keamanan 459
Menulis fungsi Lambda 466
Menggunakan fungsi AWS yang dibangun 498
Praktik terbaik dan pedoman untuk S3 Lambda Objek 500
Tutorial S3 Lambda Objek 502
Pemecahan Masalah Objek S3 Lambda 540
Melakukan operasi objek secara massal 541

Dasar-dasar Operasi Batch ... 541
Tutorial Operasi Batch S3 ... 543
Memberikan izin ... 543
Membuat pekerjaan ... 554
Operasi yang didukung ... 578
Mengelola tugas ... 622
Melacak status tugas dan laporan penyelesaian ... 627
Menggunakan tanda ... 643
Mengelola Kunci Objek dengan Operasi Batch ... 659
Tutorial: Video transcoding batch ... 684
Pemecahan Masalah Operasi Batch ... 725
Meminta data di tempat ... 728
Persyaratan dan batasan ... 728
Membuat permintaan ... 729
Kesalahan ... 731
Contoh S3 Select ... 731
Referensi SQL ... 735
Bekerja dengan bucket direktori ... 776
Nama bucket direktori ... 777
Direktori ... 777
Nama kunci ... 777
Manajemen akses ... 778
Kuota ember direktori ... 778
Membuat dan menggunakan bucket direktori ... 779
Kasus penggunaan untuk ember direktori ... 779
Beban kerja kinerja tinggi ... 780
Beban kerja residensi data ... 836
Perbedaan untuk ember direktori ... 853
Perbedaan untuk ember direktori ... 853
Operasi API didukung untuk bucket direktori ... 855
Fitur Amazon S3 tidak didukung oleh bucket direktori ... 855
Jaringan untuk ember direktori ... 856
Titik akhir ... 857
Mengkonfigurasi titik akhir gateway VPC ... 858
Aturan penamaan bucket direktori ... 858
Melihat properti ... 859

Mengelola kebijakan bucket 860
Menambahkan kebijakan bucket 860
Melihat kebijakan bucket 863
Menghapus kebijakan bucket 864
Mengosongkan bucket direktori 865
Menghapus bucket direktori 866
Mendaftar bucket direktori 869
Menentukan apakah Anda dapat mengakses bucket 872
Bekerja dengan objek di dalam bucket direktori 873
Mengimpor objek ke dalam bucket direktori 873
Bekerja dengan Siklus Hidup S3 875
Menggunakan Operasi Batch dengan S3 Express One Zone 888
Menambahkan data ke objek 891
Mengganti nama objek 894
Mengunggah Objek 900
Menyalin objek 933
Menghapus objek 939
Mengunduh objek 944
Menghasilkan presigned URLs untuk berbagi objek 946
Mengambil metadata objek 946
Membuat daftar objek 948
Keamanan untuk ember direktori 948
Perlindungan data dan enkripsi 949
Mengautentikasi dan mengotorisasi permintaan 972
Praktik terbaik keamanan 982
Bekerja dengan titik akses untuk ember direktori 986
Aturan penamaan, batasan, dan batasan 987
Mereferensikan titik akses untuk bucket direktori 989
Operasi objek titik akses 989
Mengonfigurasi kebijakan IAM 990
Pencatatan dan pemantauan 996
Membuat titik akses untuk ember direktori 997
Mengelola titik akses 1001
Masuk dengan AWS CloudTrail untuk ember direktori 1014
CloudTrail acara manajemen untuk ember direktori 1014
CloudTrail peristiwa data untuk ember direktori 1015

... 1016
Mengoptimalkan kinerja bucket direktori ... 1023
Gunakan autentikasi berbasis sesi ... 1023
Praktik terbaik checksum tambahan S3 ... 1024
Gunakan versi terbaru dari AWS SDKs pustaka runtime umum ... 1024
Berkembang dengan ember direktori ... 1024
Titik akhir regional dan Zonal untuk bucket direktori ... 1025
Bekerja dengan bucket direktori dengan menggunakan konsol S3,, dan AWS CLIAWS SDKs ... 1025
Operasi API bucket direktori ... 1027
Menandai ember direktori ... 1029
Bagaimana tag bekerja ... 1029
Cara umum untuk menggunakan tag ... 1031
Bekerja dengan tag ... 1036
Membuat ember direktori dengan tag ... 1036
Menambahkan tag ke bucket direktori ... 1041
Melihat tag bucket direktori ... 1044
Menghapus tag dari bucket direktori ... 1047
Bekerja dengan Tabel Amazon S3 dan ember meja ... 1050
Fitur Tabel S3 ... 1050
Layanan terkait ... 1052
Tutorial: Memulai dengan Tabel S3 ... 1053
Langkah 1: Buat keranjang meja dan integrasikan dengan layanan AWS analitik ... 1053
Langkah 2: Buat namespace tabel dan tabel ... 1055
(Opsional) Langkah 3: Berikan izin Lake Formation di meja Anda ... 1057
Langkah 4: Kueri data dengan SQL di Athena ... 1059
Ember meja ... 1060
Aturan penamaan ember meja ... 1061
Buat ember meja ... 1063
Hapus ember meja ... 1066
Melihat detail ember meja ... 1066
Mengelola kebijakan ... 1066
Pemeliharaan Tabel S3 ... 1068
Status pekerjaan pemeliharaan Tabel S3 ... 1068
Perawatan ember meja ... 1070
Pemeliharaan meja ... 1072

Pertimbangan dan batasan 1077
Namespace 1079
Buat namespace 1080
Hapus namespace 1082
Tabel 1082
Membuat tabel 1084
Menghapus tabel 1089
Mengelola kebijakan 1090
Mengakses tabel 1092
Mengakses tabel melalui integrasi Amazon SageMaker Lakehouse 1092
Mengakses tabel secara langsung 1093
Menggunakan Tabel S3 dengan layanan AWS analitik 1095
Mengakses tabel menggunakan AWS Glue Iceberg REST titik akhir 1107
Mengakses tabel menggunakan Tabel Amazon S3 Iceberg REST titik akhir 1112
Mengakses tabel dengan katalog klien 1118
Amazon Athena 1121
Amazon Redshift 1123
Amazon EMR 1123
QuickSight 1127
Amazon Data Firehose 1130
AWS Glue ETL 1135
Wilayah AWS, titik akhir, dan kuota 1146
Tabel S3 Wilayah AWS dan titik akhir 1147
Kuota Tabel S3 1152
Keamanan untuk Tabel S3 1152
Enkripsi 1154
Manajemen akses 1168
Konektivitas VPC 1196
Pembatasan dan batasan 1200
Logging dengan AWS CloudTrail untuk Tabel S3 1201
CloudTrail acara manajemen untuk Tabel S3 1201
CloudTrail peristiwa data untuk Tabel S3 1202
CloudTrail contoh log 1203
Kontrol akses 1207
Sumber daya S3 1208
Identitas 1213

Pemilik ember atau sumber daya 1215
Alat manajemen akses 1215
Tindakan 1222
Kasus penggunaan manajemen akses 1223
Pemecahan masalah manajemen akses 1230
Identity and Access Management (IAM) 1231
Audiens 1232
Mengautentikasi dengan identitas 1233
Mengelola akses menggunakan kebijakan 1237
Bagaimana Amazon S3 bekerja dengan IAM 1239
Meminta otorisasi 1267
Izin yang diperlukan untuk operasi API S3 1276
Kebijakan dan Izin 1316
Kebijakan bucket 1321
Kebijakan berbasis identitas 1369
Penelusuran menggunakan kebijakan 1406
Menggunakan Peran Terkait Layanan untuk Lensa Penyimpanan Amazon S3 1448
Memecahkan masalah identitas dan akses Amazon S3 1452
AWS kebijakan terkelola 1477
Bekerja dengan titik akses 1479
Aturan penamaan, batasan, dan batasan 1481
Merujuk titik akses 1484
Kompatibilitas titik akses 1487
Mengonfigurasi kebijakan IAM 1490
Pemantauan dan pencatatan 1498
Membuat titik akses 1500
Mengelola titik akses 1508
Menggunakan titik akses 1513
Mengelola akses dengan S3 Access Grants 1525
Konsep S3 Access Grants 1527
S3 Access Grants dan identitas direktori perusahaan 1532
Memulai S3 Access Grants 1540
Bekerja dengan instans S3 Access Grants 1542
Bekerja dengan lokasi S3 Access Grants 1555
Bekerja dengan hibah di S3 Access Grants 1578
Mendapatkan data S3 menggunakan hibah akses 1592

Akses S3 memberikan akses lintas akun 1608
Menggunakan AWS tag dengan S3 Access Grants 1621
Batasan S3 Access Grants 1623
Integrasi S3 Access Grants 1626
Mengelola akses dengan ACLs 1627
Gambaran umum ACL 1628
Mengkonfigurasi ACLs 1649
Contoh kebijakan 1667
Memblokir akses publik 1671
Pengaturan blokir akses publik 1673
Melakukan operasi akses publik blok pada titik akses 1677
Arti “publik” 1678
Menggunakan IAM Access Analyzer untuk S3 untuk meninjau bucket publik 1681
Izin 1682
Mengonfigurasi blokir akses publik 1683
Mengonfigurasi pengaturan akun 1683
Mengonfigurasi pengaturan bucket dan titik akses 1686
Meninjau akses bucket 1689
Informasi apa yang disediakan IAM Access Analyzer untuk S3? 1691
Mengaktifkan IAM Access Analyzer untuk S3 1692
Memblokir semua akses publik 1692
Meninjau dan mengubah akses bucket 1693
Mengarsipkan temuan bucket 1695
Mengaktifkan temuan bucket yang diarsipkan 1695
Melihat detail temuan 1696
Mengunduh IAM Access Analyzer untuk laporan S3 1696
Memverifikasi pemilik bucket 1696
Kapan harus menggunakan ketentuan pemilik bucket 1697
Memverifikasi pemilik bucket 1698
Contoh 1699
Pembatasan dan batasan 1701
Mengontrol kepemilikan objek 1702
Pengaturan Kepemilikan Objek 1704
Perubahan diperkenalkan dengan menonaktifkan ACLs 1706
Prasyarat untuk menonaktifkan ACLs 1708
Izin Kepemilikan Objek 1708

Menonaktifkan ACLs untuk semua bucket baru 1708
Replikasi dan Kepemilikan Object 1709
Mengatur Kepemilikan Objek 1709
Prasyarat untuk menonaktifkan ACLs 1710
Buat bucket 1726
Mengatur Kepemilikan Objek 1735
Melihat pengaturan Kepemilikan Objek 1738
Menonaktifkan ACLs untuk semua bucket baru 1740
Pemecahan Masalah 1743
Keamanan 1746
Praktik terbaik keamanan 1748
Praktik terbaik keamanan Amazon S3 1748
Praktik Terbaik Pemantauan dan Audit Amazon S3 1754
Memantau keamanan data 1758
Perlindungan data 1762
Enkripsi data 1763
enkripsi di sisi server 1765
Menggunakan enkripsi di sisi klien 1866
Privasi lalu lintas jaringan internet 1867
Lalu lintas antara layanan dan aplikasi dan klien lokal 1867
Lalu lintas antar AWS sumber daya di Wilayah yang sama 1868
AWS PrivateLink untuk Amazon S3 1868
Validasi kepatuhan 1886
Ketahanan 1888
Enkripsi cadangan 1890
Keamanan infrastruktur 1891
Konfigurasi dan analisis kerentanan 1892
Manajemen akses 1892
Perlindungan data 1895
Mereplikasi objek di dalam dan di seluruh Wilayah 1897
Mengapa menggunakan replikasi? 1898
Kapan menggunakan Replikasi Lintas-Wilayah 1900
Kapan menggunakan Replikasi Lintas-Wilayah 1900
Kapan menggunakan replikasi dua arah (replikasi bidireksional) 1901
Kapan menggunakan Replikasi Batch S3 1901
Persyaratan beban kerja dan replikasi langsung 1902

Apa yang direplikasi? 1903
Persyaratan dan pertimbangan untuk replikasi 1906
Menyiapkan replikasi langsung 1910
Mengelola atau menjeda replikasi langsung 2004
Mereplikasi objek yang ada 2005
Mengatasi masalah replikasi 2020
Memantau kemajuan dan mendapatkan status 2029
Mengelola lalu lintas multi-wilayah 2052
Membuat Titik Akses Multi-Wilayah 2054
Mengonfigurasi Titik Akses Multi-Wilayah 2063
Menggunakan Titik Akses Multi-Wilayah 2067
Mempertahankan beberapa versi objek 2119
Bucket tanpa versi, dengan dukungan Penentuan Versi, dan dengan Penentuan Versi ditangguhkan 2119
Menggunakan Penentuan Versi S3 dengan Siklus Hidup S3 2120
Penentuan Versi S3 2121
Mengaktifkan Penentuan Versi pada bucket 2125
Mengonfigurasi penghapusan MFA 2133
Bekerja dengan objek dengan Penentuan Versi yang diaktifkan 2135
Bekerja dengan objek dengan Penentuan Versi ditangguhkan 2166
Pemecahan masalah pembuatan versi 2170
Mengunci objek 2175
Cara kerja Kunci Objek S3 2176
Pertimbangan Kunci Objek 2180
Mengonfigurasi Kunci Objek 2186
Mencadangkan data Anda 2196
Optimalisasi biaya 2198
Pelaporan penagihan dan penggunaan 2199
Menggunakan tanda alokasi biaya 2200
Laporan penagihan 2201
Laporan penggunaan 2204
Memahami laporan penagihan dan penggunaan 2207
Penagihan untuk respons kesalahan Amazon S3 2234
Memahami dan mengelola kelas penyimpanan 2257
Objek yang sering diakses 2258

Secara otomatis mengoptimalkan data dengan pola akses yang berubah atau tidak diketahui 2259
Objek yang jarang diakses 2261
Objek yang jarang diakses 2263
Amazon S3 on Outposts 2264
Membandingkan kelas penyimpanan 2264
Mengatur kelas penyimpanan objek 2266
Analisis Kelas Penyimpanan 2270
Mengelola biaya penyimpanan dengan Amazon S3 Intelligent-Tiering 2278
Kelas penyimpanan Amazon S3 Glacier 2291
Bekerja dengan objek yang diarsipkan 2297
Mengelola siklus hidup 2311
Mengelola siklus hidup objek yang lengkap 2312
Transisi objek 2314
Mengakhiri objek 2322
Menetapkan konfigurasi siklus hidup 2325
Menggunakan konfigurasi bucket lainnya 2345
Mengonfigurasi pemberitahuan acara Siklus Hidup S3 2347
Elemen konfigurasi Siklus Hidup 2349
Konflik konfigurasi siklus hidup 2368
Contoh konfigurasi Siklus Hidup S3 2373
Memecahkan masalah siklus hidup 2389
Pencatatan dan pemantauan 2397
Alat-alat pemantauan 2401
Alat otomatis 2401
Alat manual 2401
Opsi pencatatan 2402
Logging dengan CloudTrail 2405
Menggunakan CloudTrail log dengan log akses server Amazon S3 dan Log CloudWatch .. 2407
CloudTrail melacak dengan panggilan API SOAP Amazon S3 2408
CloudTrail acara 2409
Contoh file log 2421
Mengaktifkan CloudTrail 2425
Mengidentifikasi permintaan S3 2428
Pencatatan akses server 2436
Bagaimana cara mengaktifkan log pengiriman? 2436

Format kunci objek log ... 2439
Bagaimana log dikirimkan? ... 2440
Pengiriman log server dengan upaya terbaik ... 2441
Perubahan status pencatatan log bucket memerlukan waktu ... 2441
Mengaktifkan pencatatan akses server ... 2442
Format log ... 2465
Menghapus file log ... 2480
Mengidentifikasi permintaan S3 ... 2480
Memecahkan masalah pencatatan akses server ... 2489
Memantau metrik dengan CloudWatch ... 2491
Metrik dan dimensi ... 2494
Mengakses metrik CloudWatch ... 2513
CloudWatch konfigurasi metrik ... 2514
Notifikasi Peristiwa Amazon S3 ... 2523
Ikhtisar ... 2524
Jenis dan tujuan notifikasi peristiwa ... 2525
Menggunakan SQS, SNS, dan Lambda ... 2533
Menggunakan EventBridge ... 2563
Memvisualisasikan aktivitas penyimpanan dan penggunaan ... 2573
Metrik dan fitur Lensa Penyimpanan S3 ... 2574
Memahami Lensa Penyimpanan S3 ... 2576
Glosarium metrik ... 2587
Mengatur izin ... 2622
Bekerja dengan Lensa Penyimpanan S3 ... 2626
Melihat metrik penyimpanan ... 2666
Bekerja dengan Organisasi ... 2724
Bekerja dengan grup Storage Lens ... 2734
Mengkatalogkan dan menganalisis data Anda ... 2780
Bucket inventaris Amazon S3 ... 2781
Daftar inventaris ... 2782
Mengonfigurasi Inventaris Amazon S3 ... 2787
Menemukan inventaris ... 2797
Menyiapkan notifikasi peristiwa untuk penyelesaian inventaris ... 2801
Membuat kueri inventaris dengan Athena ... 2802
Mengonversi string ID versi kosong menjadi string null ... 2808
Menggunakan bidang ACL Objek ... 2811

Mengoptimalkan performa 2814
Pedoman kinerja untuk Amazon S3 2815
Ukur performa 2816
Skala horisontal 2817
Gunakan pengambilan rentang byte 2817
Permintaan coba lagi 2817
Gabungkan Amazon S3 dan Amazon EC2 di Wilayah yang sama 2818
Gunakan Transfer Acceleration untuk meminimalkan latensi 2818
Gunakan yang terbaru AWS SDKs 2818
Pola desain kinerja untuk Amazon S3 2819
Caching konten yang sering diakses 2819
Batas waktu dan percobaan ulang untuk aplikasi yang sensitif terhadap latensi 2820
Penskalaan horizontal dan paralelisasi permintaan 2821
Mempercepat transfer data yang berbeda secara geografis 2823
Hosting situs web statis 2824
Titik akhir situs web 2825
Contoh titik akhir situs web 2826
Menambahkan CNAME DNS 2827
Menggunakan domain khusus dengan Route 53 2827
Perbedaan utama antara titik akhir situs web dan titik akhir API REST 2828
Mengaktifkan hosting situs web 2829
Mengonfigurasi dokumen indeks 2834
Indeks dokumen dan folder 2834
Mengonfigurasi dokumen indeks 2835
Mengonfigurasi dokumen kesalahan khusus 2837
Kode respons HTTP Amazon S3 2838
Mengonfigurasi dokumen kesalahan khusus 2840
Mengatur izin untuk akses situs web 2841
Langkat 1: Edit pengaturan Blokir Akses Publik S3 2842
Langkah 2: Menambahkan kebijakan bucket 2844
Daftar kontrol akses objek 2846
Mencatat lalu lintas web 2847
Mengonfigurasi pengalihan 2848
Permintaan pengalihan ke host lain 2848
Konfigurasikan aturan pengalihan 2849
Mengalihkan permintaan untuk objek 2857

Menggunakan CORS 2860
Berbagi sumber daya lintas asal: Skenario kasus penggunaan 2860
Bagaimana Amazon S3 mengevaluasi konfigurasi CORS pada bucket? 2861
Bagaimana Titik Akses Lambda Objek mendukung CORS 2861
Elemen konfigurasi CORS 2862
Mengonfigurasi CORS 2867
Menguji CORS 2876
Penyelesaian masalah CORS 2878
Tutorial situs web statis 2883
Hosting streaming video 2885
Mengonfigurasi situs web statis 2905
Mengonfigurasi situs web statis menggunakan domain kustom 2914
Menyebarkan situs web statis untuk Amplify dari Amazon S3 2939
Kuota 2942
Kuota meningkat 2942
Referensi 2943
Riwayat dokumen 2944
Pembaruan sebelumnya 2995
..... mmmxxiii

Apa itu Amazon S3?

Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek yang menawarkan skalabilitas, ketersediaan data, keamanan, dan kinerja terdepan di industri. Pelanggan dari semua ukuran dan industri dapat menggunakan Amazon S3 untuk menyimpan dan melindungi sejumlah data untuk berbagai kasus penggunaan, seperti danau data, situs web, aplikasi seluler, pencadangan dan pemulihan, arsip, aplikasi perusahaan, perangkat IoT, dan analitik big data. Amazon S3 menyediakan fitur manajemen sehingga Anda dapat mengoptimalkan, mengatur, dan mengonfigurasi akses ke data Anda untuk memenuhi persyaratan bisnis, organisasi, dan kepatuhan spesifik Anda.

> **Note**
>
> Untuk informasi selengkapnya tentang menggunakan kelas penyimpanan Amazon S3 Express One Zone dengan bucket direktori, lihat S3 Express One Zone dan Bekerja dengan bucket direktori.

Topik

- Fitur-fitur Amazon S3
- Cara kerja Amazon S3
- Model konsistensi data Amazon S3
- Layanan terkait
- Mengakses Amazon S3
- Membayar Amazon S3
- Kepatuhan PCI DSS

Fitur-fitur Amazon S3

Kelas penyimpanan

Amazon S3 menawarkan serangkaian kelas penyimpanan yang dirancang untuk berbagai kasus penggunaan. Misalnya, Anda dapat menyimpan data produksi mission-critical di S3 Standard atau S3 Express One Zone untuk akses yang sering, menghemat biaya dengan menyimpan data yang jarang diakses di S3 Standard-IA atau S3 One Zone-IA, dan mengarsipkan data dengan biaya terendah di S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval, dan S3 Glacier Deep Archive.

Amazon S3 Express One Zone adalah kelas penyimpanan Amazon S3 tunggal berkinerja tinggi yang dibuat khusus untuk menghadirkan akses data milidetik satu digit yang konsisten untuk aplikasi Anda yang paling sensitif terhadap latensi. S3 Express One Zone adalah kelas penyimpanan objek cloud latensi terendah yang tersedia saat ini, dengan kecepatan akses data hingga 10x lebih cepat dan dengan biaya permintaan 50 persen lebih rendah dari Standar S3. S3 Express One Zone adalah kelas penyimpanan S3 pertama di mana Anda dapat memilih satu Zona Ketersediaan dengan opsi untuk menempatkan bersama penyimpanan objek dengan sumber daya komputasi Anda, yang memberikan kecepatan akses setinggi mungkin. Selain itu, untuk lebih meningkatkan kecepatan akses dan mendukung ratusan ribu permintaan per detik, data disimpan dalam jenis bucket baru: bucket direktori Amazon S3. Untuk informasi selengkapnya, lihat S3 Express One Zone dan Bekerja dengan bucket direktori.

Anda dapat menyimpan data dengan pola akses yang berubah atau tidak dikenal di S3 Intelligent-Tiering, yang mengoptimalkan biaya penyimpanan dengan memindahkan data Anda secara otomatis di antara empat tingkatan akses saat pola akses Anda berubah. Keempat jenjang akses ini mencakup dua jenjang akses latensi rendah yang dioptimalkan untuk akses rutin dan tidak rutin, serta dua jenjang akses arsip pilihan yang dirancang untuk akses asinkron untuk data yang jarang diakses.

Untuk informasi selengkapnya, lihat Memahami dan mengelola kelas penyimpanan Amazon S3.

Manajemen penyimpanan

Amazon S3 memiliki fitur manajemen penyimpanan yang dapat Anda gunakan untuk mengelola biaya, memenuhi persyaratan peraturan, mengurangi latensi, dan menyimpan beberapa salinan data Anda yang berbeda untuk persyaratan kepatuhan.

- Siklus Hidup S3–Konfigurasikan konfigurasi siklus hidup untuk mengelola objek Anda dan menyimpannya secara hemat di sepanjang siklus hidupnya. Anda dapat mentransisikan objek ke kelas penyimpanan S3 lainnya atau objek kedaluwarsa yang mencapai akhir masa pakainya.
- Kunci Objek S3–Cegah objek Amazon S3 dihapus atau ditimpa selama jangka waktu tertentu atau tanpa batas waktu. Anda dapat menggunakan Object Lock untuk membantu memenuhi persyaratan peraturan yang memerlukan penyimpanan write-once-read-many(WORM) atau hanya menambahkan lapisan perlindungan lain terhadap perubahan dan penghapusan objek.
- Replikasi S3 - Replikasi objek dan metadata masing-masing dan tag objek ke satu atau beberapa bucket tujuan yang sama atau berbeda Wilayah AWS untuk mengurangi latensi, kepatuhan, keamanan, dan kasus penggunaan lainnya.
- Operasi Batch S3–Kelola miliaran objek dalam skala besar dengan satu permintaan API S3 atau beberapa klik di konsol Amazon S3. Anda dapat menggunakan Operasi Batch untuk melakukan

operasi seperti Salin, Memanggil fungsi AWS Lambda, dan Pulihkan pada jutaan atau miliaran objek.

Manajemen akses dan keamanan

Amazon S3 menyediakan fitur untuk mengaudit dan mengelola akses ke bucket dan objek Anda. Secara default, bucket S3 dan objek di dalamnya bersifat pribadi. Anda hanya memiliki akses ke sumber daya S3 yang Anda buat. Untuk memberikan izin sumber daya terperinci yang mendukung kasus penggunaan spesifik Anda atau untuk mengaudit izin sumber daya Amazon S3, Anda dapat menggunakan fitur berikut.

- Blokir Akses Publik S3 – Memblokir akses publik ke bucket dan objek S3. Secara default, pengaturan Blokir Akses Publik diaktifkan di tingkat bucket. Kami menyarankan Anda untuk tetap mengaktifkan semua pengaturan Blokir Akses Publik kecuali Anda tahu bahwa Anda perlu mematikan satu atau beberapa pengaturan untuk kasus penggunaan spesifik Anda. Untuk informasi selengkapnya, lihat Mengonfigurasi pengaturan blokir akses publik untuk bucket S3 Anda.
- AWS Identity and Access Management (IAM) - IAM adalah layanan web yang membantu Anda mengontrol akses ke AWS sumber daya dengan aman, termasuk sumber daya Amazon S3 Anda. Dengan IAM, Anda dapat mengelola izin secara terpusat yang mengontrol AWS sumber daya mana yang dapat diakses pengguna. Anda menggunakan IAM untuk mengontrol siapa yang diautentikasi (masuk) dan diotorisasi (memiliki izin) untuk menggunakan sumber daya.
- Kebijakan bucket–Gunakan bahasa kebijakan berbasis IAM untuk mengonfigurasi izin berbasis sumber daya untuk bucket S3 dan objek di dalamnya.
- Titik akses Amazon S3–Konfigurasi titik akhir jaringan bernama dengan kebijakan akses khusus untuk mengelola akses data dalam skala besar untuk set data bersama di Amazon S3.
- Daftar kontrol akses (ACLs) — Berikan izin baca dan tulis untuk masing-masing bucket dan objek kepada pengguna yang berwenang. Sebagai aturan umum, sebaiknya gunakan kebijakan berbasis sumber daya S3 (kebijakan bucket dan kebijakan jalur akses) atau kebijakan pengguna IAM untuk kontrol akses, bukan. ACLs Kebijakan adalah opsi kontrol akses yang disederhanakan dan lebih fleksibel. Dengan kebijakan bucket dan kebijakan titik akses, Anda dapat menentukan aturan yang berlaku secara luas di semua permintaan ke sumber daya Amazon S3 Anda. Untuk informasi selengkapnya tentang kasus tertentu saat Anda menggunakan, ACLs bukan kebijakan berbasis sumber daya atau kebijakan pengguna IAM, lihat. Mengelola akses dengan ACLs
- Kepemilikan Objek S3–Ambil kepemilikan setiap objek di bucket Anda, sederhanakan manajemen akses untuk data yang disimpan di Amazon S3. S3 Object Ownership adalah pengaturan tingkat

ember Amazon S3 yang dapat Anda gunakan untuk menonaktifkan atau mengaktifkan. ACLs Secara default, ACLs dinonaktifkan. Dengan ACLs dinonaktifkan, pemilik bucket memiliki semua objek di bucket dan mengelola akses ke data secara eksklusif dengan menggunakan kebijakan manajemen akses.

- IAM Access Analyzer for S3–Mengevaluasi dan memantau kebijakan akses bucket S3 Anda, memastikan bahwa kebijakan hanya menyediakan akses yang dimaksudkan ke sumber daya S3 Anda.

Pemrosesan data

Untuk mengubah data dan memicu alur kerja untuk mengotomatiskan berbagai aktivitas pemrosesan lainnya dalam skala besar, Anda dapat menggunakan fitur berikut.

- Lambda Objek S3–Tambahkan kode Anda sendiri ke permintaan S3 GET, HEAD, dan LIST untuk memodifikasi dan memproses data saat dikembalikan ke aplikasi. Filter baris, ubah ukuran gambar secara dinamis, edit data rahasia, dan banyak lagi.
- Pemberitahuan peristiwa - Memicu alur kerja yang menggunakan Amazon Simple Notification Service (Amazon SNS), Amazon Simple Queue Service (Amazon SQS), AWS Lambda dan saat perubahan dilakukan pada sumber daya S3 Anda.

Pencatatan dan pemantauan penyimpanan

Amazon S3 menyediakan alat pencatatan dan pemantauan yang dapat Anda gunakan untuk memantau dan mengontrol penggunaan sumber daya Amazon S3. Untuk informasi selengkapnya, lihat Alat pemantauan.

Alat-alat pemantauan otomatis

- CloudWatchMetrik Amazon untuk Amazon S3 — Lacak kesehatan operasional sumber daya S3 Anda dan konfigurasikan peringatan penagihan saat perkiraan biaya mencapai ambang batas yang ditentukan pengguna.
- AWS CloudTrail— Rekam tindakan yang diambil oleh pengguna, peran, atau Layanan AWS di Amazon S3. CloudTrail log memberi Anda pelacakan API terperinci untuk operasi tingkat ember dan tingkat objek S3.

Alat-alat pemantauan manual

- Pencatatan akses server–menyediakan catatan terperinci untuk permintaan yang dilakukan ke bucket Anda. Anda dapat menggunakan log akses server untuk banyak kasus penggunaan, seperti melakukan keamanan dan audit akses, mempelajari basis pelanggan Anda, dan memahami tagihan Amazon S3.
- AWS Trusted Advisor - Evaluasi akun Anda dengan menggunakan pemeriksaan praktik AWS terbaik untuk mengidentifikasi cara mengoptimalkan AWS infrastruktur Anda, meningkatkan keamanan dan kinerja, mengurangi biaya, dan memantau kuota layanan. Anda kemudian dapat mengikuti rekomendasi untuk mengoptimalkan layanan dan sumber daya Anda.

Analitik dan wawasan

Amazon S3 menawarkan fitur untuk membantu Anda mendapatkan visibilitas ke dalam penggunaan penyimpanan, yang memberdayakan Anda untuk lebih memahami, menganalisis, dan mengoptimalkan penyimpanan Anda dalam skala besar.

- Lensa Penyimpanan Amazon S3–Memahami, menganalisis, dan mengoptimalkan penyimpanan Anda. S3 Storage Lens menyediakan 60+ metrik penggunaan dan aktivitas serta dasbor interaktif untuk mengumpulkan data untuk seluruh organisasi, akun tertentu, bucket, atau awalan Anda. Wilayah AWS
- Analisis Kelas Penyimpanan–Analisis pola akses penyimpanan untuk memutuskan kapan saatnya memindahkan data ke kelas penyimpanan yang lebih hemat biaya.
- Inventaris S3 dengan laporan Inventaris–Audit dan laporkan objek dan metadata terkait serta konfigurasikan fitur Amazon S3 lainnya untuk mengambil tindakan dalam laporan Inventaris. Misalnya, Anda dapat melaporkan replikasi dan status enkripsi objek Anda. Untuk daftar semua metadata yang tersedia untuk setiap objek dalam laporan Inventaris, lihat daftar Inventaris Amazon S3.

Konsistensi kuat

Amazon S3 memberikan read-after-write konsistensi yang kuat untuk permintaan PUT dan DELETE objek di bucket Amazon S3 Anda secara keseluruhan. Wilayah AWS Perilaku ini berlaku untuk penulisan objek baru dan permintaan PUT yang menimpa objek yang sudah ada dan permintaan DELETE. Selain itu, operasi baca di Amazon S3 Select, daftar kontrol akses Amazon S3 ACLs

(), Tag Objek Amazon S3, dan metadata objek (misalnya, objek HEAD) sangat konsisten. Untuk informasi selengkapnya, lihat Model konsistensi data Amazon S3.

Cara kerja Amazon S3

Amazon S3 adalah layanan penyimpanan objek yang menyimpan data sebagai objek, data hierarkis, atau data tabular dalam bucket. Objek adalah file data dan metadata apa pun yang mendeskripsikan file tersebut. Bucket adalah kontainer untuk objek.

Untuk menyimpan data Anda di Amazon S3, Anda terlebih dahulu membuat bucket dan menentukan nama bucket dan Wilayah AWS. Kemudian, Anda mengunggah data Anda ke bucket itu sebagai objek di Amazon S3. Setiap objek memiliki kunci (atau nama kunci), yang merupakan pengidentifikasi unik untuk objek di dalam bucket.

S3 menyediakan fitur yang dapat Anda konfigurasi untuk mendukung kasus penggunaan spesifik Anda. Sebaga contoh, Anda dapat menggunakan Penentuan Versi S3 untuk menyimpan beberapa versi dari sebuah objek di dalam satu bucket sehingga Anda dapat memulihkan objek yang tidak sengaja terhapus atau tertimpa.

Bucket dan objek di dalamnya bersifat pribadi dan hanya dapat diakses jika Anda secara eksplisit memberikan izin akses. Anda dapat menggunakan kebijakan bucket, kebijakan AWS Identity and Access Management (IAM), daftar kontrol akses (ACLs), dan Titik Akses S3 untuk mengelola akses.

Topik

- Bucket
- Objek
- Kunci
- Penentuan Versi S3
- ID Versi
- Kebijakan bucket
- Titik akses S3
- Daftar kontrol akses (ACLs)
- Wilayah

Bucket

Amazon S3 mendukung tiga jenis ember — bucket tujuan umum, bucket direktori, dan bucket meja. Setiap jenis bucket menyediakan serangkaian fitur unik untuk kasus penggunaan yang berbeda.

Bucket tujuan umum — Bucket tujuan umum direkomendasikan untuk sebagian besar kasus penggunaan dan pola akses dan merupakan tipe bucket S3 asli. Bucket tujuan umum adalah wadah untuk objek yang disimpan di Amazon S3, dan Anda dapat menyimpan sejumlah objek dalam ember dan di semua kelas penyimpanan (kecuali untuk S3 Express One Zone), sehingga Anda dapat menyimpan objek secara berlebihan di beberapa Availability Zone. Untuk informasi selengkapnya, lihat Membuat, mengonfigurasi, dan bekerja dengan bucket tujuan umum Amazon S3.

> **Note**
>
> Secara default, semua bucket tujuan umum bersifat pribadi. Namun, Anda dapat memberikan akses publik ke ember tujuan umum. Anda dapat mengontrol akses ke bucket tujuan umum di tingkat bucket, awalan (folder), atau tag objek. Untuk informasi selengkapnya, lihat Kontrol akses di Amazon S3.

Bucket direktori — Direkomendasikan untuk kasus penggunaan latensi rendah dan kasus penggunaan residensi data. Secara default, Anda dapat membuat hingga 100 bucket direktori di direktori Anda Akun AWS, tanpa batasan jumlah objek yang dapat Anda simpan dalam ember direktori. Bucket direktori mengatur objek ke dalam direktori hierarkis (awalan) alih-alih struktur penyimpanan datar dari bucket tujuan umum. Jenis bucket ini tidak memiliki batasan awalan dan direktori individual dapat menskalakan secara horizontal. Untuk informasi selengkapnya, lihat Bekerja dengan bucket direktori.

- Untuk kasus penggunaan latensi rendah, Anda dapat membuat bucket direktori dalam satu AWS Availability Zone untuk menyimpan data. Bucket direktori di Availability Zones mendukung kelas penyimpanan S3 Express One Zone. Dengan S3 Express One Zone, data Anda disimpan secara berlebihan di beberapa perangkat dalam satu Zona Ketersediaan. Kelas penyimpanan S3 Express One Zone direkomendasikan jika aplikasi Anda sensitif terhadap kinerja dan mendapat manfaat dari milidetik PUT dan latensi satu digit. GET Untuk mempelajari selengkapnya tentang membuat bucket direktori di Availability Zone, lihat Beban kerja berkinerja tinggi.
- Untuk kasus penggunaan residensi data, Anda dapat membuat bucket direktori dalam satu Zona Lokal AWS Khusus (DLZ) untuk menyimpan data. Di Zona Lokal Khusus, Anda dapat membuat bucket S3 direktori untuk menyimpan data dalam perimeter data tertentu, yang membantu

mendukung kasus penggunaan residensi dan isolasi data Anda. Bucket direktori di Local Zones mendukung kelas penyimpanan S3 One Zone-Infrequent Access (S3 One Zone-IA; Z-IA). Untuk mempelajari selengkapnya tentang membuat bucket direktori di Local Zones, lihat Beban kerja residensi data.

> **Note**
>
> Bucket direktori memiliki semua akses publik dinonaktifkan secara default. Perilaku ini tidak dapat diubah. Anda tidak dapat memberikan akses ke objek yang disimpan dalam bucket direktori. Anda hanya dapat memberikan akses ke bucket direktori Anda. Untuk informasi selengkapnya, lihat Mengautentikasi dan mengotorisasi permintaan.

Bucket tabel — Direkomendasikan untuk menyimpan data tabular, seperti transaksi pembelian harian, data sensor streaming, atau tayangan iklan. Data tabular mewakili data dalam kolom dan baris, seperti dalam tabel database. Bucket meja menyediakan S3 penyimpanan yang dioptimalkan untuk beban kerja analitik dan pembelajaran mesin, dengan fitur yang dirancang untuk terus meningkatkan kinerja kueri dan mengurangi biaya penyimpanan untuk tabel. Tabel S3 dibuat khusus untuk menyimpan data tabular dalam format Apache Iceberg. Anda dapat menanyakan data tabular di Tabel S3 dengan mesin kueri populer, termasuk, Amazon Athena Amazon Redshift, dan Apache Spark. Secara default, Anda dapat membuat hingga 10 ember meja per Akun AWS per Wilayah AWS dan hingga 10.000 tabel per ember meja. Untuk informasi selengkapnya, lihat Bekerja dengan Tabel S3 dan bucket tabel.

> **Note**
>
> Semua ember meja dan meja bersifat pribadi dan tidak dapat dipublikasikan. Sumber daya ini hanya dapat diakses oleh pengguna yang secara eksplisit diberikan akses. Untuk memberikan akses, Anda dapat menggunakan kebijakan berbasis sumber daya IAM untuk bucket dan tabel tabel, serta kebijakan berbasis identitas IAM untuk pengguna dan peran. Untuk informasi selengkapnya, lihat Keamanan untuk Tabel S3.

Informasi tambahan tentang semua jenis ember

Saat Anda membuat bucket, memasukkan nama bucket dan memilih Wilayah AWS tempat bucket tersebut akan berada. Setelah membuat bucket, Anda tidak dapat mengubah nama atau Wilayah-nya. Nama bucket harus mengikuti aturan penamaan bucket berikut:

- Aturan penamaan bucket tujuan umum
- Aturan penamaan bucket direktori
- Aturan penamaan ember meja

Bucket juga:

- Mengatur ruang nama Amazon S3 di tingkat tertinggi. Untuk bucket tujuan umum, namespace ini adalah. `S3` Untuk bucket direktori, namespace ini adalah. `s3express` Untuk bucket tabel, namespace ini adalah. `s3tables`
- Mengidentifikasi akun yang bertanggung jawab atas biaya penyimpanan dan transfer data.
- Berfungsi sebagai unit agregasi untuk pelaporan penggunaan.

Objek

Objek adalah entitas dasar yang disimpan di Amazon S3. Objek terdiri atas data objek dan metadata. Metadata adalah serangkaian pasangan nilai-nama yang menjelaskan objek. Pasangan ini mencakup beberapa metadata default, seperti tanggal terakhir diubah, dan metadata HTTP standar, seperti `Content-Type`. Anda juga dapat menentukan metadata kustom pada saat objek disimpan.

Setiap objek dimuat dalam bucket. Misalnya, jika objek bernama `photos/puppy.jpg` disimpan dalam bucket tujuan `amzn-s3-demo-bucket` umum di Wilayah AS Barat (Oregon), maka objek tersebut dapat dialamatkan dengan menggunakan URL. `https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/photos/puppy.jpg` Untuk informasi lebih lanjut, lihat Mengakses Bucket.

Objek diidentifikasi secara unik dalam bucket dengan kunci (nama) dan ID versi (jika Penentuan Versi S3 diaktifkan di bucket). Untuk informasi selengkapnya tentang objek, lihat Gambaran umum objek Amazon S3.

Kunci

Kunci objek (atau nama kunci) adalah pengidentifikasi unik untuk objek dalam bucket. Setiap objek dalam bucket memiliki persis satu kunci. Kombinasi bucket, kunci objek, dan ID versi opsional (jika Penentuan Versi S3 diaktifkan untuk bucket) secara unik mengidentifikasi setiap objek. Jadi Anda dapat membayangkan Amazon S3 sebagai peta data dasar antara "bucket + kunci + versi" dan objek itu sendiri.

Setiap objek di Amazon S3 dapat dialamatkan secara unik melalui kombinasi titik akhir layanan web, nama bucket, kunci, dan secara opsional, versi. Misalnya, di URL `https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/photos/puppy.jpg`, *amzn-s3-demo-bucket* adalah nama bucket dan `photos/puppy.jpg` merupakan kuncinya.

Untuk informasi selengkapnya tentang kunci objek, lihat Menamai objek Amazon S3.

Penentuan Versi S3

Anda dapat menggunakan Penentuan Versi S3 untuk menyimpan beberapa varian objek dalam bucket yang sama. Dengan Penentuan Versi S3, Anda dapat menyimpan, mengambil, dan memulihkan setiap versi dari setiap objek yang disimpan dalam bucket Anda. Anda dapat lebih mudah memulihkan dari tindakan pengguna yang tidak diinginkan dan kegagalan aplikasi.

Untuk informasi selengkapnya, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.

ID Versi

Saat Anda mengaktifkan Penentuan Versi S3 dalam bucket, S3 Amazon menghasilkan ID versi unik untuk setiap objek yang ditambahkan ke bucket. Objek yang sudah ada di bucket pada saat Anda mengaktifkan Penentuan Versi memiliki ID versi . Jika Anda memodifikasi objek ini (atau lainnya) dengan operasi lain, seperti CopyObjectdan PutObject, objek baru mendapatkan ID versi unik.

Untuk informasi selengkapnya, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.

Kebijakan bucket

Kebijakan bucket adalah kebijakan berbasis sumber daya AWS Identity and Access Management (IAM) yang dapat Anda gunakan untuk memberikan izin akses ke bucket dan objek di dalamnya. Hanya pemilik bucket yang dapat mengaitkan kebijakan dengan bucket. Izin yang dipasang pada

bucket berlaku untuk semua objek di bucket yang dimiliki oleh pemilik bucket. Kebijakan bucket dibatasi hingga ukuran 20 KB.

Kebijakan bucket menggunakan bahasa kebijakan akses berbasis JSON yang standar di seluruh AWS. Anda dapat menggunakan kebijakan bucket untuk menambah atau menolak izin objek dalam bucket. Kebijakan bucket mengizinkan atau menolak permintaan berdasarkan elemen-elemen dalam kebijakan, termasuk pemohon, tindakan S3, sumber daya, dan aspek atau kondisi permintaan (misalnya, alamat IP yang digunakan untuk membuat permintaan). Misalnya, Anda dapat membuat kebijakan bucket yang memberikan izin lintas akun untuk mengunggah objek ke bucket S3 sekaligus memastikan pemilik bucket memiliki kendali penuh atas objek yang diunggah. Untuk informasi selengkapnya, lihat Contoh kebijakan bucket Amazon S3.

Dalam kebijakan bucket, Anda dapat menggunakan karakter wildcard di Amazon Resource Names (ARNs) dan nilai lainnya untuk memberikan izin ke subset objek. Misalnya, Anda dapat mengendalikan akses ke kelompok objek yang dimulai dengan prefiks umum atau diakhiri dengan ekstensi tertentu, seperti `.html`.

Titik akses S3

Titik akses Amazon S3 diberi nama titik akhir jaringan dengan kebijakan akses khusus yang menjelaskan bagaimana data dapat diakses menggunakan titik akhir tersebut. Titik akses dilampirkan ke sumber data yang mendasarinya, seperti bucket tujuan umum, bucket direktori, atau volume FSx untuk OpenZFS, yang dapat Anda gunakan untuk melakukan operasi objek S3, seperti dan. `GetObject` `PutObject` Titik akses menyederhanakan pengelolaan akses data dalam skala besar untuk set data bersama di Amazon S3.

Setiap titik akses memiliki kebijakan jalur aksesnya sendiri. Anda dapat mengonfigurasi pengaturan Blokir Akses Publik untuk setiap titik akses yang dilampirkan ke bucket. Untuk membatasi akses data Amazon S3 ke jaringan privat, Anda juga dapat mengonfigurasi titik akses apa pun untuk menerima permintaan hanya dari cloud privat virtual (VPC).

Untuk informasi selengkapnya tentang titik akses untuk bucket tujuan umum, lihatMengelola akses ke kumpulan data bersama dengan titik akses. Untuk informasi selengkapnya tentang titik akses untuk bucket direktori, lihatMengelola akses ke kumpulan data bersama dalam bucket direktori dengan titik akses.

Daftar kontrol akses (ACLs)

Anda dapat menggunakannya ACLs untuk memberikan izin baca dan tulis kepada pengguna yang berwenang untuk bucket dan objek tujuan umum individu. Setiap bucket dan objek tujuan umum

memiliki ACL yang melekat padanya sebagai subresource. ACL mendefinisikan Akun AWS atau kelompok mana yang diberikan akses dan jenis akses. ACLs adalah mekanisme kontrol akses yang mendahului IAM. Untuk informasi lebih lanjut tentang ACLs, lihatGambaran umum daftar kontrol akses (ACL).

S3 Object Ownership adalah setelan tingkat ember Amazon S3 yang dapat Anda gunakan untuk mengontrol kepemilikan objek yang diunggah ke bucket dan untuk menonaktifkan atau mengaktifkan. ACLs Secara default, Kepemilikan Objek disetel ke setelan diberlakukan pemilik Bucket, dan semuanya ACLs dinonaktifkan. Ketika ACLs dinonaktifkan, pemilik bucket memiliki semua objek di bucket dan mengelola akses ke mereka secara eksklusif dengan menggunakan kebijakan manajemen akses.

Mayoritas kasus penggunaan modern di Amazon S3 tidak lagi memerlukan penggunaan. ACLs Kami menyarankan agar Anda tetap ACLs dinonaktifkan, kecuali dalam keadaan yang tidak biasa di mana Anda perlu mengontrol akses untuk setiap objek secara individual. Dengan ACLs dinonaktifkan, Anda dapat menggunakan kebijakan untuk mengontrol akses ke semua objek di bucket, terlepas dari siapa yang mengunggah objek ke bucket Anda. Untuk informasi selengkapnya, lihat Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.

Wilayah

Anda dapat memilih geografis Wilayah AWS tempat Amazon S3 menyimpan ember yang Anda buat. Anda dapat memilih Wilayah untuk mengoptimalkan latensi, meminimalkan biaya, atau memenuhi persyaratan peraturan. Objek yang disimpan di Wilayah AWS tidak pernah meninggalkan Wilayah kecuali Anda secara eksplisit mentransfer atau mereplikasi mereka ke Wilayah lain. Misalnya, objek yang disimpan di Wilayah Eropa (Irlandia) tidak pernah keluar dari Wilayah tersebut.

> **Note**
>
> Anda dapat mengakses Amazon S3 dan fitur-fiturnya hanya di Wilayah AWS yang diaktifkan untuk akun Anda. Untuk informasi selengkapnya tentang mengaktifkan Wilayah untuk membuat dan mengelola AWS sumber daya, lihat Mengelola Wilayah AWS di. Referensi Umum AWS

Untuk daftar titik akhir dan Wilayah Amazon S3, lihat Titik akhir dan Wilayah di Referensi Umum AWS.

Model konsistensi data Amazon S3

Amazon S3 memberikan read-after-write konsistensi yang kuat untuk permintaan PUT dan DELETE objek di bucket Amazon S3 Anda secara keseluruhan. Wilayah AWS Perilaku ini berlaku untuk penulisan objek baru dan permintaan PUT yang menimpa objek yang sudah ada dan permintaan DELETE. Selain itu, operasi baca di Amazon S3 Select, daftar kontrol akses Amazon S3 ACLs (), Tag Objek Amazon S3, dan metadata objek (misalnya, objek HEAD) sangat konsisten.

Pembaruan untuk satu kunci bersifat atomik. Misalnya, jika Anda membuat permintaan PUT ke kunci yang sudah ada dari satu utas dan melakukan permintaan GET pada kunci yang sama dari utas kedua secara bersamaan, Anda akan mendapatkan data lama atau data baru, tetapi tidak pernah sebagian atau data yang rusak.

Amazon S3 mencapai ketersediaan tinggi dengan mereplikasi data di beberapa server di pusat data AWS . Jika permintaan PUT berhasil, data Anda disimpan dengan aman. Setiap pembacaan (permintaan GET atau LIST) yang dimulai setelah penerimaan respons PUT yang berhasil akan mengembalikan data yang ditulis oleh permintaan PUT. Berikut adalah contoh dari perilaku ini:

- Proses menulis objek baru ke Amazon S3 dan segera mencantumkan kunci di dalam bucket-nya. Objek baru akan muncul dalam daftar.
- Proses menggantikan objek yang sudah ada dan segera mencoba membacanya. Amazon S3 mengembalikan data baru.
- Proses menggantikan objek yang sudah ada dan segera mencoba membacanya. Amazon S3 tidak mengembalikan data apa pun karena objek telah dihapus.
- Proses menghapus objek yang sudah ada dan segera mencantumkan kunci di dalam bucket-nya. Objek tidak muncul dalam daftar.

> **Note**
>
> - Amazon S3 tidak mendukung penguncian objek untuk penulisan bersamaan. Jika dua permintaan PUT secara bersamaan dibuat ke kunci yang sama, permintaan dengan stempel waktu terbaru akan menang. Jika ini menjadi masalah, Anda harus membangun mekanisme penguncian objek ke dalam aplikasi Anda.
> - Pembaruan bersifat berbasis kunci. Tidak ada cara untuk melakukan pembaruan atomik lintas kunci. Sebagai contoh, Anda tidak dapat memperbarui satu kunci menjadi tergantung

pada pembaruan kunci lain kecuali Anda merancang fungsionalitas ini ke dalam aplikasi Anda.

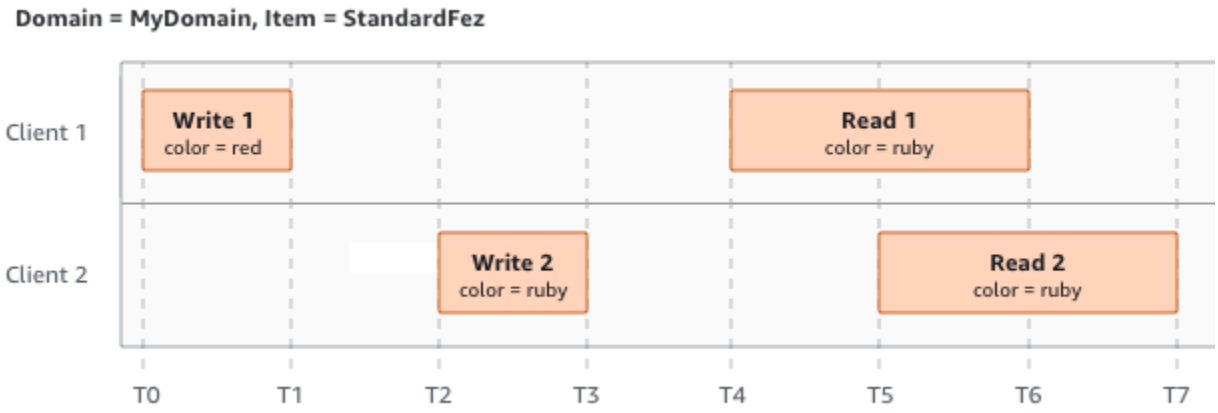
Konfigurasi bucket memiliki model konsistensi pada akhir. Secara khusus, ini berarti:

- Jika Anda menghapus bucket dan langsung membuat daftar semua bucket, bucket yang dihapus mungkin masih akan muncul dalam daftar.
- Jika Anda mengaktifkan Penentuan Versi pada bucket untuk pertama kalinya, mungkin diperlukan beberapa saat agar perubahan dapat dilakukan sepenuhnya. Kami menyarankan Anda menunggu selama 15 menit setelah mengaktifkan Penentuan Versi sebelum memberikan operasi tulis (PUT atau DELETE) pada objek di dalam bucket.

Aplikasi yang bersamaan

Bagian ini memberikan contoh perilaku yang diharapkan dari Amazon S3 ketika beberapa klien menulis ke item yang sama.

Dalam contoh ini, W1 (tulis 1) dan W2 (tulis 2) selesai sebelum awal R1 (baca 1) dan R2 (baca 2). Karena S3 sangat konsisten, R1 dan R2 keduanya mengembalikan `color = ruby`.



Dalam contoh berikutnya, W2 tidak selesai sebelum awal R1. Oleh karena itu, R1 mungkin mengembalikan `color = ruby` atau `color = garnet`. Namun, karena W1 dan W2 selesai sebelum dimulainya R2, R2 mengembalikan `color = garnet`.

Dalam contoh terakhir, W2 dimulai sebelum W1 telah menerima pengakuan. Oleh karena itu, penulisan ini dianggap bersamaan. Amazon S3 secara internal menggunakan last-writer-wins semantik untuk menentukan penulisan mana yang diutamakan. Namun, urutan di mana Amazon S3 menerima permintaan dan urutan aplikasi menerima pengakuan tidak dapat diprediksi karena berbagai faktor, seperti latensi jaringan. Misalnya, W2 mungkin diprakarsai oleh EC2 instance Amazon di Wilayah yang sama, sedangkan W1 mungkin diprakarsai oleh host yang lebih jauh. Cara terbaik untuk menentukan nilai akhir adalah dengan melakukan pembacaan setelah kedua penulisan telah diakui.

Layanan terkait

Setelah memuat data ke Amazon S3, Anda dapat menggunakannya dengan layanan lain AWS . Berikut adalah layanan yang mungkin paling sering Anda gunakan:

- Amazon Elastic Compute Cloud (Amazon EC2) — Menyediakan kapasitas komputasi yang aman dan dapat diskalakan di. AWS Cloud Menggunakan Amazon EC2 menghilangkan kebutuhan Anda

untuk berinvestasi dalam perangkat keras di muka, sehingga Anda dapat mengembangkan dan menerapkan aplikasi lebih cepat. Anda dapat menggunakan Amazon EC2 untuk meluncurkan server virtual sebanyak atau sesedikit yang Anda butuhkan, mengonfigurasi keamanan dan jaringan, dan mengelola penyimpanan.

- Amazon EMR–Membantu bisnis, peneliti, analis data, dan pengembang memproses data dalam jumlah besar dengan mudah dan hemat biaya. Amazon EMR menggunakan kerangka kerja Hadoop yang dihosting yang berjalan pada infrastruktur skala web Amazon dan Amazon S3. EC2
- AWS Keluarga Salju — Membantu pelanggan yang perlu menjalankan operasi di lingkungan non-pusat data yang keras, dan di lokasi di mana ada kurangnya konektivitas jaringan yang konsisten. Anda dapat menggunakan perangkat AWS Snow Family untuk mengakses penyimpanan dan daya komputasi penyimpanan secara lokal dan hemat biaya AWS Cloud di tempat-tempat di mana koneksi internet mungkin bukan pilihan.
- AWS Transfer Family–Menyediakan dukungan yang dikelola sepenuhnya untuk transfer file langsung masuk dan keluar dari Amazon S3 atau Amazon Elastic File System (Amazon EFS) menggunakan Secure Shell (SSH) File Transfer Protocol (SFTP), File Transfer Protocol melalui SSL (FTPS), dan File Transfer Protocol (FTP).

Mengakses Amazon S3

Anda dapat bekerja dengan Amazon S3 dengan salah satu cara berikut ini:

AWS Management Console

Konsol adalah antarmuka pengguna berbasis web untuk mengelola Amazon S3 AWS dan sumber daya. Jika Anda telah mendaftar Akun AWS, Anda dapat mengakses konsol Amazon S3 dengan masuk ke AWS Management Console dan memilih S3 dari halaman beranda. AWS Management Console

AWS Command Line Interface

Anda dapat menggunakan alat baris AWS perintah untuk mengeluarkan perintah atau membangun skrip di baris perintah sistem Anda untuk melakukan tugas AWS (termasuk S3).

The AWS Command Line Interface (AWS CLI) menyediakan perintah untuk satu set yang luas Layanan AWS. AWS CLI Ini didukung di Windows, macOS, dan Linux. Untuk memulai, lihat Panduan Pengguna AWS Command Line Interface . Untuk informasi selengkapnya tentang perintah untuk Amazon S3, lihat s3api dan s3control di AWS CLI Referensi Perintah.

AWS SDKs

AWS menyediakan SDKs (perangkat pengembangan perangkat lunak) yang terdiri dari pustaka dan kode sampel untuk berbagai bahasa dan platform pemrograman (Java, Python, Ruby, .NET, iOS, Android, dan sebagainya). AWS SDKs Menyediakan cara mudah untuk membuat akses terprogram ke S3 dan. AWS Amazon S3 adalah layanan REST. Anda dapat mengirim permintaan ke Amazon S3 menggunakan pustaka AWS SDK, yang membungkus API REST Amazon S3 yang mendasarinya dan menyederhanakan tugas pemrograman Anda. Misalnya, SDKs mengurus tugas-tugas seperti menghitung tanda tangan, menandatangani permintaan secara kriptografis, mengelola kesalahan, dan mencoba ulang permintaan secara otomatis. Untuk informasi tentang AWS SDKs, termasuk cara mengunduh dan menginstalnya, lihat Alat untuk AWS.

Setiap interaksi dengan Amazon S3 diautentikasi atau dilakukan secara anonim. Jika Anda menggunakan AWS SDKs, pustaka menghitung tanda tangan untuk otentikasi dari kunci yang Anda berikan. Untuk informasi selengkapnya tentang cara membuat permintaan ke Amazon S3, lihat Membuat permintaan.

API REST Amazon S3

Arsitektur Amazon S3 dirancang untuk menjadi bahasa pemrograman-netral, menggunakan antarmuka yang didukung AWS untuk menyimpan dan mengambil objek. Anda dapat mengakses S3 dan AWS secara terprogram menggunakan API REST Amazon S3. API REST adalah antarmuka HTTP ke Amazon S3. Dengan API REST, Anda menggunakan permintaan HTTP standar untuk membuat, dan menghapus bucket dan objek.

Untuk menggunakan API REST, Anda dapat menggunakan toolkit yang mendukung HTTP. Anda bahkan dapat menggunakan peramban untuk mengambil objek, selama objek tersebut dapat dibaca secara anonim.

API REST menggunakan header HTTP standar dan kode status, sehingga peramban dan toolkit standar bekerja sesuai harapan. Di beberapa area, kami telah menambahkan fungsi ke HTTP (misalnya, kami menambahkan header untuk mendukung kontrol akses). Dalam kasus ini, kami telah melakukan yang terbaik untuk menambahkan fungsionalitas baru dengan cara yang cocok dengan cara penggunaan HTTP standar.

Jika Anda melakukan panggilan API REST langsung di aplikasi Anda, Anda harus menulis kode untuk menghitung tanda tangan dan menambahkannya ke permintaan. Untuk informasi selengkapnya tentang cara membuat permintaan ke Amazon S3, lihat Membuat permintaan di Referensi API Amazon S3.

> **Note**
>
> Dukungan SOAP API melalui HTTP dihilangkan, tetapi masih tersedia melalui HTTPS. Fitur Amazon S3 baru tidak didukung untuk SOAP. Kami menyarankan Anda menggunakan REST API atau AWS SDKs.

Membayar Amazon S3

Harga untuk Amazon S3 dirancang agar Anda tidak perlu merencanakan persyaratan penyimpanan aplikasi Anda. Sebagian besar penyedia penyimpanan mengharuskan Anda untuk membeli kapasitas penyimpanan dan transfer jaringan dalam jumlah yang telah ditentukan sebelumnya. Dalam skenario ini, jika Anda melebihi kapasitas tersebut, layanan Anda akan dimatikan atau Anda akan dikenai biaya kelebihan penggunaan yang tinggi. Jika Anda tidak melebihi kapasitas tersebut, Anda membayar seolah-olah Anda menggunakan semuanya.

Amazon S3 mengenakan biaya hanya untuk yang benar-benar Anda gunakan, tanpa biaya tersembunyi dan biaya kelebihan. Model ini memberi Anda layanan biaya variabel yang dapat tumbuh dengan bisnis Anda sambil memberi Anda keuntungan biaya infrastruktur. AWS Untuk informasi selengkapnya, lihat Harga Amazon S3.

Ketika Anda mendaftar AWS, Anda Akun AWS secara otomatis mendaftar untuk semua layanan di AWS, termasuk Amazon S3. Nmaun, Anda hanya dikenai biaya untuk layanan yang digunakan. Jika Anda adalah pelanggan baru Amazon S3, Anda dapat memulai Amazon S3 secara gratis. Untuk informasi selengkapnya, lihat AWS tingkat gratis.

Untuk melihat tagihan Anda, buka Dasbor Manajemen Penagihan dan Biaya di konsol AWS Manajemen Penagihan dan Biaya. Untuk mempelajari selengkapnya tentang Akun AWS penagihan, lihat Panduan AWS Billing Pengguna. Jika Anda memiliki pertanyaan tentang AWS penagihan dan Akun AWS, hubungi AWS Support.

Kepatuhan PCI DSS

Amazon S3 mendukung pemrosesan, penyimpanan, dan transmisi data kartu kredit oleh pedagang atau penyedia layanan, serta telah divalidasi sesuai dengan Standar Keamanan Data (DSS) Industri Kartu Pembayaran (PCI). Untuk informasi selengkapnya tentang PCI DSS, termasuk cara meminta salinan PCI AWS Compliance Package, lihat PCI DSS Level 1.

Memulai dengan Amazon S3

Anda dapat memulai dengan Amazon S3 dengan bekerja dengan bucket dan objek. Bucket adalah kontainer untuk objek. Objek adalah file data dan metadata apa pun yang mendeskripsikan file tersebut.

Untuk menyimpan objek di Amazon S3, Anda membuat bucket dan kemudian mengunggah objek ke bucket. Bila objek ada di dalam bucket, Anda bisa membuka, mengunduh, dan memindahkannya. Saat Anda tidak lagi memerlukan objek atau bucket, Anda dapat membersihkan sumber daya Anda.

Dengan Amazon S3, Anda hanya membayar sesuai penggunaan Anda. Untuk informasi selengkapnya tentang fitur dan harga Amazon S3, lihat Amazon S3. Jika Anda adalah pelanggan baru Amazon S3, Anda dapat memulai Amazon S3 secara gratis. Untuk informasi selengkapnya, lihat AWS Tingkat Gratis.

> **Note**
>
> Untuk informasi selengkapnya tentang menggunakan kelas penyimpanan Amazon S3 Express One Zone dengan bucket direktori, lihat S3 Express One Zone dan Bekerja dengan bucket direktori.

Video: Memulai Amazon S3

Prasyarat

Sebelum memulai, konfirmasikan Anda telah menyelesaikan langkah-langkah dalam Menyiapkan Amazon S3.

Menyiapkan Amazon S3

Ketika Anda mendaftar AWS, Anda Akun AWS secara otomatis mendaftar untuk semua layanan di AWS, termasuk Amazon S3. Anda hanya membayar biaya layanan yang Anda gunakan.

Dengan Amazon S3, Anda hanya membayar sesuai penggunaan Anda. Untuk informasi selengkapnya tentang fitur dan harga Amazon S3, lihat Amazon S3. Jika Anda adalah pelanggan baru Amazon S3, Anda dapat memulai Amazon S3 secara gratis. Untuk informasi selengkapnya, lihat AWS Tingkat Gratis.

Untuk mengatur Amazon S3, gunakan langkah-langkah di bagian berikut.

Saat Anda mendaftar AWS dan menyiapkan Amazon S3, Anda dapat secara opsional mengubah bahasa tampilan di. AWS Management Console Untuk informasi lebih lanjut, lihat Mengubah bahasa AWS Management Console di AWS Management Console Panduan memulai.

Topik

- Mendaftar untuk Akun AWS
- Buat pengguna dengan akses administratif

Mendaftar untuk Akun AWS

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar untuk Akun AWS

1. Buka https://portal.aws.amazon.com/billing/pendaftaran.
2. Ikuti petunjuk online.

   Bagian dari prosedur pendaftaran melibatkan menerima panggilan telepon atau pesan teks dan memasukkan kode verifikasi pada keypad telepon.

   Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWSdibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya di akun. Sebagai praktik keamanan terbaik, tetapkan akses administratif ke pengguna, dan gunakan hanya pengguna root untuk melakukan tugas yang memerlukan akses pengguna root.

AWS mengirimi Anda email konfirmasi setelah proses pendaftaran selesai. Kapan saja, Anda dapat melihat aktivitas akun Anda saat ini dan mengelola akun Anda dengan masuk https://aws.amazon.comke/ dan memilih Akun Saya.

Buat pengguna dengan akses administratif

Setelah Anda mendaftar Akun AWS, amankan Pengguna root akun AWS, aktifkan AWS IAM Identity Center, dan buat pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

Amankan Anda Pengguna root akun AWS

1. Masuk ke AWS Management Consolesebagai pemilik akun dengan memilih pengguna Root dan memasukkan alamat Akun AWS email Anda. Di laman berikutnya, masukkan kata sandi.

   Untuk bantuan masuk dengan menggunakan pengguna root, lihat Masuk sebagai pengguna root di AWS Sign-In Panduan Pengguna.
2. Mengaktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

   Untuk petunjuk, lihat Mengaktifkan perangkat MFA virtual untuk pengguna Akun AWS root (konsol) Anda di Panduan Pengguna IAM.

Buat pengguna dengan akses administratif

1. Aktifkan Pusat Identitas IAM.

   Untuk mendapatkan petunjuk, silakan lihat Mengaktifkan AWS IAM Identity Center di Panduan Pengguna AWS IAM Identity Center .
2. Di Pusat Identitas IAM, berikan akses administratif ke pengguna.

   Untuk tutorial tentang menggunakan Direktori Pusat Identitas IAM sebagai sumber identitas Anda, lihat Mengkonfigurasi akses pengguna dengan default Direktori Pusat Identitas IAM di Panduan AWS IAM Identity Center Pengguna.

Masuk sebagai pengguna dengan akses administratif

- Untuk masuk dengan pengguna Pusat Identitas IAM, gunakan URL masuk yang dikirim ke alamat email saat Anda membuat pengguna Pusat Identitas IAM.

  Untuk bantuan masuk menggunakan pengguna Pusat Identitas IAM, lihat Masuk ke portal AWS akses di Panduan AWS Sign-In Pengguna.

Tetapkan akses ke pengguna tambahan

1. Di Pusat Identitas IAM, buat set izin yang mengikuti praktik terbaik menerapkan izin hak istimewa paling sedikit.

   Untuk petunjuknya, lihat Membuat set izin di Panduan AWS IAM Identity Center Pengguna.

2. Tetapkan pengguna ke grup, lalu tetapkan akses masuk tunggal ke grup.

   Untuk petunjuk, lihat Menambahkan grup di Panduan AWS IAM Identity Center Pengguna.

Langkah 1: Buat bucket S3 pertama Anda

Setelah mendaftar AWS, Anda siap membuat ember di Amazon S3 menggunakan file. AWS Management Console Setiap objek di Amazon S3 disimpan di bucket. Sebelum Anda dapat menyimpan data di Amazon S3, Anda harus membuat bucket.

> **Note**
>
> Untuk informasi selengkapnya tentang menggunakan kelas penyimpanan Amazon S3 Express One Zone dengan bucket direktori, lihat S3 Express One Zone dan Bekerja dengan bucket direktori.

> **Note**
>
> Anda tidak dikenakan biaya untuk membuat bucket. Anda hanya dikenakan untuk menyimpan objek di dalam bucket dan untuk memindahkan objek masuk dan keluar dari bucket. Biaya yang Anda tanggung melalui mengikuti contoh dalam panduan ini adalah minimal (kurang dari $1). Untuk informasi selengkapnya tentang biaya penyimpanan, lihat Harga Amazon S3.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi di bagian atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah tempat Anda ingin membuat ember.

   > **Note**
   >
   > - Setelah membuat bucket, Anda tidak dapat mengubah Region.
   > - Untuk meminimalkan latensi dan biaya serta memenuhi persyaratan regulasi, pilih Wilayah yang dekat dengan Anda. Objek yang disimpan di Wilayah tidak pernah keluar dari Wilayah kecuali Anda secara tegas mentransfer atau mereplikasinya ke

Wilayah lain. Untuk daftar Amazon S3 Wilayah AWS, lihat Layanan AWS titik akhir di. Referensi Umum Amazon Web Services

3. Di panel navigasi kiri, pilih Bucket tujuan umum.
4. Pilih Buat bucket. Halaman Buat bucket terbuka.
5. Untuk Nama bucket, masukkan nama untuk bucket Anda.

   Nama bucket harus:

   - Unik dalam partisi. Partisi adalah pengelompokan Wilayah. AWS saat ini memiliki tiga partisi: `aws` (Kawasan komersial), `aws-cn` (Wilayah Tiongkok), dan `aws-us-gov` (AWS GovCloud (US) Regions).
   - Panjangnya antara 3 hingga 63 karakter.
   - Hanya terdiri dari huruf kecil, angka, titik (`.`), dan tanda hubung (). `-` Untuk kompatibilitas terbaik, sebaiknya hindari penggunaan period (`.`) dalam nama bucket, kecuali bucket yang hanya digunakan untuk hosting situs statis.
   - Dimulai dan diakhiri dengan huruf atau angka.
   - Untuk daftar lengkap aturan penamaan ember, lihat. Aturan penamaan bucket tujuan umum

   Important

   - Setelah membuat bucket, Anda tidak dapat mengubah namanya.
   - Jangan sertakan informasi sensitif dalam nama bucket. Nama bucket terlihat dalam URLs yang menunjuk objek dalam bucket.

6. (Opsional) Di bawah Konfigurasi umum, Anda dapat memilih untuk menyalin setelan bucket yang ada ke bucket baru Anda. Jika Anda tidak ingin menyalin pengaturan bucket yang ada, lewati ke langkah berikutnya.

   Note

   Opsi ini:

   - Tidak tersedia di AWS CLI dan hanya tersedia di konsol Amazon S3
   - Tidak menyalin kebijakan bucket dari bucket yang ada ke bucket baru

Untuk menyalin setelan bucket yang ada, di bagian Salin setelan dari bucket yang ada, pilih Pilih bucket. Jendela Select bucket terbuka. Temukan bucket dengan pengaturan yang ingin Anda salin, lalu pilih Pilih bucket. Jendela Choose bucket ditutup, dan jendela Create bucket terbuka kembali.

Di bawah Salin pengaturan dari bucket yang ada, Anda sekarang melihat nama bucket yang Anda pilih. Pengaturan bucket baru Anda sekarang cocok dengan pengaturan bucket yang Anda pilih. Jika Anda ingin menghapus pengaturan yang disalin, pilih Pulihkan default. Tinjau setelan bucket yang tersisa di halaman Buat bucket. Jika Anda tidak ingin membuat perubahan apa pun, Anda dapat melompat ke langkah terakhir.

7. Di bawah Kepemilikan Objek, untuk menonaktifkan atau mengaktifkan ACLs dan mengontrol kepemilikan objek yang diunggah di bucket Anda, pilih salah satu setelan berikut:

   ACLs dinonaktifkan

   - Pemilik bucket diberlakukan (default) — ACLs dinonaktifkan, dan pemilik bucket secara otomatis memiliki dan memiliki kontrol penuh atas setiap objek di bucket tujuan umum. ACLs tidak lagi memengaruhi izin akses ke data di bucket tujuan umum S3. Bucket menggunakan kebijakan secara eksklusif untuk menentukan kontrol akses.

     Secara default, ACLs dinonaktifkan. Mayoritas kasus penggunaan modern di Amazon S3 tidak lagi memerlukan penggunaan. ACLs Kami menyarankan agar Anda tetap ACLs dinonaktifkan, kecuali dalam keadaan yang tidak biasa di mana Anda harus mengontrol akses untuk setiap objek secara individual. Untuk informasi selengkapnya, lihat Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.

   ACLs diaktifkan

   - Pemilik bucket yang dipilih—Pemilik bucket memiliki dan diberikan kendali penuh atas objek baru yang ditulis akun lain ke bucket dengan ACL `bucket-owner-full-control` yang dibatasi.

     Jika Anda menerapkan pengaturan Pemilik bucket yang dipilih, agar semua unggahan Amazon S3 menyertakan ACL `bucket-owner-full-control` yang terekam, Anda dapat menambahkan kebijakan bucket yang hanya mengizinkan unggahan objek yang menggunakan ACL ini.

- Penulis objek — Akun AWS Yang mengunggah objek memiliki objek, memiliki kontrol penuh atasnya, dan dapat memberikan pengguna lain akses ke sana melalui. ACLs

> **Note**
>
> Pengaturan default-nya adalah Pemilik Bucket yang diberlakukan. Untuk menerapkan pengaturan default dan tetap ACLs dinonaktifkan, hanya `s3:CreateBucket` izin yang diperlukan. Untuk mengaktifkan ACLs, Anda harus memiliki `s3:PutBucketOwnershipControls` izin.

8. Di bawah Pengaturan Blokir Akses Publik untuk bucket ini, pilih pengaturan Blokir Akses Publik yang ingin Anda terapkan ke bucket.

   Secara default, semua pengaturan Blokir Akses Publik untuk bucket direktori diaktifkan. Kami menyarankan Anda tetap mengaktifkan semua pengaturan, kecuali Anda tahu bahwa Anda perlu menonaktifkan satu atau beberapa pengaturan untuk kasus penggunaan spesifik Anda. Untuk informasi lebih lanjut tentang pemblokiran akses publik, lihat Melakukan blok akses publik ke penyimpanan Amazon S3 Anda.

   > **Note**
   >
   > Untuk mengaktifkan semua pengaturan Blokir Akses Publik, hanya izin `s3:CreateBucket` yang diperlukan. Untuk mematikan pengaturan Blokir Akses Publik, Anda harus memiliki izin `s3:PutBucketPublicAccessBlock`.

9. (Opsional) Secara default, Bucket Versioning dinonaktifkan. Penentuan Versi adalah cara menyimpan beberapa varian objek dalam bucket yang sama. Anda dapat menggunakan Penentuan Versi untuk menyimpan, mengambil, dan memulihkan setiap versi dari setiap objek yang disimpan dalam bucket Anda. Dengan versioning, Anda dapat lebih mudah memulihkan dari tindakan pengguna yang tidak diinginkan dan kegagalan aplikasi. Untuk informasi selengkapnya tentang penentuan versi, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.

   Untuk mengaktifkan pembuatan versi di bucket, pilih Aktifkan.

10. (Opsional) Di bawah Tanda, Anda dapat memilih untuk menambahkan tanda ke bucket Anda. Dengan alokasi AWS biaya, Anda dapat menggunakan tag bucket untuk membubuhi keterangan penagihan penggunaan bucket. Tanda adalah pasangan nilai kunci yang mewakili label yang

Anda tetapkan ke bucket. Untuk informasi selengkapnya, lihat the section called “Menggunakan tanda alokasi biaya”.

Untuk menambahkan tanda bucket, masukkan Kunci dan secara opsional Nilai, lalu pilih Tambahkan Tanda.

11. Untuk mengonfigurasi enkripsi Default, di bawah Jenis enkripsi, pilih salah satu dari berikut ini:

    - Enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3)
    - Enkripsi sisi server dengan AWS Key Management Service kunci (SSE-KMS)
    - Enkripsi sisi server dua lapis dengan kunci () (DSSE-KMS AWS Key Management Service)AWS KMS

    > **Important**
    >
    > Jika Anda menggunakan opsi SSE-KMS atau DSSE-KMS untuk konfigurasi enkripsi default Anda, Anda tunduk pada kuota permintaan per detik (RPS) sebesar. AWS KMSUntuk informasi selengkapnya tentang AWS KMS kuota dan cara meminta kenaikan kuota, lihat Kuota di Panduan Pengembang AWS Key Management Service .

    Bucket dan objek baru dienkripsi dengan menggunakan enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3) sebagai tingkat dasar konfigurasi enkripsi. Untuk informasi selengkapnya tentang enkripsi default, lihat Mengatur perilaku enkripsi sisi server default untuk bucket Amazon S3. Untuk informasi selengkapnya tentang SSE-S3, lihat. Menggunakan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3)

    Untuk informasi selengkapnya tentang penggunaan enkripsi sisi server untuk mengenkripsi data Anda, lihat. the section called “Enkripsi data”

12. Jika Anda memilih enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3) atau enkripsi sisi server dual-layer dengan kunci () (DSSE-KMS), lakukan hal berikut: AWS Key Management Service AWS KMS

    a. Di bawah AWS KMS kunci, tentukan kunci KMS Anda dengan salah satu cara berikut ini:

        - Untuk memilih dari daftar kunci KMS yang tersedia, pilih Pilih dari Anda AWS KMS keys, dan pilih kunci KMS Anda dari daftar kunci yang tersedia.

Kunci Kunci yang dikelola AWS (`aws/s3`) dan kunci terkelola pelanggan Anda muncul dalam daftar ini. Untuk informasi selengkapnya tentang CMK, lihat Kunci pelanggan dan AWS kunci di AWS Key Management Service Panduan Pengembang.

- Untuk memasukkan ARN kunci KMS, pilih Masukkan AWS KMS key ARN, dan masukkan ARN kunci KMS Anda di bidang yang muncul.
- Untuk membuat kunci terkelola pelanggan baru di AWS KMS konsol, pilih Buat kunci KMS.

  Untuk informasi selengkapnya tentang membuat AWS KMS key, lihat Membuat kunci di Panduan AWS Key Management Service Pengembang.

> **Important**
>
> Anda hanya dapat menggunakan tombol KMS yang tersedia Wilayah AWS sama dengan bucket. Konsol Amazon S3 hanya mencantumkan kunci 100 KMS pertama di Wilayah yang sama dengan bucket. Untuk menggunakan kunci KMS yang tidak terdaftar, Anda harus memasukkan ARN kunci KMS Anda. Jika Anda ingin menggunakan kunci KMS yang dimiliki oleh akun yang berbeda, Anda harus terlebih dahulu memiliki izin untuk menggunakan kunci tersebut, dan kemudian Anda harus memasukkan ARN kunci KMS. Untuk informasi selengkapnya tentang izin lintas akun untuk kunci KMS, lihat Membuat kunci KMS yang dapat digunakan akun lain di Panduan Pengembang AWS Key Management Service . Untuk informasi lebih lanjut tentang SSE-KMS, lihat Menentukan enkripsi di sisi server dengan AWS KMS (SSE-KMS). Untuk informasi lebih lanjut tentang DSSE-KMS, lihat the section called “Enkripsi di sisi server dua lapis (DSSE-KMS)”.
>
> Saat Anda menggunakan enkripsi sisi server AWS KMS key untuk Amazon S3, Anda harus memilih kunci KMS enkripsi simetris. Amazon S3 hanya mendukung kunci KMS enkripsi simetris dan tidak mendukung kunci KMS asimetris. Untuk informasi selengkapnya, lihat Mengidentifikasi tombol KMS simetris dan asimetris dalam Panduan Pengembang AWS Key Management Service .

b. Saat mengonfigurasi bucket untuk menggunakan enkripsi default dengan SSE-KMS, Anda juga dapat menggunakan S3 Bucket Keys. S3 Bucket Keys menurunkan biaya enkripsi dengan mengurangi lalu lintas permintaan dari Amazon S3 ke. AWS KMS Untuk informasi selengkapnya, lihat Mengurangi biaya SSE-KMS dengan Kunci Bucket Amazon S3. Kunci Bucket S3 tidak didukung untuk DSSE-KMS.

Secara default, Kunci Bucket S3 diaktifkan di konsol Amazon S3. Sebaiknya biarkan Kunci Bucket S3 diaktifkan untuk menurunkan biaya Anda. Untuk menonaktifkan Kunci Bucket S3 untuk bucket Anda, di bawah Kunci Bucket, pilih Nonaktifkan.

13. (Opsional) Kunci Objek S3 membantu melindungi objek baru agar tidak dihapus atau ditimpa. Untuk informasi selengkapnya, lihat Mengunci objek dengan Object Lock. Jika Anda ingin mengaktifkan S3 Object Lock, lakukan hal berikut:

    a. Pilih Pengaturan lanjutan.

    > **Important**
    >
    > Mengaktifkan Object Lock secara otomatis memungkinkan pembuatan versi untuk bucket. Setelah mengaktifkan dan berhasil membuat bucket, Anda juga harus mengonfigurasi pengaturan penyimpanan default Object Lock dan penahanan legal pada tab Properties bucket.

    b. Jika Anda ingin mengaktifkan Kunci Objek, pilih Aktifkan, baca peringatan yang muncul, lalu setujui.

    > **Note**
    >
    > Untuk membuat bucket yang diaktifkan Object Lock, Anda harus memiliki izin berikut:`s3:CreateBucket`,`s3:PutBucketVersioning`, dan`s3:PutBucketObjectLockConfiguration`.

14. Pilih Buat bucket.

Anda telah membuat bucket di Amazon S3.

Langkah selanjutnya

Untuk menambahkan objek ke bucket Anda, lihat Langkah 2: Unggah objek ke bucket Anda.

Langkah 2: Unggah objek ke bucket Anda

Setelah membuat bucket di Amazon S3, Anda siap untuk mengunggah objek ke bucket. Objek dapat berupa jenis file apa pun: file teks, foto, video, dan sebagainya.

> **Note**
>
> Untuk informasi selengkapnya tentang menggunakan kelas penyimpanan Amazon S3 Express One Zone dengan bucket direktori, lihat S3 Express One Zone dan Bekerja dengan bucket direktori.

Untuk mengunggah objek ke bucket

1. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di daftar Bucket, pilih nama bucket tempat Anda ingin mengunggah objek.
3. Di tab Objek untuk bucket Anda, pilih Unggah.
4. Di Bawah File dan folder, pilih Tambahkan file.
5. Pilih file yang akan diunggah, lalu pilih Buka.
6. Pilih Unggah.

Anda berhasil mengunggah objek ke bucket Anda.

Langkah selanjutnya

Untuk melihat objek Anda, lihat Langkah 3: Mengunduh objek.

Langkah 3: Mengunduh objek

Setelah mengunggah sebuah objek ke bucket, Anda dapat melihat informasi tentang objek tersebut dan mengunduhnya ke komputer lokal.

> **Note**
>
> Untuk informasi selengkapnya tentang menggunakan kelas penyimpanan Amazon S3 Express One Zone dengan bucket direktori, lihat S3 Express One Zone dan Bekerja dengan bucket direktori.

Menggunakan konsol S3

Bagian ini menjelaskan cara menggunakan konsol Amazon S3 untuk mengunduh objek dari bucket S3.

> **Note**
>
> - Anda hanya dapat mengunduh satu objek dalam satu waktu.
> - Jika Anda menggunakan konsol Amazon S3 untuk mengunduh objek yang nama kuncinya diakhiri dengan titik (`.`), titik tersebut dihapus dari nama kunci objek yang diunduh. Untuk mempertahankan periode di akhir nama objek yang diunduh, Anda harus menggunakan AWS Command Line Interface (AWS CLI) AWS SDKs, atau Amazon S3 REST API.

Untuk mengunduh objek dari bucket S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum atau ember Direktori.
3. Dalam daftar bucket, pilih nama bucket tempat Anda ingin mengunduh objek.
4. Anda dapat mengunduh objek dari bucket S3 dengan cara berikut:
   - Pilih kotak centang di samping objek, dan pilih Unduh. Jika Anda ingin mengunduh objek ke folder tertentu, pada menu Tindakan, pilih Unduh sebagai.
   - Jika Anda ingin mengunduh versi objek tertentu, aktifkan Tampilkan versi (terletak di samping kotak pencarian). Centang kotak di samping versi objek yang Anda inginkan, dan pilih Unduh. Jika Anda ingin mengunduh objek ke folder tertentu, pada menu Tindakan, pilih Unduh sebagai.

Anda berhasil mengunduh objek.

Langkah selanjutnya

Untuk menyalin dan menempel objek Anda di Amazon S3, lihat Langkah 4: Salin objek ke folder.

Langkah 4: Salin objek ke folder

Anda sudah menambahkan objek ke dalam bucket dan mengunduh objek. Sekarang, buat folder dan salin objek dan tempelkan ke folder tersebut.

> **Note**
>
> Untuk informasi selengkapnya tentang menggunakan kelas penyimpanan Amazon S3 Express One Zone dengan bucket direktori, lihat S3 Express One Zone dan Bekerja dengan bucket direktori.

Untuk menyalin objek ke folder

1. Di daftar Bucket, pilih nama bucket Anda.
2. Pilih Buat folder dan konfigurasikan folder baru:
   a. Masukkan nama folder (misalnya, `favorite-pics`).
   b. Untuk pengaturan enkripsi folder, pilih Nonaktifkan.
   c. Pilih Simpan.
3. Navigasikan ke bucket atau folder Amazon S3 yang berisi objek yang ingin Anda salin.
4. Centang kotak di sebelah kiri nama objek yang ingin Anda salin.
5. Pilih Tindakan dan pilih Salin dari daftar opsi yang muncul.

   Atau, pilih Salin dari opsi di kanan atas.
6. Pilih folder tujuan:
   a. Pilih Jelajahi S3.
   b. Pilih tombol opsi di sebelah kiri nama folder.

      Untuk menavigasiakn ke folder dan memilih subfolder sebagai tujuan, pilih nama folder.
   c. Pilih Pilih tujuan.

   Jalur ke folder tujuan akan muncul di kotak Tujuan. Di Tujuan, Anda dapat secara bergantian memasukkan jalur tujuan Anda, misalnya, s3:////. *`bucket-name folder-name`*
7. Di kanan bawah, pilih Salin.

Amazon S3 menyalin objek Anda ke folder tujuan.

Langkah selanjutnya

Untuk menghapus objek dan bucket di Amazon S3, lihat Langkah 5: Hapus objek dan bucket Anda.

Langkah 5: Hapus objek dan bucket Anda

Saat Anda tidak lagi memerlukan objek atau bucket, kami menyarankan Anda menghapusnya untuk mencegah biaya lebih lanjut. Jika Anda menyelesaikan panduan memulai ini sebagai latihan pembelajaran, dan Anda tidak berencana untuk menggunakan bucket atau objek Anda, kami sarankan Anda menghapus bucket dan objek Anda sehingga biaya tidak lagi bertambah.

Sebelum menghapus bucket, kosongkan bucket atau hapus objek di dalam bucket. Setelah Anda menghapus objek dan bucket Anda, objek tersebut tidak lagi tersedia.

Jika Anda ingin terus menggunakan nama bucket yang sama, sebaiknya hapus objek atau kosongkan bucket, tetapi jangan menghapus bucket. Setelah Anda menghapus bucket, nama akan tersedia untuk digunakan kembali. Namun, yang lain Akun AWS mungkin membuat ember dengan nama yang sama sebelum Anda memiliki kesempatan untuk menggunakannya kembali.

> **Note**
>
> Untuk informasi selengkapnya tentang menggunakan kelas penyimpanan Amazon S3 Express One Zone dengan bucket direktori, lihat S3 Express One Zone dan Bekerja dengan bucket direktori.

Topik

- Menghapus objek
- Mengosongkan bucket Anda
- Menghapus bucket Anda

Menghapus objek

Jika ingin memilih objek mana yang Anda hapus tanpa mengosongkan semua objek dari bucket, Anda dapat menghapus objek.

1. Di daftar Bucket, pilih nama bucket yang ingin Anda hapus objeknya.
2. Pilih objek yang ingin Anda hapus.
3. Pilih Hapus dari opsi di kanan atas.
4. Pada halaman Hapus objek, ketik **`delete`** untuk mengonfirmasi penghapusan objek Anda.
5. Pilih Hapus objek.

Mengosongkan bucket Anda

Apabila Anda berencana menghapus bucket, Anda harus terlebih dahulu mengosongkan bucket Anda, yang akan menghapus seluruh objek dalam bucket.

Mengosongkan bucket

1. Di daftar Bucket, pilih bucket yang ingin Anda kosongkan, lalu pilih Kosongkan.
2. Untuk mengonfirmasi bahwa Anda ingin mengosongkan bucket dan menghapus semua objek di dalamnya, di Bucket kosong, ketik **`permanently delete`**.

   > **Important**
   >
   > Mengosongkan bucket tidak dapat diurungkan. Objek yang ditambahkan ke bucket saat tindakan bucket kosong sedang berlangsung akan dihapus.

3. Untuk mengosongkan bucket dan menghapus semua objek di dalamnya, lalu pilih Kosongkan.

   Halaman Bucket kosong: Status yang terbuka dapat Anda gunakan untuk meninjau ringkasan penghapusan objek yang gagal dan berhasil.
4. Untuk kembali ke daftar bucket Anda, pilih Keluar.

Menghapus bucket Anda

Setelah mengosongkan bucket atau menghapus semua objek dari bucket, Anda dapat menghapus bucket.

1. Untuk menghapus bucket, dalam daftar Bucket, pilih bucket.
2. Pilih Hapus.

3. Untuk mengonfirmasi penghapusan, di Hapus bucket, ketik nama bucket.

> **Important**
>
> Menghapus bucket tidak dapat dibatalkan. Nama bucket bersifat unik. Jika Anda menghapus bucket, AWS pengguna lain dapat menggunakan nama tersebut. Jika Anda ingin terus menggunakan nama bucket yang sama, jangan menghapus bucket Anda. Sebaliknya, kosongkan dan simpan bucket.

4. Untuk menghapus bucket Anda, pilih Hapus bucket.

Langkah selanjutnya

Dalam contoh sebelumnya, Anda telah mempelajari cara melakukan beberapa tugas dasar Amazon S3.

Topik-topik berikut ini menjelaskan jalur pembelajaran yang dapat Anda gunakan untuk mendapatkan pemahaman yang lebih mendalam tentang Amazon S3 sehingga Anda dapat mengimplementasikannya di aplikasi Anda.

> **Note**
>
> Untuk informasi selengkapnya tentang menggunakan kelas penyimpanan Amazon S3 Express One Zone dengan bucket direktori, lihat S3 Express One Zone dan Bekerja dengan bucket direktori.

Topik

- Memahami kasus penggunaan umum
- Mengontrol akses ke bucket dan objek Anda
- Lindungi dan pantau penyimpanan Anda
- Mengembangkan dengan Amazon S3
- Belajar dari tutorial
- Menjelajahi pelatihan dan dukungan

Memahami kasus penggunaan umum

Anda dapat menggunakan Amazon S3 untuk mendukung kasus penggunaan spesifik Anda. AWS Pustaka Solusi dan AWS Blog menyediakan informasi dan tutorial khusus kasus penggunaan. Berikut ini adalah beberapa kasus penggunaan umum untuk Amazon S3:

- Cadangan dan penyimpanan–Menggunakan fitur manajemen penyimpanan Amazon S3 untuk mengelola biaya, memenuhi persyaratan peraturan, mengurangi latensi, dan menyimpan beberapa salinan data Anda yang berbeda untuk persyaratan kepatuhan.
- Hosting aplikasi–Menyebarkan, menginstal, dan mengelola aplikasi web yang andal, sangat dapat diskalakan, dan berbiaya rendah. Misalnya, Anda dapat mengonfigurasi bucket Amazon S3 untuk melakukan hosting situs web statis. Untuk informasi selengkapnya, lihat Hosting situs web statis menggunakan Amazon S3.
- Hosting media–Membangun infrastruktur yang sangat tersedia yang menampung unggahan dan unduhan video, foto, atau musik.
- Pengiriman perangkat lunak–Melakukan hosting aplikasi perangkat lunak untuk diunduh pelanggan.

Mengontrol akses ke bucket dan objek Anda

Amazon S3 menyediakan beragam fitur dan alat keamanan. Untuk ikhtisar, lihat Kontrol akses di Amazon S3.

Secara default, bucket S3 dan objek di dalamnya bersifat pribadi. Anda hanya memiliki akses ke sumber daya S3 yang Anda buat. Anda dapat menggunakan fitur berikut untuk memberikan izin sumber daya terperinci yang mendukung kasus penggunaan spesifik Anda atau untuk mengaudit izin sumber daya Amazon S3 Anda.

- Blokir Akses Publik S3–Memblokir akses publik ke bucket dan objek S3. Secara default, pengaturan Blokir Akses Publik diaktifkan di tingkat bucket.
- AWS Identity and Access Management Identitas (IAM) — Gunakan IAM atau AWS IAM Identity Center untuk membuat identitas IAM di Anda untuk mengelola akses Akun AWS ke sumber daya Amazon S3 Anda. Misalnya, Anda dapat menggunakan IAM dengan Amazon S3 untuk mengontrol jenis akses yang dimiliki pengguna atau grup pengguna ke bucket Amazon S3 yang Anda miliki. Akun AWS Untuk informasi selengkapnya tentang identitas IAM dan praktik terbaik, lihat Identitas IAM (pengguna, grup pengguna, dan peran) dalam Panduan Pengguna IAM.

- Kebijakan bucket–Gunakan bahasa kebijakan berbasis IAM untuk mengonfigurasi izin berbasis sumber daya untuk bucket S3 dan objek di dalamnya.
- Daftar kontrol akses (ACLs) — Berikan izin baca dan tulis untuk masing-masing bucket dan objek kepada pengguna yang berwenang. Sebagai aturan umum, sebaiknya gunakan kebijakan berbasis sumber daya S3 (kebijakan bucket dan kebijakan jalur akses) atau kebijakan pengguna IAM untuk kontrol akses, bukan. ACLs Kebijakan adalah opsi kontrol akses yang disederhanakan dan lebih fleksibel. Dengan kebijakan bucket dan kebijakan titik akses, Anda dapat menentukan aturan yang berlaku secara luas di semua permintaan ke sumber daya Amazon S3 Anda. Untuk informasi selengkapnya tentang kasus tertentu saat Anda menggunakan, ACLs bukan kebijakan berbasis sumber daya atau kebijakan pengguna IAM, lihat. Identity and Access Management untuk Amazon S3
- Kepemilikan Objek S3–Ambil kepemilikan setiap objek di bucket Anda, sederhanakan manajemen akses untuk data yang disimpan di Amazon S3. S3 Object Ownership adalah pengaturan tingkat ember Amazon S3 yang dapat Anda gunakan untuk menonaktifkan atau mengaktifkan. ACLs Secara default, ACLs dinonaktifkan. Dengan ACLs dinonaktifkan, pemilik bucket memiliki semua objek di bucket dan mengelola akses ke data secara eksklusif dengan menggunakan kebijakan manajemen akses.
- IAM Access Analyzer for S3–Mengevaluasi dan memantau kebijakan akses bucket S3 Anda, memastikan bahwa kebijakan hanya menyediakan akses yang dimaksudkan ke sumber daya S3 Anda.

Lindungi dan pantau penyimpanan Anda

- Melindungi penyimpanan Anda — Setelah Anda membuat bucket dan mengunggah objek di Amazon S3, Anda dapat melindungi penyimpanan objek Anda. Misalnya, Anda dapat menggunakan kontrol failover Versi S3, Replikasi S3, dan Titik Akses Multi-Wilayah untuk pemulihan bencana, untuk mencadangkan data Anda, dan Kunci Objek S3 AWS Backup untuk mengatur periode retensi, mencegah penghapusan dan penimpaan, serta memenuhi persyaratan kepatuhan.
- Memantau penyimpanan Anda — Pemantauan adalah bagian penting dalam menjaga keandalan, ketersediaan, dan kinerja Amazon S3 dan solusi Anda AWS . Anda dapat memantau aktivitas dan biaya penyimpanan. Selain itu, kami menyarankan Anda mengumpulkan data pemantauan dari semua bagian solusi AWS Anda sehingga Anda dapat lebih mudah melakukan debug kegagalan multipoin jika terjadi.

Anda juga dapat menggunakan analitik dan wawasan di Amazon S3 untuk memahami, menganalisis, dan mengoptimalkan penggunaan penyimpanan Anda. Misalnya, menggunakan Lensa Penyimpanan Amazon S3 untuk memahami, menganalisis, dan mengoptimalkan penyimpanan Anda. Lensa Penyimpanan S3 menyediakan 29+ metrik penggunaan dan aktivitas serta dasbor interaktif untuk mengumpulkan data untuk seluruh organisasi, akun tertentu, Wilayah, bucket, atau awalan Anda. Gunakan Analisis Kelas Penyimpanan untuk menganalisis pola akses penyimpanan guna memutuskan kapan saatnya memindahkan data Anda ke kelas penyimpanan yang lebih hemat biaya. Untuk mengelola biaya, Anda dapat menggunakan Siklus Hidup S3.

Mengembangkan dengan Amazon S3

Amazon S3 adalah layanan REST. Anda dapat mengirim permintaan ke Amazon S3 menggunakan REST API atau pustaka AWS SDK, yang membungkus API Amazon S3 REST yang mendasarinya, menyederhanakan tugas pemrograman Anda. Anda juga dapat menggunakan AWS Command Line Interface (AWS CLI) untuk melakukan panggilan API Amazon S3. Untuk informasi selengkapnya, lihat Membuat permintaan di Referensi API Amazon S3.

API REST Amazon S3 adalah antarmuka HTTP ke Amazon S3. Dengan API REST, Anda menggunakan permintaan HTTP standar untuk membuat, dan menghapus bucket dan objek. Untuk menggunakan API REST, Anda dapat menggunakan toolkit yang mendukung HTTP. Anda bahkan dapat menggunakan peramban untuk mengambil objek, selama objek tersebut dapat dibaca secara anonim. Untuk informasi selengkapnya, lihat Mengembangkan dengan Amazon S3 di Referensi API Amazon S3.

Untuk membantu Anda membangun aplikasi menggunakan bahasa pilihan Anda, kami menyediakan sumber daya berikut.

AWS CLI

Anda dapat mengakses fitur-fitur Amazon S3 menggunakan file AWS CLI. Untuk mengunduh dan mengonfigurasi AWS CLI, lihat Mengembangkan dengan Amazon S3 menggunakan AWS CLI di Referensi API Amazon S3.

AWS CLI Ini menyediakan dua tingkatan perintah untuk mengakses Amazon S3: perintah tingkat tinggi (s3) dan tingkat API (s3api dan perintah. `s3control` Perintah S3 tingkat tinggi menyederhanakan operasi tugas-tugas umum, seperti membuat, memanipulasi, dan menghapus objek dan bucket. Perintah s3api dan s3control mengekspos akses langsung ke semua operasi

Amazon S3 API, yang dapat Anda gunakan untuk melakukan operasi lanjutan yang mungkin tidak mungkin dilakukan dengan perintah tingkat tinggi saja.

Untuk daftar AWS CLI perintah Amazon S3, lihat s3, s3api, dan s3control.

AWS SDKs dan Penjelajah

Anda dapat menggunakan AWS SDKs saat mengembangkan aplikasi dengan Amazon S3. AWS SDKs Menyederhanakan tugas pemrograman Anda dengan membungkus REST API yang mendasarinya. JavaScript Perpustakaan AWS Mobile SDKs dan Amplify juga tersedia untuk membangun aplikasi seluler dan web yang terhubung menggunakan. AWS

Selain itu AWS SDKs, AWS Explorers tersedia untuk Visual Studio dan Eclipse untuk Java IDE. Dalam hal ini, SDKs dan penjelajah dibundel bersama sebagai Toolkit. AWS

Untuk informasi selengkapnya, lihat Mengembangkan dengan Amazon S3 menggunakan Referensi API Amazon S3 AWS SDKs di Amazon S3.

Sampel Kode dan Pustaka

AWS Pusat Developer dan AWS Katalog Sampel Kode memiliki sampel kode dan pustaka yang ditulis khusus untuk Amazon S3. Anda dapat menggunakan sampel kode ini untuk memahami cara mengimplementasikan API Amazon S3. Anda juga dapat melihat Referensi API Amazon Simple Storage Service untuk memahami operasi Amazon S3 API secara detail.

Belajar dari tutorial

Anda dapat memulai dengan step-by-step tutorial untuk mempelajari lebih lanjut tentang Amazon S3. Tutorial ini ditujukan untuk lingkungan tipe lab, dan tutorial menggunakan nama perusahaan fiktif, nama pengguna fikif, dan sebagainya. Tujuannya adalah untuk memberikan pedoman umum. Produk ini tidak dimaksudkan untuk penggunaan secara langsung di lingkungan produksi tanpa tinjauan dan adaptasi yang cermat untuk memenuhi kebutuhan unik di lingkungan organisasi Anda.

Memulai

- Tutorial: Menyimpan dan mengambil file dengan Amazon S3
- Tutorial: Mulai menggunakan S3 Intelligent-Tiering
- Tutorial: Mulai menggunakan kelas penyimpanan Amazon S3 Glacier

Mengoptimalkan biaya penyimpanan

- Tutorial: Mulai menggunakan S3 Intelligent-Tiering
- Tutorial: Mulai menggunakan kelas penyimpanan Amazon S3 Glacier
- Tutorial: Mengoptimalkan biaya dan mendapatkan visibilitas ke dalam penggunaan dengan Lensa Penyimpanan S3

Mengelola penyimpanan

- Tutorial: Memulai dengan Titik Akses Multi-Wilayah Amazon S3
- Tutorial: Mereplikasi objek yang ada di bucket Amazon S3 Anda dengan Replikasi Batch S3

Hosting video dan situs web

- Tutorial: Hosting video streaming sesuai permintaan dengan Amazon S3, Amazon, dan CloudFront Amazon Route 53
- Tutorial: Mengonfigurasi situs web statis untuk Amazon S3
- Tutorial: Mengonfigurasi situs web statis menggunakan domain kustom yang terdaftar di Route 53

Pemrosesan data

- Tutorial: Mengubah data untuk aplikasi Anda dengan S3 Object Lambda
- Tutorial: Mendeteksi dan menyunting data PII dengan Objek S3 Lambda dan Amazon Comprehend
- Tutorial: Menggunakan S3 Lambda Objek untuk menandai gambar secara dinamis saat gambarnya diambil
- Tutorial: Video transcoding batch dengan Operasi Batch S3

Melindungi data

- Tutorial: Memeriksa integritas data di Amazon S3 dengan checksum tambahan
- Tutorial: Mereplikasi data di dalam dan di antara Wilayah AWS menggunakan Replikasi S3
- Tutorial: Melindungi data di Amazon S3 dari penghapusan yang tidak disengaja atau bug aplikasi menggunakan Penentuan Versi S3, Kunci Objek S3, dan Replikasi S3
- Tutorial: Mereplikasi objek yang ada di bucket Amazon S3 Anda dengan Replikasi Batch S3

Menjelajahi pelatihan dan dukungan

Anda dapat belajar dari AWS para ahli untuk memajukan keterampilan Anda dan mendapatkan bantuan ahli untuk mencapai tujuan Anda.

- Pelatihan–Materi pelatihan memberikan pendekatan langsung untuk mempelajari Amazon S3. Untuk informasi selengkapnya, lihat AWS Pelatihan dan sertifikasi dan AWS Pembicaraan teknologi online.
- Forum Diskusi–Di forum, Anda dapat meninjau posting untuk memahami apa yang dapat dan tidak dapat Anda lakukan dengan Amazon S3. Anda juga dapat memposting pertanyaan Anda. Untuk informasi selengkapnya, lihat Forum Diskusi.
- Dukungan Teknis–Jika Anda memiliki pertanyaan lebih lanjut, Anda dapat menghubungi Dukungan Teknis.

Memulai dengan Amazon S3 menggunakan AWS CLI

Anda dapat memulai dengan Amazon S3 dengan menggunakan AWS Command Line Interface (AWS CLI) untuk bekerja dengan bucket dan objek tujuan umum. Bucket adalah kontainer untuk objek. Objek adalah file data dan metadata apa pun yang mendeskripsikan file tersebut.

Untuk menyimpan objek di Amazon S3, Anda membuat bucket dan kemudian mengunggah objek ke bucket. Bila objek ada di dalam bucket, Anda bisa membuka, mengunduh, dan memindahkannya. Saat Anda tidak lagi memerlukan objek atau bucket, Anda dapat membersihkan sumber daya Anda.

Dengan Amazon S3, Anda hanya membayar sesuai penggunaan Anda. Untuk informasi selengkapnya tentang fitur dan harga Amazon S3, lihat Amazon S3. Jika Anda adalah pelanggan baru Amazon S3, Anda dapat memulai Amazon S3 secara gratis. Untuk informasi selengkapnya, lihat AWS Tingkat Gratis.

> **Note**
>
> Untuk informasi selengkapnya tentang menggunakan kelas penyimpanan Amazon S3 Express One Zone dengan bucket direktori, lihat Tutorial: Memulai dengan S3 Express One Zone dan Bekerja dengan bucket direktori.

Pengaturan

Sebelum Anda mulai menggunakan AWS CLI dengan Amazon S3, pastikan Anda memiliki:

- Mendaftar untuk sebuah AWS akun. Untuk petunjuk, lihat Menyiapkan Amazon S3.
- Membuat pengguna dengan `s3:*` izin. Untuk petunjuk, lihat Menyiapkan Amazon S3.
- Menginstal dan mengkonfigurasi file AWS CLI. Untuk petunjuk, lihat Menginstal atau memperbarui versi terbaru AWS CLI dari Panduan AWS Command Line Interface Pengguna.

Untuk memverifikasi bahwa AWS CLI sudah dikonfigurasi dengan benar, jalankan perintah berikut:

```
aws sts get-caller-identity
```

Untuk informasi selengkapnya, lihat get-caller-identity dalam AWS CLI Referensi Perintah.

Perintah ini menampilkan daftar perintah Amazon S3 yang tersedia jika diinstal dan AWS CLI dikonfigurasi dengan benar.

Langkah 1: Buat bucket Amazon S3 pertama Anda

Setelah mengaturnya AWS CLI, Anda siap membuat ember di Amazon S3. Setiap objek di Amazon S3 disimpan di bucket. Sebelum Anda dapat menyimpan data di Amazon S3, Anda harus membuat bucket.

> **Note**
>
> Anda tidak dikenakan biaya untuk membuat bucket. Anda hanya dikenakan untuk menyimpan objek di dalam bucket dan untuk memindahkan objek masuk dan keluar dari bucket. Biaya yang Anda tanggung melalui mengikuti contoh dalam panduan ini adalah minimal (kurang dari $1). Untuk informasi selengkapnya tentang biaya penyimpanan, lihat Harga Amazon S3.

Untuk membuat bucket

1. Buat ember menggunakan `s3api create-bucket` perintah. Ganti *`amzn-s3-demo-bucket`* dengan nama bucket unik dan `us-east-1` dengan Wilayah yang Anda inginkan:

```
aws s3api create-bucket --bucket amzn-s3-demo-bucket --region us-east-1
```

Untuk Wilayah selain us-east-1, Anda perlu menentukan batasan lokasi:

```
aws s3api create-bucket --bucket amzn-s3-demo-bucket --region us-west-2 --create-
bucket-configuration LocationConstraint=us-west-2
```

> **Note**
>
> - Setelah membuat bucket, Anda tidak dapat mengubah Region.
> - Untuk meminimalkan latensi dan biaya serta memenuhi persyaratan regulasi, pilih Wilayah yang dekat dengan Anda. Objek yang disimpan di Wilayah tidak pernah keluar dari Wilayah kecuali Anda secara tegas mentransfer atau mereplikasinya ke Wilayah lain. Untuk daftar AWS Wilayah Amazon S3, lihat titik akhir AWS layanan di. Referensi Umum Amazon Web Services
> - Nama bucket harus:
>   - Unik dalam partisi. Partisi adalah pengelompokan Wilayah. AWS Saat ini memiliki tiga partisi: `aws` (Kawasan komersial), `aws-cn` (Wilayah Tiongkok), dan `aws-us-gov` (AWS GovCloud (US) Wilayah).
>   - Panjangnya antara 3 hingga 63 karakter.
>   - Hanya terdiri dari huruf kecil, angka, titik (.), dan tanda hubung (-). Untuk kompatibilitas terbaik, sebaiknya hindari penggunaan periode (.) dalam nama bucket, kecuali bucket yang hanya digunakan untuk hosting situs web statis.
>   - Dimulai dan diakhiri dengan huruf atau angka.
> - Setelah membuat bucket, Anda tidak dapat mengubah namanya.
> - Jangan sertakan informasi sensitif dalam nama bucket. Nama ember terlihat di titik URLs itu ke objek di ember.

2. Verifikasi bahwa bucket Anda dibuat dengan mencantumkan semua bucket:

```
aws s3 ls
```

3. Untuk Kepemilikan Objek, Anda dapat menonaktifkan atau mengaktifkan ACLs dan mengontrol kepemilikan objek yang diunggah ke bucket.

ACLs dinonaktifkan

Untuk menyetel pemilik Bucket diberlakukan (default) — ACLs dinonaktifkan, dan pemilik bucket secara otomatis memiliki dan memiliki kontrol penuh atas setiap objek di bucket tujuan umum:

```
aws s3api put-bucket-ownership-controls --bucket amzn-s3-demo-bucket --ownership-
controls="Rules=[{ObjectOwnership=BucketOwnerEnforced}]"
```

> **Note**
>
> Secara default, ACLs dinonaktifkan. Mayoritas kasus penggunaan modern di Amazon S3 tidak lagi memerlukan penggunaan. ACLs Kami menyarankan agar Anda tetap ACLs dinonaktifkan, kecuali dalam keadaan yang tidak biasa di mana Anda harus mengontrol akses untuk setiap objek secara individual. Untuk informasi selengkapnya, lihat Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.

ACLs diaktifkan

- Untuk mengatur lebih disukai pemilik Bucket — Pemilik bucket memiliki dan memiliki kontrol penuh atas objek baru yang ditulis akun lain ke ember dengan ACL `bucket-owner-full-control` kalengan:

  ```
  aws s3api put-bucket-ownership-controls --bucket amzn-s3-demo-bucket --ownership-
  controls="Rules=[{ObjectOwnership=BucketOwnerPreferred}]"
  ```

  Jika Anda menerapkan setelan pilihan pemilik Bucket dan ingin meminta semua unggahan Amazon S3 menyertakan ACL yang `bucket-owner-full-control` dikalengkan, Anda hanya dapat mengizinkan unggahan objek Memerlukan ACL bucket-owner-full-control kalengan untuk operasi Amazon PUT S3 (lebih disukai pemilik ember) yang menggunakan ACL ini.
- Untuk mengatur Object writer — AWS Akun yang mengunggah objek memiliki objek, memiliki kontrol penuh atas objek tersebut, dan dapat memberikan akses kepada pengguna lain melalui: ACLs

```
aws s3api put-bucket-ownership-controls --bucket amzn-s3-demo-bucket --ownership-
controls="Rules=[{ObjectOwnership=ObjectWriter}]"
```

> **Note**
>
> Pengaturan default-nya adalah Pemilik Bucket yang diberlakukan. Untuk menerapkan pengaturan default dan tetap ACLs dinonaktifkan, hanya `s3:CreateBucket` izin yang diperlukan. Untuk mengaktifkan ACLs, Anda harus memiliki `s3:PutBucketOwnershipControls` izin.

Untuk memeriksa setelan Kepemilikan Objek saat ini untuk bucket Anda:

```
aws s3api get-bucket-ownership-controls --bucket amzn-s3-demo-bucket
```

4. Untuk memverifikasi bahwa Blokir Akses Publik diaktifkan (diaktifkan secara default untuk bucket baru):

```
aws s3api get-public-access-block --bucket amzn-s3-demo-bucket
```

Secara default, keempat pengaturan Blokir Akses Publik diaktifkan untuk bucket baru. Kami menyarankan Anda tetap mengaktifkan semua pengaturan, kecuali Anda tahu bahwa Anda perlu menonaktifkan satu atau beberapa pengaturan untuk kasus penggunaan spesifik Anda. Untuk informasi lebih lanjut tentang pemblokiran akses publik, lihat Melakukan blok akses publik ke penyimpanan Amazon S3 Anda.

Jika Anda perlu mengaktifkan Blokir Akses Publik, gunakan perintah berikut:

```
aws s3api put-public-access-block --bucket amzn-s3-demo-bucket --public-access-
block-configuration
 "BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPublicBuckets=t
```

> **Note**
>
> Untuk mengaktifkan semua pengaturan Blokir Akses Publik, hanya izin `s3:CreateBucket` yang diperlukan. Untuk mematikan pengaturan Blokir Akses Publik, Anda harus memiliki izin `s3:PutBucketPublicAccessBlock`.

5. Untuk mengaktifkan pembuatan versi bucket Anda:

```
aws s3api put-bucket-versioning --bucket amzn-s3-demo-bucket --versioning-
configuration Status=Enabled
```

Secara default, Bucket Versioning dinonaktifkan. Penentuan Versi adalah cara menyimpan beberapa varian objek dalam bucket yang sama. Anda dapat menggunakan Penentuan Versi untuk menyimpan, mengambil, dan memulihkan setiap versi dari setiap objek yang disimpan dalam bucket Anda. Dengan versioning, Anda dapat lebih mudah memulihkan dari tindakan pengguna yang tidak diinginkan dan kegagalan aplikasi. Untuk informasi selengkapnya tentang penentuan versi, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.

6. Amazon S3 Object Lock membantu melindungi objek baru agar tidak dihapus atau ditimpa. Untuk informasi selengkapnya, lihat Mengunci objek dengan Object Lock. Untuk mengaktifkan Mengunci objek dengan Object Lock (memerlukan pembuatan versi bucket):

Untuk ember baru:

```
aws s3api create-bucket --bucket amzn-s3-demo-bucket --region us-east-1 --object-
lock-enabled-for-bucket
```

Untuk ember yang ada:

```
aws s3api put-object-lock-configuration --bucket amzn-s3-demo-bucket --object-lock-
configuration '{"ObjectLockEnabled": "Enabled"}'
```

Jika Anda ingin mengatur default Mengunci objek dengan Object Lock bersama dengan mengaktifkan Object Lock, Anda dapat menggunakan:

```
aws s3api put-object-lock-configuration --bucket amzn-s3-demo-bucket --object-
lock-configuration '{"ObjectLockEnabled":"Enabled","Rule":{"DefaultRetention":
{"Mode":"COMPLIANCE","Days":30}}}'
```

Anda dapat mengganti `"COMPLIANCE"` dengan `"GOVERNANCE"` untuk mode yang tidak terlalu membatasi, dan menyesuaikan jumlah hari sesuai kebutuhan.

> **Note**
>
> Untuk membuat bucket yang diaktifkan Object Lock, Anda harus memiliki izin berikut:`s3:CreateBucket`,`s3:PutBucketVersioning`, dans`3:PutBucketObjectLockConfiguration`.

7. Anda dapat menambahkan tag ke ember Anda. Dengan alokasi AWS biaya, Anda dapat menggunakan tag bucket untuk membubuhi keterangan penagihan penggunaan bucket. Tanda adalah pasangan nilai kunci yang mewakili label yang Anda tetapkan ke bucket. Untuk informasi selengkapnya, lihat Menggunakan tanda alokasi biaya bucket S3.

   Untuk menambahkan tag ke bucket Anda:

   ```
   aws s3api put-bucket-tagging --bucket amzn-s3-demo-bucket --tagging
    'TagSet=[{Key=Purpose,Value=Testing},{Key=Environment,Value=Development}]'
   ```

8. Bucket dan objek baru dienkripsi dengan menggunakan enkripsi sisi server dengan kunci terkelola Amazon S3 `SSE-S3` () sebagai tingkat dasar konfigurasi enkripsi. Untuk memverifikasi enkripsi default bucket Anda, gunakan perintah berikut:

   ```
   aws s3api get-bucket-encryption --bucket amzn-s3-demo-bucket
   ```

   Anda juga dapat mengonfigurasi enkripsi sisi server dengan AWS KMS kunci (`SSE-KMS`) dan enkripsi sisi server dual-layer dengan keys () untuk bucket Anda. AWS KMS `DSSE-KMS` Kunci AWS terkelola (`aws/s3`) dan kunci terkelola pelanggan Anda dapat digunakan sebagai AWS KMS kunci untuk `SSE-KMS` dan konfigurasi `DSSE-KMS` enkripsi Anda. Untuk informasi selengkapnya tentang kunci yang dikelola pelanggan, lihat Kunci dan AWS kunci pelanggan di Panduan AWS Key Management Service Pengembang. Untuk informasi selengkapnya tentang membuat AWS KMS kunci, lihat Membuat kunci di Panduan AWS Key Management Service Pengembang.

> **Important**
>
> AWS KMS Kuncinya harus berada di AWS Wilayah yang sama dengan bucket Amazon S3 Anda. Kunci KMS lintas wilayah tidak didukung untuk enkripsi bucket Amazon S3.

Saat mengonfigurasi bucket untuk menggunakan enkripsi default dengan SSE-KMS, Anda juga dapat menggunakan Kunci Bucket Amazon S3. Amazon S3 Bucket Keys menurunkan biaya enkripsi dengan mengurangi lalu lintas permintaan dari Amazon S3 ke. AWS KMS Untuk informasi selengkapnya, lihat Mengurangi biaya SSE-KMS dengan Kunci Bucket Amazon S3. Kunci Bucket Amazon S3 tidak didukung untuk DSSE-KMS. Di dalam AWS CLI, Amazon S3 Bucket Keys TIDAK diaktifkan secara default saat membuat bucket baru. Ini berbeda dari perilaku konsol di mana mereka diaktifkan secara default.

Untuk mengonfigurasi `SSE-KMS` dan mengaktifkan Tombol Bucket Amazon S3:

```
aws s3api put-bucket-encryption --bucket amzn-s3-demo-bucket --server-side-
encryption-configuration "{\"Rules\":[{\"ApplyServerSideEncryptionByDefault
\":{\"SSEAlgorithm\":\"aws:kms\",\"KMSMasterKeyID\":\"YOUR-KMS-KEY-ARN\"},
\"BucketKeyEnabled\":true}]}"
```

Untuk memeriksa apakah Kunci Bucket Amazon S3 diaktifkan untuk bucket:

```
aws s3api get-bucket-encryption --bucket amzn-s3-demo-bucket
```

Output akan mencakup `BucketKeyEnabled` bidang yang disetel ke salah satu `true` atau`false`.

Untuk mengkonfigurasi`DSSE-KMS`, gunakan perintah berikut:

```
aws s3api put-bucket-encryption --bucket amzn-s3-demo-bucket --server-side-
encryption-configuration '{"Rules":[{"ApplyServerSideEncryptionByDefault":
{"SSEAlgorithm":"aws:kms:dsse","KMSMasterKeyID":"YOUR-KMS-KEY-ARN"}}]}'
```

Untuk informasi selengkapnya tentang enkripsi default, lihat Mengatur perilaku enkripsi sisi server default untuk bucket Amazon S3. Untuk informasi selengkapnya tentang SSE-S3, lihat. Menggunakan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3)

> **Important**
>
> Jika Anda menggunakan opsi SSE-KMS atau DSSE-KMS untuk konfigurasi enkripsi default Anda, Anda tunduk pada kuota permintaan per detik (RPS) sebesar. AWS KMS Anda dapat mengurangi panggilan API KMS dengan mengaktifkan Amazon S3 Bucket Keys, yang mengurangi jumlah permintaan yang dikirim. AWS KMS Untuk informasi selengkapnya tentang kuota AWS KMS dan cara meminta kenaikan kuota, lihat Kuota di Panduan Pengembang AWS Key Management Service .

Anda telah membuat bucket di Amazon S3. Langkah selanjutnya adalah mengunggah objek ke bucket Anda.

Langkah 2: Unggah objek ke bucket Anda

Setelah membuat bucket di Amazon S3, Anda siap untuk mengunggah objek ke bucket. Objek dapat berupa jenis file apa pun: file teks, foto, video, dan sebagainya.

Untuk mengunggah objek ke bucket

1. Buat file teks sederhana untuk diunggah. Anda dapat menggunakan editor teks apa pun atau menjalankan perintah berikut:

   ```
   echo 'Hello, Amazon S3!' > example.txt
   ```

2. Unggah file ke bucket Anda menggunakan perintah s3 cp:

   ```
   aws s3 cp example.txt s3://amzn-s3-demo-bucket/
   ```

   Jika unggahan berhasil, Anda akan melihat output yang mirip dengan:

   ```
   upload: ./example.txt to s3://amzn-s3-demo-bucket/example.txt
   ```

3. Verifikasi bahwa objek telah diunggah dengan mencantumkan konten bucket Anda:

   ```
   aws s3 ls s3://amzn-s3-demo-bucket/
   ```

Anda berhasil mengunggah objek ke bucket Anda. Langkah selanjutnya adalah mengunduh objek.

Langkah 3: Mengunduh objek

Setelah mengunggah sebuah objek ke bucket, Anda dapat melihat informasi tentang objek tersebut dan mengunduhnya ke komputer lokal.

Untuk mengunduh objek dari bucket Amazon S3

1. Untuk mendapatkan informasi tentang objek Anda:

   ```
   aws s3api head-object --bucket amzn-s3-demo-bucket --key example.txt
   ```

   Perintah ini mengembalikan metadata tentang objek, termasuk jenis konten, panjang konten, dan tanggal modifikasi terakhir.
2. Unduh objek ke komputer lokal Anda:

   ```
   aws s3 cp s3://amzn-s3-demo-bucket/example.txt downloaded-example.txt
   ```

   Jika unduhan berhasil, Anda akan melihat output yang mirip dengan:

   ```
   download: s3://amzn-s3-demo-bucket/example.txt to ./downloaded-example.txt
   ```
3. Verifikasi isi file yang diunduh:

   ```
   cat downloaded-example.txt
   ```

> **Note**
>
> - Berbeda dengan konsol, AWS CLI dapat mengunduh beberapa objek sekaligus menggunakan wildcard atau `--recursive` bendera.
> - Saat mengunduh objek dengan AWS CLI, periode (.) di akhir nama kunci objek dipertahankan, tidak seperti di konsol tempat mereka dihapus. Ini penting jika kunci objek Anda diakhiri dengan titik.

Contoh mengunduh beberapa objek:

Untuk mengunduh beberapa objek dari bucket Amazon S3 dengan ekstensi file tertentu, gunakan perintah salin rekursif dengan `exclude` dan `include` filter seperti yang ditunjukkan pada contoh.

```
aws s3 cp s3://amzn-s3-demo-bucket/ . --recursive --exclude "*" --include "*.txt"
```

Anda berhasil mengunduh objek. Langkah selanjutnya adalah menyalin objek Anda ke folder.

Langkah 4: Salin objek ke folder

Anda sudah menambahkan objek ke dalam bucket dan mengunduh objek. Sekarang, Anda membuat folder dan menyalin objek ke folder.

Untuk menyalin objek ke folder

1. Di Amazon S3, folder diwakili oleh awalan dalam kunci objek. Buat “folder” dengan menyalin objek dengan awalan:

   ```
   aws s3 cp s3://amzn-s3-demo-source-bucket/example.txt s3://amzn-s3-demo-destination-bucket/favorite-files/example.txt
   ```

   Jika salinan berhasil, Anda akan melihat output yang mirip dengan:

   ```
   copy: s3://amzn-s3-demo-source-bucket/example.txt to s3://amzn-s3-demo-destination-bucket/favorite-files/example.txt
   ```

2. Verifikasi bahwa objek telah disalin dengan mencantumkan isi folder:

   ```
   aws s3 ls s3://amzn-s3-demo-destination-bucket/favorite-files/
   ```

Anda telah berhasil menyalin objek Anda ke folder. Langkah selanjutnya adalah menghapus objek dan ember Anda.

Langkah 5: Hapus objek dan bucket Anda

Saat Anda tidak lagi memerlukan objek atau bucket, kami menyarankan Anda menghapusnya untuk mencegah biaya lebih lanjut. Jika Anda menyelesaikan panduan memulai ini sebagai latihan pembelajaran, dan Anda tidak berencana untuk menggunakan bucket atau objek Anda, kami sarankan Anda menghapus bucket dan objek Anda sehingga biaya tidak lagi bertambah.

Sebelum menghapus bucket, kosongkan bucket atau hapus objek di dalam bucket. Setelah Anda menghapus objek dan bucket Anda, objek tersebut tidak lagi tersedia.

Jika Anda ingin terus menggunakan nama bucket yang sama, sebaiknya hapus objek atau kosongkan bucket, tetapi jangan menghapus bucket. Setelah Anda menghapus bucket, nama akan tersedia untuk digunakan kembali. Namun, AWS akun lain mungkin membuat ember dengan nama yang sama sebelum Anda memiliki kesempatan untuk menggunakannya kembali.

Menghapus objek

Jika ingin memilih objek mana yang Anda hapus tanpa mengosongkan semua objek dari bucket, Anda dapat menghapus objek.

Hapus objek tertentu:

```
aws s3 rm s3://amzn-s3-demo-bucket/example.txt
```

Jika penghapusan berhasil, Anda akan melihat output yang mirip dengan:

```
delete: s3://amzn-s3-demo-bucket/example.txt
```

Mengosongkan bucket Anda

Jika Anda berencana untuk menghapus bucket, Anda harus terlebih dahulu mengosongkan bucket Anda, yang menghapus semua objek, versi, dan menghapus spidol di bucket.

Mengosongkan bucket

> **Important**
>
> Mengosongkan bucket tidak dapat diurungkan. Objek yang ditambahkan ke bucket saat tindakan bucket kosong sedang berlangsung akan dihapus.

1. Opsi 1: Untuk ember yang lebih kecil, gunakan `rm` perintah dengan `--recursive` bendera untuk menghapus semua objek di ember:

   ```
   aws s3 rm s3://amzn-s3-demo-bucket --recursive
   ```

Perintah ini menghapus semua objek di bucket, termasuk objek dalam folder.

> **Note**
>
> Jika bucket Anda berisi banyak objek atau benda besar, perintah ini mungkin habis. Untuk bucket dengan data dalam jumlah besar, gunakan aturan Siklus Hidup Amazon S3 untuk menghapus objek dalam bucket.

Opsi 2: Gunakan aturan Siklus Hidup Amazon S3 (disarankan untuk ember besar)

Untuk bucket dengan banyak objek atau objek besar, gunakan aturan Siklus Hidup Amazon S3 untuk secara otomatis kedaluwarsa dan menghapus semua objek. Tunggu hingga aturan siklus hidup diproses (ini mungkin memakan waktu hingga 24 jam). Untuk informasi selengkapnya tentang penggunaan aturan siklus hidup untuk mengosongkan bucket, lihat Bagaimana cara mengosongkan bucket Amazon S3 menggunakan aturan konfigurasi siklus hidup? .

2. Verifikasi bahwa ember kosong:

```
aws s3 ls s3://amzn-s3-demo-bucket
```

3. Jika bucket Anda mengaktifkan versi, gunakan perintah berikut untuk menghapus objek berversi dan menghapus penanda.

   Hapus objek berversi:

```
aws s3api delete-objects --bucket amzn-s3-demo-bucket --delete "$(aws s3api list-
object-versions --bucket amzn-s3-demo-bucket --output json --query='{Objects:
 Versions[].{Key:Key,VersionId:VersionId}}')"
```

   Hapus penanda hapus:

```
aws s3api delete-objects --bucket amzn-s3-demo-bucket --delete "$(aws s3api list-
object-versions --bucket amzn-s3-demo-bucket --output json --query='{Objects:
 DeleteMarkers[].{Key:Key,VersionId:VersionId}}')"
```

4. Verifikasi bahwa bucket kosong dari semua versi objek dan hapus penanda:

```
aws s3api list-object-versions --bucket amzn-s3-demo-bucket
```

Output seharusnya tidak menunjukkan versi atau menghapus penanda yang tersisa.

Menghapus bucket Anda

Setelah mengosongkan bucket atau menghapus semua objek dari bucket, Anda dapat menghapus bucket.

> **Important**
>
> Menghapus ember tidak dapat dibatalkan. Nama bucket bersifat unik. Jika Anda menghapus bucket, AWS pengguna lain dapat menggunakan nama tersebut. Jika Anda ingin terus menggunakan nama bucket yang sama, jangan menghapus bucket Anda. Sebaliknya, kosongkan dan simpan bucket.

Untuk menghapus ember Anda

1. Hapus ember Anda:

```
aws s3api delete-bucket --bucket amzn-s3-demo-bucket
```

2. Verifikasi bahwa bucket telah dihapus dengan mencantumkan semua bucket Anda:

```
aws s3 ls
```

Langkah selanjutnya

Dalam contoh sebelumnya, Anda belajar cara melakukan beberapa tugas dasar Amazon S3 menggunakan. AWS CLI

Topik-topik berikut ini menjelaskan jalur pembelajaran yang dapat Anda gunakan untuk mendapatkan pemahaman yang lebih mendalam tentang Amazon S3 sehingga Anda dapat mengimplementasikannya di aplikasi Anda.

- Memahami kasus penggunaan umum
- Mengontrol akses ke bucket dan objek Anda
- Lindungi dan pantau penyimpanan Anda

- Mengembangkan dengan Amazon S3
- Memahami kasus penggunaan umum
- Menjelajahi pelatihan dan dukungan

Daftar berikut menunjukkan AWS CLI perintah umum untuk Amazon S3:

- cp - Menyalin file atau objek antara sistem file lokal Anda dan Amazon S3, atau antara lokasi Amazon S3
- ls - Daftar objek Amazon S3 dan awalan umum di bawah bucket dan awalan yang ditentukan
- mb - Membuat ember Amazon S3
- mv - Memindahkan file atau objek antara sistem file lokal Anda dan Amazon S3, atau antara lokasi Amazon S3
- presign - Menghasilkan URL yang telah ditandatangani sebelumnya untuk objek Amazon S3 yang memungkinkan akses sementara tanpa kredensyal AWS
- rb - Menghapus ember Amazon S3 kosong. Anda dapat menggunakan `--force` bendera untuk secara otomatis mengosongkan dan menghapus ember dengan konten dalam satu perintah. Tindakan ini tidak dapat dibatalkan.
- rm - Menghapus objek dari Amazon S3
- sinkronisasi - Menyinkronkan direktori dan awalan Amazon S3 dengan menyalin file baru dan diperbarui secara rekursif dari direktori sumber ke tujuan.
- situs web - Mengkonfigurasi ember sebagai situs web statis

Untuk informasi selengkapnya tentang AWS CLI perintah untuk Amazon S3, lihat sumber daya berikut:

- s3 - Perintah Amazon S3 tingkat tinggi yang menyederhanakan operasi umum
- s3api - Akses langsung ke semua operasi API Amazon S3
- s3control - Akses langsung ke semua operasi API Kontrol Amazon S3

Membuat, mengonfigurasi, dan bekerja dengan bucket tujuan umum Amazon S3

Untuk menyimpan data Anda di Amazon S3, Anda bekerja dengan sumber daya yang dikenal sebagai bucket dan objek. Bucket adalah kontainer untuk objek. Objek adalah file data dan metadata apa pun yang mendeskripsikan file tersebut.

Untuk menyimpan objek di Amazon S3, Anda membuat bucket dan kemudian mengunggah objek tersebut ke dalam bucket. Bila objek ada di dalam bucket, Anda bisa membuka, mengunduh, dan memindahkannya. Saat Anda tidak lagi memerlukan objek atau bucket, Anda dapat membersihkan sumber daya Anda.

Topik di bagian ini memberikan ikhtisar tentang bekerja dengan bucket tujuan umum di Amazon S3. Mereka termasuk informasi tentang penamaan, pembuatan, akses, dan penghapusan bucket tujuan umum. Untuk informasi selengkapnya tentang melihat atau mencantumkan objek dalam bucket, lihat Mengatur, membuat daftar, dan bekerja dengan objek Anda.

Ada beberapa jenis bucket Amazon S3. Sebelum membuat bucket, pastikan Anda memilih jenis bucket yang paling sesuai dengan aplikasi dan persyaratan kinerja Anda. Untuk informasi lebih lanjut tentang berbagai jenis bucket dan kasus penggunaan yang sesuai untuk masing-masing, lihatBucket.

> **Note**
>
> Untuk informasi selengkapnya tentang menggunakan kelas penyimpanan Amazon S3 Express One Zone dengan bucket direktori, lihat S3 Express One Zone dan Bekerja dengan bucket direktori.

> **Note**
>
> Dengan Amazon S3, Anda hanya membayar sesuai penggunaan Anda. Untuk informasi selengkapnya tentang fitur dan harga Amazon S3, lihat Amazon S3. Jika Anda adalah pelanggan baru Amazon S3, Anda dapat memulai Amazon S3 secara gratis. Untuk informasi selengkapnya, lihat AWS Tingkat Gratis.

Topik

- Ikhtisar ember tujuan umum
- Pola bucket tujuan umum umum untuk membangun aplikasi di Amazon S3
- Aturan penamaan bucket tujuan umum
- Kuota, batasan, dan batasan bucket tujuan umum
- Mengakses bucket tujuan umum Amazon S3
- Membuat ember tujuan umum
- Melihat properti untuk bucket tujuan umum S3
- Daftar ember tujuan umum Amazon S3
- Mengosongkan ember tujuan umum
- Menghapus bucket tujuan umum
- Pasang bucket Amazon S3 sebagai sistem file lokal
- Bekerja dengan Browser Penyimpanan untuk Amazon S3
- Mengonfigurasi transfer file yang cepat dan aman menggunakan Amazon S3 Transfer Acceleration
- Menggunakan Requester Pays ember tujuan umum untuk transfer dan penggunaan penyimpanan

Ikhtisar ember tujuan umum

Untuk mengunggah data Anda (foto, video, dokumen, dll.) ke Amazon S3, Anda harus terlebih dahulu membuat bucket S3 di salah satu Wilayah AWS.

Ada beberapa jenis bucket Amazon S3. Sebelum membuat bucket, pastikan Anda memilih jenis bucket yang paling sesuai dengan aplikasi dan persyaratan kinerja Anda. Untuk informasi lebih lanjut tentang berbagai jenis bucket dan kasus penggunaan yang sesuai untuk masing-masing, lihatBucket.

Bagian berikut memberikan informasi selengkapnya tentang bucket tujuan umum, termasuk aturan penamaan bucket, kuota, dan detail konfigurasi bucket. Untuk daftar pembatasan dan batasan yang terkait dengan bucket Amazon S3, lihat,. Kuota, batasan, dan batasan bucket tujuan umum

Topik

- Ikhtisar ember tujuan umum
- Pola ember tujuan umum umum
- Izin
- Mengelola akses publik ke ember tujuan umum
- opsi konfigurasi bucket tujuan umum

- operasi bucket tujuan umum
- pemantauan kinerja bucket tujuan umum

Ikhtisar ember tujuan umum

Setiap objek dimuat dalam bucket. Misalnya, jika objek bernama `photos/puppy.jpg` disimpan dalam bucket tujuan `amzn-s3-demo-bucket` umum di Wilayah AS Barat (Oregon), maka objek tersebut dapat dialamatkan dengan menggunakan URL. `https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/photos/puppy.jpg` Untuk informasi lebih lanjut, lihat Mengakses Bucket.

- Kuota bucket tujuan umum untuk Wilayah komersial hanya dapat dilihat dan dikelola dari US East (Virginia N.).
- Kuota bucket tujuan umum hanya AWS GovCloud (US) dapat dilihat dan dikelola dari AWS GovCloud (AS-Barat).

Dalam hal implementasi, bucket dan objek adalah AWS sumber daya, dan Amazon S3 APIs menyediakan bagi Anda untuk mengelolanya. Misalnya, Anda dapat membuat bucket serta mengunggah objek menggunakan API Amazon S3. Anda juga dapat menggunakan konsol Amazon S3 untuk melakukan operasi ini. Konsol menggunakan Amazon S3 APIs untuk mengirim permintaan ke Amazon S3.

Bagian ini menjelaskan cara bekerja dengan ember tujuan umum. Untuk informasi tentang bekerja dengan objek, lihat Gambaran umum objek Amazon S3.

Amazon S3 mendukung bucket tujuan umum global, yang berarti bahwa setiap nama bucket harus unik di semua bagian Akun AWS dalam partisi Wilayah AWS . Partisi adalah pengelompokan Wilayah. AWS saat ini memiliki tiga partisi: `aws` (Wilayah Standar), `aws-cn` (Wilayah Tiongkok), dan `aws-us-gov` (AWS GovCloud (US)).

Setelah bucket tujuan umum dibuat, nama bucket tersebut tidak dapat digunakan oleh yang lain Akun AWS di partisi yang sama sampai bucket dihapus. Anda tidak boleh bergantung pada konvensi penamaan bucket kustom untuk tujuan ketersediaan, atau verifikasi keamanan. Untuk panduan penamaan bucket, lihat Aturan penamaan bucket tujuan umum.

Amazon S3 membuat bucket di Wilayah yang Anda tentukan. Untuk mengurangi latensi, meminimalkan biaya, atau memenuhi persyaratan peraturan, pilih Wilayah AWS yang secara geografis dekat dengan Anda. Misalnya, jika Anda tinggal di Eropa, Anda mungkin akan

mendapatkan keuntungan jika membuat bucket di Wilayah Eropa (Irlandia) atau Eropa (Frankfurt). Untuk daftar semua Wilayah dan titik akhir Amazon S3, lihat Wilayah dan Titik Akhir dalam AWS Referensi Umum.

> **Note**
>
> Objek yang termasuk dalam bucket yang Anda buat di spesifik Wilayah AWS tidak pernah meninggalkan Wilayah itu, kecuali jika Anda secara eksplisit mentransfernya ke Wilayah lain. Misalnya, objek yang disimpan di Wilayah Eropa (Irlandia) tidak pernah keluar dari Wilayah tersebut.

Pola ember tujuan umum umum

Saat membuat aplikasi di Amazon S3, Anda dapat menggunakan bucket tujuan umum yang unik untuk memisahkan kumpulan data atau beban kerja yang berbeda. Tergantung pada kasus penggunaan Anda, ada pola desain yang berbeda dan praktik terbaik untuk menggunakan ember tujuan umum. Untuk informasi selengkapnya, lihat Pola bucket tujuan umum umum untuk membangun aplikasi di Amazon S3.

Izin

Anda dapat menggunakan Pengguna root akun AWS kredensil untuk membuat bucket tujuan umum dan melakukan operasi Amazon S3 lainnya. Namun, kami tidak merekomendasikan penggunaan kredensial pengguna root dari Akun AWS Anda untuk membuat permintaan seperti membuat bucket. Sebagai gantinya, buat pengguna AWS Identity and Access Management (IAM) dan berikan akses penuh kepada pengguna tersebut (pengguna secara default tidak memiliki izin).

Pengguna ini disebut sebagai administrator. Anda dapat menggunakan kredensil pengguna administrator, bukan kredenal pengguna root akun Anda, untuk berinteraksi AWS dan melakukan tugas, seperti membuat bucket, membuat pengguna, dan memberi mereka izin.

Untuk informasi lebih lanjut, lihat kredensial Pengguna root akun AWS dan kredensial pengguna IAM dalam AWS Referensi Umum dan Praktik terbaik keamanan di IAM dalam Panduan Pengguna IAM.

Akun AWS Yang menciptakan sumber daya memiliki sumber daya itu. Misalnya, jika Anda membuat pengguna IAM di dalam Akun AWS dan memberikan izin kepada pengguna untuk membuat bucket, pengguna dapat membuat bucket. Namun, pengguna tidak memiliki bucket tersebut; Akun AWS yang memiliki pengguna itulah pemilik bucket-nya. Pengguna memerlukan izin tambahan dari pemilik

sumber daya untuk melakukan operasi bucket lainnya. Untuk informasi selengkapnya tentang mengelola izin untuk sumber daya Amazon S3, lihat Identity and Access Management untuk Amazon S3.

Mengelola akses publik ke ember tujuan umum

Akses publik diberikan ke bucket dan objek tujuan umum melalui kebijakan bucket, daftar kontrol akses (ACLs), atau keduanya. Untuk membantu Anda mengelola akses publik ke sumber daya Amazon S3, Amazon S3 menyediakan pengaturan untuk memblokir akses publik. Setelan Akses Publik Blok Amazon S3 dapat mengganti ACLs dan menyusun kebijakan sehingga Anda dapat menerapkan batasan seragam pada akses publik ke sumber daya ini. Anda dapat menerapkan pengaturan Blokir Akses Publik ke bucket individu, atau semua bucket dalam akun Anda.

Untuk memastikan bahwa semua bucket dan objek tujuan umum Amazon S3 Anda diblokir akses publiknya, keempat pengaturan untuk Blokir Akses Publik diaktifkan secara default saat Anda membuat bucket baru. Kami merekomendasikan agar Anda mengaktifkan keempat pengaturan untuk Blokir Akses Publik untuk akun Anda juga. Pengaturan ini memblokir semua akses publik untuk semua bucket saat ini, dan yang akan datang.

Sebelum menerapkan pengaturan ini, verifikasi bahwa aplikasi Anda akan bekerja dengan baik tanpa akses publik. Jika Anda memerlukan beberapa tingkat akses publik ke bucket atau objek Anda—misalnya, untuk menjadi host situs web statis seperti yang dijelaskan di Hosting situs web statis menggunakan Amazon S3—Anda dapat menyesuaikan pengaturan individual agar sesuai dengan kasus penggunaan penyimpanan Anda. Untuk informasi selengkapnya, lihat Melakukan blok akses publik ke penyimpanan Amazon S3 Anda.

Namun, kami menyarankan agar Anda membiarkan Blokir Akses Publik diaktifkan. Jika Anda ingin mengaktifkan keempat pengaturan Blokir Akses Publik dan meng-host situs web statis, Anda dapat menggunakan Amazon CloudFront Origin Access Control (OAC). Amazon CloudFront menyediakan kemampuan yang diperlukan untuk menyiapkan situs web statis yang aman. Situs web statis Amazon S3 hanya mendukung titik akhir HTTP. Amazon CloudFront menggunakan penyimpanan Amazon S3 yang tahan lama sambil menyediakan header keamanan tambahan, seperti HTTPS. HTTPS menambahkan keamanan dengan mengenkripsi permintaan HTTP normal, dan melindungi dari serangan siber umum.

Untuk informasi selengkapnya, lihat Memulai situs web statis aman di Panduan CloudFront Pengembang Amazon.

> **Note**
>
> Jika Anda melihat `Error` kapan Anda mencantumkan bucket tujuan umum dan setelan akses publiknya, Anda mungkin tidak memiliki izin yang diperlukan. Pastikan bahwa Anda menambahkan izin berikut ke kebijakan pengguna atau peran Anda:
>
> ```
> s3:GetAccountPublicAccessBlock
> s3:GetBucketPublicAccessBlock
> s3:GetBucketPolicyStatus
> s3:GetBucketLocation
> s3:GetBucketAcl
> s3:ListAccessPoints
> s3:ListAllMyBuckets
> ```
>
> Dalam beberapa kasus yang jarang terjadi, permintaan juga dapat gagal karena Wilayah AWS gangguan.

opsi konfigurasi bucket tujuan umum

Amazon S3 mendukung berbagai opsi bagi Anda untuk mengonfigurasi bucket tujuan umum Anda. Misalnya, Anda dapat mengonfigurasi bucket untuk hosting situs web, menambahkan konfigurasi untuk mengelola siklus hidup objek di bucket, dan mengonfigurasi bucket untuk mencatat semua akses ke bucket. Amazon S3 mendukung sub-sumber daya untuk menyimpan dan mengelola informasi konfigurasi bucket. Anda dapat menggunakan API Amazon S3 untuk membuat dan mengelola sub-sumber daya ini. Namun, Anda juga dapat menggunakan konsol atau AWS SDKs.

> **Note**
>
> Terdapat konfigurasi tingkat objek juga. Misalnya, Anda dapat mengonfigurasi izin tingkat objek dengan mengonfigurasi daftar kontrol akses (ACL) kustom untuk objek tersebut.

Ini disebut sebagai sub-sumber daya karena ada dalam konteks bucket atau objek tertentu. Tabel berikut ini mencantumkan sub-sumber daya yang memungkinkan Anda mengelola konfigurasi kustom bucket.

Sub-sumber daya	Deskripsi
cors (berbagi sumber daya lintas asal)	Anda dapat mengonfigurasi bucket Anda untuk mengizinkan permintaan lintas asal. Untuk informasi selengkapnya, lihat Berbagi sumber daya lintas asal (CORS).
pemberitahuan peristiwa	Anda dapat mengaktifkan bucket Anda untuk mengirimkan pemberitahuan peristiwa bucket tertentu. Untuk informasi selengkapnya, lihat Notifikasi Peristiwa Amazon S3.
siklus hidup	Anda dapat menentukan aturan siklus aktif objek dalam bucket Anda, yang memiliki siklus aktif yang ditetapkan dengan baik. Misalnya, Anda dapat menetapkan aturan untuk mengarsipkan objek satu tahun setelah pembuatan , atau menghapus objek 10 tahun setelah pembuatan. Untuk informasi selengkapnya, lihat Mengelola siklus hidup objek.
lokasi	Saat membuat bucket, Anda menentukan Wilayah AWS tempat Amazon S3 yang Anda inginkan untuk membuat bucket. Amazon S3 menyimpan informasi ini di sub-sumber daya lokasi dan menyediakan API bagi Anda untuk mengambil informasi ini.
pencatatan log	Pencatatan log memungkinkan Anda untuk melacak permintaan akses ke bucket Anda. Setiap catatan log akses memberikan perincian tentang permintaan akses tunggal, seperti pemohon, nama bucket, waktu permintaa n, tindakan permintaan, status respons, dan kode kesalahan, jika ada. Informasi log akses dapat berguna dalam audit keamanan dan akses. Ini juga dapat membantu Anda untuk mempelajari basis pelanggan Anda, serta memahami tagihan Amazon S3. Untuk informasi selengkapnya, lihat Pencatatan permintaan dengan pencatatan akses server.
penguncian objek	Untuk menggunakan Kunci Objek S3, Anda harus mengaktifkannya untuk bucket. Anda juga dapat secara opsional mengonfigurasi mode dan periode

Sub-sumber daya	Deskripsi
	retensi default yang berlaku untuk objek baru yang ditempatkan dalam bucket. Untuk informasi selengkapnya, lihat Mengunci objek dengan Object Lock.
kebijakan dan ACL (daftar kontrol akses)	Semua sumber daya Anda (seperti bucket dan objek) bersifat pribadi secara default. Amazon S3 mendukung opsi kebijakan bucket dan daftar kontrol akses (ACL) bagi Anda untuk memberikan dan mengelola izin tingkat bucket. Amazon S3 menyimpan informasi izin di sub-sumber daya kebijakan dan acl. Untuk informasi selengkapnya, lihat Identity and Access Management untuk Amazon S3.
replikasi	Replikasi adalah penyalinan objek secara otomatis dan asinkron di seluruh bucket di Wilayah AWS yang berbeda atau sama. Untuk informasi selengkap nya, lihat Mereplikasi objek di dalam dan di seluruh Wilayah.
requestPayment	Secara default, Akun AWS yang membuat bucket (pemilik bucket) membayar unduhan dari bucket. Dengan menggunakan sub-sumber daya ini, pemilik bucket dapat menetapkan bahwa orang yang meminta unduhan akan dikenakan biaya untuk pengunduhan. Amazon S3 menyediakan API bagi Anda untuk dapat mengelola sub-sumber daya ini. Untuk informasi selengkapnya, lihat Menggunakan Requester Pays ember tujuan umum untuk transfer dan penggunaan penyimpanan.
penandaan	Anda dapat menambahkan tag alokasi biaya ke bucket Anda untuk mengkategorikan dan melacak biaya Anda. AWS Amazon S3 menyediak an sub-sumber daya pemberian tag untuk menyimpan dan mengelola tag di bucket. Menggunakan tag yang Anda terapkan ke bucket Anda, AWS mengh asilkan laporan alokasi biaya dengan penggunaan dan biaya yang dikumpulk an oleh tanda Anda. Untuk informasi selengkapnya, lihat Pelaporan penagihan dan penggunaan untuk Amazon S3.

Sub-sumber daya	Deskripsi
akselerasi transfer	Akselerasi Transfer memungkinkan transfer file yang cepat, mudah, dan aman dalam jarak jauh antara klien Anda dan bucket S3. Transfer Accelerat ion memanfaatkan lokasi tepi Amazon yang didistribusikan secara global CloudFront. Untuk informasi selengkapnya, lihat Mengonfigurasi transfer file yang cepat dan aman menggunakan Amazon S3 Transfer Acceleration.
Penentuan Versi	Penentuan Versi membantu Anda memulihkan penghapusan dan penimpaan secara tidak sengaja. Kami merekomendasikan Penentuan Versi sebagai praktik terbaik untuk memulihkan objek agar tidak dihapus atau ditimpa secara tidak sengaja. Untuk informasi selengkapnya, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.
situs web	Anda dapat mengonfigurasi bucket untuk menghosting situs web statis. Amazon S3 menyimpan konfigurasi ini dengan membuat sub-sumber daya situs web. Untuk informasi selengkapnya, lihat Hosting situs web statis menggunakan Amazon S3.

operasi bucket tujuan umum

Rekayasa ketersediaan tinggi Amazon S3 berfokus pada operasi get, put, list, dan delete. Karena operasi bucket tujuan umum bekerja melawan ruang sumber daya global yang terpusat, sebaiknya jangan membuat, menghapus, atau mengonfigurasi bucket pada jalur kode ketersediaan tinggi aplikasi Anda. Lebih baik membuat, menghapus, atau mengonfigurasi bucket dalam inisialisasi atau rutinitas pengaturan terpisah yang jarang Anda jalankan.

pemantauan kinerja bucket tujuan umum

Ketika Anda memiliki aplikasi penting dan proses bisnis yang bergantung pada AWS sumber daya, penting untuk memantau dan mendapatkan peringatan untuk sistem Anda. Memantau data Anda

dapat membantu menjaga keandalan, ketersediaan, dan kinerja Amazon S3 dan solusi Anda AWS . Ada beberapa AWS layanan yang dapat Anda gunakan untuk mengumpulkan dan menggabungkan metrik dan log untuk bucket S3 Anda.

Bergantung pada kasus penggunaan Anda, Anda dapat memilih AWS layanan mana yang paling sesuai dengan kebutuhan organisasi Anda untuk men-debug masalah, memantau data Anda, mengoptimalkan biaya penyimpanan, atau memecahkan masalah multi-titik. Misalnya:

- Untuk meningkatkan kinerja aplikasi yang menggunakan S3: Siapkan CloudWatch alarm untuk memantau data penyimpanan, metrik replikasi, atau metrik permintaan Anda.
- Untuk merencanakan penggunaan penyimpanan, mengoptimalkan biaya penyimpanan, atau untuk mengetahui berapa banyak penyimpanan yang Anda miliki di seluruh organisasi Anda: Gunakan Lensa Penyimpanan Amazon S3. Atau, Anda dapat menggunakan S3 Storage Lens untuk meningkatkan kinerja data Anda dengan mengaktifkan metrik lanjutan dan menggunakan metrik kode status terperinci untuk mendapatkan hitungan permintaan yang berhasil atau gagal.
- Untuk tampilan terpadu kesehatan operasional Anda: Publikasikan metrik penggunaan dan aktivitas Lensa Penyimpanan S3 ke dasbor.Amazon CloudWatch

  > **Note**
  >
  > Opsi Amazon CloudWatch penerbitan tersedia untuk dasbor Lensa Penyimpanan S3 yang ditingkatkan ke metrik dan rekomendasi lanjutan. Anda dapat mengaktifkan opsi CloudWatch penerbitan untuk konfigurasi dasbor baru atau yang sudah ada di S3 Storage Lens.

- Untuk mendapatkan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan: Siapkan AWS CloudTrail log. Anda juga dapat menggunakan AWS CloudTrail log untuk meninjau panggilan API untuk Amazon S3 sebagai peristiwa.
- Untuk menerima pemberitahuan tentang kapan peristiwa tertentu terjadi di bucket S3 Anda: Siapkan notifikasi acara Amazon S3.
- Untuk mendapatkan catatan terperinci untuk permintaan yang dibuat ke bucket S3: Siapkan log akses S3.

Untuk daftar semua AWS layanan berbeda yang dapat Anda gunakan untuk memantau data Anda, lihat Pencatatan dan pemantauan di Amazon S3.

Pola bucket tujuan umum umum untuk membangun aplikasi di Amazon S3

Saat membuat aplikasi di Amazon S3, Anda dapat menggunakan bucket tujuan umum yang unik untuk memisahkan kumpulan data atau beban kerja yang berbeda. Saat Anda membuat aplikasi yang melayani pengguna akhir atau grup pengguna yang berbeda, gunakan pola desain praktik terbaik kami untuk membangun aplikasi yang dapat memanfaatkan fitur dan skalabilitas Amazon S3 dengan sebaik-baiknya.

> **Important**
>
> Kami menyarankan Anda membuat nama bucket tujuan umum yang tidak dapat diprediksi. Jangan menulis kode dengan asumsi nama bucket pilihan Anda tersedia kecuali Anda telah membuat bucket. Salah satu metode untuk membuat nama bucket yang tidak dapat diprediksi adalah dengan menambahkan Globally Unique Identifier (GUID) ke nama bucket Anda, misalnya,. `amzn-s3-demo-bucket-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111` Untuk informasi selengkapnya tentang aturan penamaan bucket tujuan umum, lihatAturan penamaan bucket tujuan umum.

Topik

- Pola bucket serba guna multi-penyewa
- Bucket-per-use pola

Pola bucket serba guna multi-penyewa

Dengan bucket multi-tenant, Anda membuat satu bucket tujuan umum untuk tim atau beban kerja. Anda menggunakan awalan S3 unik untuk mengatur objek yang Anda simpan di ember. Prefiks adalah string karakter di bagian awal nama kunci objek. Prefiks dapat memiliki panjang berapa pun, tergantung pada panjang maksimum dari nama kunci objek (1.024 byte). Anda dapat menganggap prefiks sebagai cara untuk mengatur data Anda dengan cara yang mirip dengan direktori. Namun, prefiks bukan direktori.

Misalnya, untuk menyimpan informasi tentang kota, Anda dapat mengaturnya berdasarkan benua, lalu berdasarkan negara, lalu berdasarkan provinsi atau negara bagian. Karena nama-nama ini biasanya tidak mengandung tanda baca, Anda dapat menggunakan garis miring (/) sebagai

pembatas. Contoh berikut menunjukkan awalan yang digunakan untuk mengatur nama kota berdasarkan benua, negara, dan kemudian provinsi atau negara bagian, menggunakan pembatas garis miring (/).

- Europe/France/NouvelleA-Aquitaine/Bordeaux
- Utara America/Canada/Quebec/Montreal
- Utara America/USA/Washington/Bellevue
- Utara America/USA/Washington/Seattle

Pola ini berskala baik ketika Anda memiliki ratusan kumpulan data unik dalam ember tujuan umum. Dengan awalan, Anda dapat dengan mudah mengatur dan mengelompokkan kumpulan data ini.

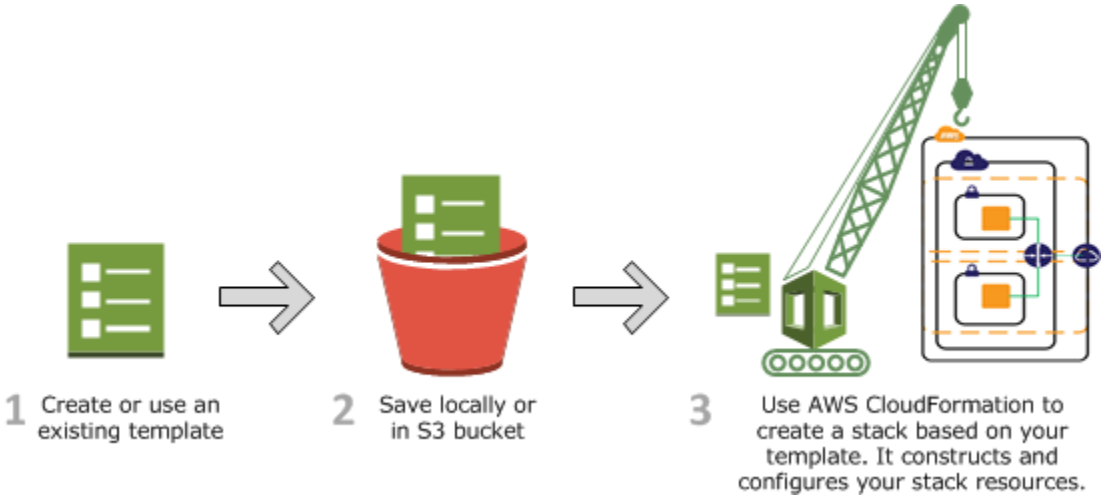
Namun, satu kelemahan potensial pada pola bucket tujuan umum multi-tenant adalah bahwa banyak fitur tingkat ember S3 seperti enkripsi bucket default, S3 Versioning, dan S3 Requester Pays ditetapkan pada tingkat ember dan bukan level awalan. Jika kumpulan data yang berbeda dalam bucket multi-tenant memiliki persyaratan unik, fakta bahwa Anda tidak dapat mengonfigurasi banyak fitur tingkat ember S3 di tingkat awalan dapat menyulitkan Anda untuk menentukan pengaturan yang benar untuk setiap kumpulan data. Selain itu, dalam bucket multi-tenant, alokasi biaya dapat menjadi kompleks saat Anda bekerja untuk memahami penyimpanan, permintaan, dan transfer data yang terkait dengan awalan tertentu.

Bucket-per-use pola

Dengan bucket-per-use pola tersebut, Anda membuat bucket tujuan umum untuk setiap kumpulan data, pengguna akhir, atau tim yang berbeda. Karena Anda dapat mengonfigurasi fitur tingkat ember S3 untuk masing-masing bucket ini, Anda dapat menggunakan pola ini untuk mengonfigurasi pengaturan tingkat ember yang unik. Misalnya, Anda dapat mengonfigurasi fitur seperti enkripsi bucket default, S3 Versioning, dan S3 Requester Pays dengan cara yang disesuaikan dengan kumpulan data di setiap bucket. Menggunakan satu bucket untuk setiap kumpulan data, pengguna akhir, atau tim yang berbeda juga dapat membantu Anda menyederhanakan manajemen akses dan strategi alokasi biaya.

Kelemahan potensial dari strategi ini adalah Anda perlu mengelola ribuan ember yang berpotensi. Semua Akun AWS memiliki kuota default 10.000 bucket tujuan umum. Anda dapat menambah kuota bucket untuk akun dengan mengirimkan permintaan kenaikan kuota. Untuk meminta peningkatan bucket tujuan umum, kunjungi konsol Service Quotas.

Untuk mengelola bucket-per-use pola Anda dan menyederhanakan manajemen infrastruktur Anda, Anda dapat menggunakannya AWS CloudFormation. Anda dapat membuat AWS CloudFormation template khusus untuk pola Anda yang sudah menentukan semua pengaturan yang Anda inginkan untuk bucket tujuan umum S3 Anda sehingga Anda dapat dengan mudah menerapkan dan melacak setiap perubahan pada infrastruktur Anda. Untuk informasi selengkapnya, lihat AWS::S3::Bucket di AWS CloudFormation Panduan Pengguna.



Aturan penamaan bucket tujuan umum

Saat membuat bucket tujuan umum, pastikan Anda mempertimbangkan panjang, karakter yang valid, pemformatan, dan keunikan nama bucket. Bagian berikut memberikan informasi tentang penamaan bucket tujuan umum, termasuk aturan penamaan, praktik terbaik, dan contoh untuk membuat bucket tujuan umum dengan nama yang menyertakan pengenal unik global (GUID).

Untuk informasi tentang nama kunci objek, lihat Membuat nama kunci objek.

Untuk membuat ember tujuan umum, lihatthe section called “Membuat ember tujuan umum”.

Topik

- Aturan penamaan bucket tujuan umum
- Contoh nama bucket tujuan umum
- Praktik terbaik
- Membuat bucket yang menggunakan GUID dalam nama bucket

Aturan penamaan bucket tujuan umum

Aturan penamaan berikut berlaku untuk bucket tujuan umum.

- Panjang nama bucket harus antara 3 (menit) dan 63 (maks) karakter.
- Nama bucket hanya dapat terdiri dari huruf kecil, angka, titik (`.`), dan tanda hubung (). `-`
- Nama bucket harus diawali dan juga diakhiri dengan huruf atau, nomor.
- Nama bucket tidak boleh berisi dua periode yang berdekatan.
- Nama bucket tidak boleh diformat sebagai alamat IP (misalnya,`192.168.5.4`).
- Nama bucket tidak boleh dimulai dengan prefiks `xn--`.
- Nama bucket tidak boleh dimulai dengan prefiks `sthree-`.
- Nama bucket tidak boleh dimulai dengan prefiks `amzn-s3-demo-`.
- Nama bucket tidak boleh diakhiri dengan sufiks `-s3alias`. Sufiks ini dicadangkan untuk nama alias titik akses. Untuk informasi selengkapnya, lihat Alias titik akses.
- Nama bucket tidak boleh diakhiri dengan sufiks `--ol-s3`. Sufiks ini dicadangkan untuk nama alias Titik Akses Lambda Objek. Untuk informasi selengkapnya, lihat Cara menggunakan alias gaya bucket untuk bucket S3 Anda Titik Akses Lambda Objek.
- Nama bucket tidak boleh diakhiri dengan sufiks `.mrap`. Akhiran ini dicadangkan untuk nama Titik Akses Multi-Wilayah. Untuk informasi selengkapnya, lihat Aturan penamaan Titik Akses Multi-Wilayah Amazon S3.
- Nama bucket tidak boleh diakhiri dengan sufiks `--x-s3`. Akhiran ini dicadangkan untuk ember direktori. Untuk informasi selengkapnya, lihat Aturan penamaan bucket direktori.
- Nama bucket tidak boleh diakhiri dengan sufiks `--table-s3`. Sufiks ini dicadangkan untuk ember Tabel S3. Untuk informasi selengkapnya, lihat Aturan penamaan bucket, tabel, dan namespace Amazon S3.
- Bucket yang digunakan dengan Amazon S3 Transfer Acceleration tidak dapat memiliki period `.` () dalam namanya. Untuk informasi lebih lanjut tentang Transfer Acceleration, lihat Mengonfigurasi transfer file yang cepat dan aman menggunakan Amazon S3 Transfer Acceleration.

> ⚠ Important
>
> - Nama bucket harus unik Akun AWS di semua bagian Wilayah AWS dalam partisi. Partisi adalah pengelompokan Wilayah. AWS Saat ini memiliki tiga partisi: `aws` (Kawasan komersial), `aws-cn` (Wilayah Tiongkok), dan `aws-us-gov` (AWS GovCloud (US) Wilayah).

- Nama bucket tidak dapat digunakan oleh orang lain Akun AWS di partisi yang sama sampai bucket dihapus. Setelah Anda menghapus bucket, ketahuilah bahwa bucket lain Akun AWS di partisi yang sama dapat menggunakan nama bucket yang sama untuk bucket baru dan karenanya berpotensi menerima permintaan yang ditujukan untuk bucket yang dihapus. Jika Anda ingin mencegah hal ini, atau jika Anda ingin terus menggunakan nama bucket yang sama, jangan hapus bucket. Kami menyarankan Anda mengosongkan ember dan menyimpannya, dan sebagai gantinya, memblokir permintaan bucket apa pun sesuai kebutuhan. Untuk ember yang tidak lagi digunakan secara aktif, kami sarankan untuk mengosongkan ember dari semua objek untuk meminimalkan biaya sambil mempertahankan ember itu sendiri.
- Saat Anda membuat ember tujuan umum, Anda memilih namanya dan Wilayah AWS untuk membuatnya. Setelah membuat bucket tujuan umum, Anda tidak dapat mengubah nama atau Region.
- Jangan sertakan informasi sensitif dalam nama bucket. Nama bucket terlihat dalam URLs yang menunjuk objek dalam bucket.

Note

Sebelum 1 Maret 2018, bucket yang dibuat di Wilayah AS Timur (Virginia Utara) dapat memiliki nama hingga mencapai 255 karakter, dan menyertakan huruf besar dan garis bawah. Mulai 1 Maret 2018, bucket baru di AS Timur (Virginia Utara) harus mematuhi aturan yang sama yang berlaku di semua Wilayah lainnya.

Contoh nama bucket tujuan umum

Nama bucket berikut menunjukkan contoh karakter mana yang diizinkan dalam nama bucket tujuan umum: a-z, 0-9, dan tanda hubung (). - Awalan `amzn-s3-demo-` cadangan digunakan di sini hanya untuk ilustrasi. Karena ini adalah awalan cadangan, Anda tidak dapat membuat nama bucket yang dimulai dengan`amzn-s3-demo-`.

- `amzn-s3-demo-bucket1-a1b2c3d4-5678-90ab-cdef-example11111`
- `amzn-s3-demo-bucket`

Contoh nama bucket berikut valid tetapi tidak direkomendasikan untuk penggunaan selain hosting situs web statis karena mengandung periode (`.`):

- `example.com`
- `www.example.com`
- `my.example.s3.bucket`

Contoh nama bucket berikut bersifat tidak valid:

- `amzn_s3_demo_bucket` (mengandung garis bawah)
- `AmznS3DemoBucket` (mengandung huruf besar)
- `amzn-s3-demo-bucket-`(dimulai dengan `amzn-s3-demo-` awalan dan diakhiri dengan tanda hubung)
- `example..com`(berisi dua periode berturut-turut)
- `192.168.5.4`(cocok dengan format alamat IP)

Praktik terbaik

Saat memberi nama bucket tujuan umum Anda, pertimbangkan praktik terbaik penamaan bucket berikut.

Pilih skema penamaan ember yang tidak mungkin menyebabkan konflik penamaan

Jika aplikasi Anda secara otomatis membuat bucket, pilih skema penamaan bucket yang tidak mungkin menyebabkan konflik penamaan. Pastikan bahwa logika aplikasi Anda akan memilih nama bucket yang berbeda jika nama bucket-nya sudah diambil.

Tambahkan pengidentifikasi unik global (GUIDs) ke nama bucket

Kami menyarankan Anda membuat nama bucket yang tidak dapat diprediksi. Jangan menulis kode dengan asumsi nama bucket pilihan Anda tersedia kecuali Anda telah membuat bucket. Salah satu metode untuk membuat nama bucket yang tidak dapat diprediksi adalah dengan menambahkan Globally Unique Identifier (GUID) ke nama bucket Anda, misalnya,. `amzn-s3-demo-bucket-a1b2c3d4-5678-90ab-cdef-example11111` Untuk informasi selengkapnya, lihat the section called “Membuat bucket yang menggunakan GUID dalam nama bucket”.

Hindari menggunakan period (`.`) dalam nama bucket

Untuk kompatibilitas terbaik, sebaiknya hindari penggunaan period (`.`) dalam nama bucket, kecuali bucket yang hanya digunakan untuk hosting situs statis. Jika menyertakan periode dalam nama bucket, Anda tidak dapat menggunakan virtual-host-style pengalamatan melalui HTTPS, kecuali Anda melakukan validasi sertifikat sendiri. Sertifikat keamanan yang digunakan untuk hosting virtual bucket tidak berfungsi untuk ember dengan titik dalam nama mereka.

Keterbatasan ini tidak memengaruhi bucket yang digunakan untuk hosting situs web statis, karena hosting situs web statis hanya tersedia melalui HTTP. Untuk informasi lebih lanjut tentang virtual-host-style pengalamatan, lihatHosting virtual dari ember tujuan umum. Untuk informasi tentang hosting situs web, lihat Hosting situs web statis menggunakan Amazon S3.

Pilih nama yang relevan

Saat Anda memberi nama ember, kami sarankan Anda memilih nama yang relevan dengan Anda atau bisnis Anda. Hindari menggunakan nama yang terkait dengan orang lain. Misalnya, hindari menggunakan `AWS` atau `Amazon` dalam nama bucket Anda.

Jangan hapus bucket sehingga Anda dapat menggunakan kembali nama bucket

Jika sebuah bucket kosong, Anda dapat menghapusnya. Setelah Anda menghapus bucket, namanya akan tersedia untuk digunakan kembali. Namun, Anda tidak dijamin dapat menggunakan kembali nama itu segera, atau sama sekali. Setelah Anda menghapus bucket, beberapa waktu mungkin berlalu sebelum Anda dapat menggunakan kembali nama tersebut. Selain itu, yang lain Akun AWS mungkin membuat ember dengan nama yang sama sebelum Anda dapat menggunakan kembali nama tersebut.

Setelah Anda menghapus bucket tujuan umum, ketahuilah bahwa bucket lain Akun AWS di partisi yang sama dapat menggunakan nama bucket yang sama untuk bucket baru dan karenanya berpotensi menerima permintaan yang ditujukan untuk bucket tujuan umum yang dihapus. Jika Anda ingin mencegah hal ini, atau jika Anda ingin terus menggunakan nama bucket tujuan umum yang sama, jangan hapus bucket tujuan umum. Kami menyarankan Anda mengosongkan ember dan menyimpannya, dan sebagai gantinya, memblokir permintaan bucket apa pun sesuai kebutuhan.

Membuat bucket yang menggunakan GUID dalam nama bucket

Contoh berikut menunjukkan cara membuat bucket tujuan umum yang menggunakan GUID di akhir nama bucket.

Menggunakan AWS CLI

AWS CLI Contoh berikut membuat bucket tujuan umum di Wilayah Wilayah () AS Barat (California) dengan contoh nama bucket yang menggunakan pengenal unik global (GUID). `us-west-1` Untuk menggunakan perintah contoh ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3api create-bucket \
    --bucket amzn-s3-demo-bucket1$(uuidgen | tr -d - | tr '[:upper:]' '[:lower:]' ) \
    --region us-west-1 \
    --create-bucket-configuration LocationConstraint=us-west-1
```

Menggunakan AWS SDK for Java

Contoh berikut menunjukkan kepada Anda cara membuat dengan GUID di akhir nama bucket di Wilayah AS Timur (Virginia Utara) (`us-east-1`) dengan menggunakan. AWS SDK untuk Java Untuk menggunakan contoh ini, ganti *`user input placeholders`*dengan informasi Anda sendiri. Untuk informasi tentang lainnya AWS SDKs, lihat Alat untuk Dibangun AWS.

```
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.Bucket;
import com.amazonaws.services.s3.model.CreateBucketRequest;

import java.util.List;
import java.util.UUID;

public class CreateBucketWithUUID {
    public static void main(String[] args) {
        final AmazonS3 s3 =
 AmazonS3ClientBuilder.standard().withRegion(Regions.US_EAST_1).build();
        String bucketName = "amzn-s3-demo-bucket" +
 UUID.randomUUID().toString().replace("-", "");
        CreateBucketRequest createRequest = new CreateBucketRequest(bucketName);
        System.out.println(bucketName);
        s3.createBucket(createRequest);
    }
}
```

Kuota, batasan, dan batasan bucket tujuan umum

Bucket tujuan umum Amazon S3 dimiliki oleh Akun AWS yang membuatnya. Kepemilikan bucket tidak dapat dialihkan ke akun lain.

Kuota ember

Secara default, Anda dapat membuat hingga 10.000 ember tujuan umum per Akun AWS. Untuk meminta peningkatan kuota untuk bucket tujuan umum, kunjungi konsol Service Quotas.

Important

Kami sangat menyarankan hanya menggunakan `ListBuckets` permintaan paginasi. `ListBuckets`Permintaan tanpa paginasi hanya didukung untuk Akun AWS disetel ke kuota bucket tujuan umum default sebesar 10.000. Jika Anda memiliki kuota bucket tujuan umum yang disetujui di atas 10.000, Anda harus mengirim `ListBuckets` permintaan paginasi untuk mencantumkan bucket akun Anda. Semua `ListBuckets` permintaan tanpa paginasi akan ditolak Akun AWS dengan kuota bucket tujuan umum lebih dari 10.000.

Note

Anda harus menggunakan yang berikut ini Wilayah AWS untuk melihat kuota Anda, penggunaan bucket, atau meminta peningkatan untuk ember tujuan umum Anda di Anda. Akun AWS

- Kuota bucket tujuan umum untuk Wilayah komersial hanya dapat dilihat dan dikelola dari US East (Virginia N.).
- Kuota bucket tujuan umum hanya AWS GovCloud (US) dapat dilihat dan dikelola dari AWS GovCloud (AS-Barat).

Untuk informasi tentang kuota layanan, lihat kuota AWS layanan di. Referensi Umum Amazon Web Services

Objek dan batasan bucket

Tidak ada ukuran bucket maksimal atau batas jumlah objek yang dapat Anda simpan dalam satu bucket. Anda dapat menyimpan semua objek Anda dalam satu bucket, atau Anda dapat mengaturnya di dalam beberapa bucket. Namun, Anda tidak dapat membuat bucket dari bucket lain.

Peraturan penamaan bucket

Saat Anda membuat ember, Anda memilih namanya dan Wilayah AWS untuk membuatnya. Setelah membuat bucket, Anda tidak dapat mengubah nama atau Wilayahnya. Untuk informasi selengkapnya tentang bucket, lihat Aturan penamaan bucket tujuan umum.

Mengakses bucket tujuan umum Amazon S3

Anda dapat mengakses bucket tujuan umum Amazon S3 dengan menggunakan konsol Amazon S3, AWS Command Line Interface, atau Amazon S3 AWS SDKs REST API. Setiap metode mengakses bucket tujuan umum S3 mendukung kasus penggunaan tertentu. Untuk informasi selengkapnya, silakan lihat bagian-bagian berikut ini.

Topik

- Kasus penggunaan
- Konsol Amazon S3
- AWS CLI
- AWS SDKs
- API REST Amazon S3

Kasus penggunaan

Bergantung pada kasus penggunaan untuk bucket tujuan umum Amazon S3 Anda, ada berbagai metode yang disarankan untuk mengakses data yang mendasarinya di bucket Anda. Daftar berikut mencakup kasus penggunaan umum untuk mengakses data Anda.

- Situs web statis–Anda dapat menggunakan Amazon S3 untuk host situs web statis. Dalam kasus penggunaan ini, Anda dapat mengonfigurasi bucket tujuan umum S3 agar berfungsi seperti situs web. Untuk contoh yang memandu Anda melalui langkah-langkah menghosting situs web di Amazon S3, lihat Tutorial: Mengonfigurasi situs web statis untuk Amazon S3.

Untuk meng-host situs web statis dengan pengaturan keamanan seperti Blokir Akses Publik diaktifkan, sebaiknya gunakan Amazon CloudFront dengan Origin Access Control (OAC) dan menerapkan header keamanan tambahan, seperti HTTPS. Untuk informasi selengkapnya, lihat Memulai dengan situs web statis yang aman.

> **Note**
>
> Amazon S3 mendukung gaya host virtual dan gaya jalur untuk akses situs web statis. URLs Karena bucket dapat diakses menggunakan gaya path-style dan virtual-hosted—style, sebaiknya Anda membuat bucket dengan nama bucket yang sesuai dengan URLs DNS. Untuk informasi selengkapnya, lihat Kuota, batasan, dan batasan bucket tujuan umum.

- Kumpulan data bersama — Saat Anda menskalakan di Amazon S3, adalah umum untuk mengadopsi model multi-penyewa, di mana Anda menetapkan pelanggan akhir atau unit bisnis yang berbeda ke awalan unik dalam bucket tujuan umum bersama. Dengan menggunakan Titik akses Amazon S3, Anda dapat membagi satu kebijakan bucket besar menjadi kebijakan titik akses terpisah dan terpisah untuk setiap aplikasi yang perlu mengakses set data bersama. Pendekatan ini membuatnya lebih mudah untuk fokus pada membangun kebijakan akses yang tepat untuk aplikasi tanpa mengganggu apa yang dilakukan aplikasi lain dalam set data bersama. Untuk informasi selengkapnya, lihat Mengelola akses ke kumpulan data bersama dengan titik akses.
- Beban kerja throughput tinggi - Mountpoint untuk Amazon S3 adalah klien file sumber terbuka throughput tinggi untuk memasang bucket tujuan umum Amazon S3 sebagai sistem file lokal. Dengan Mountpoint, aplikasi Anda dapat mengakses objek yang disimpan di Amazon S3 melalui operasi sistem file, seperti buka dan baca. Mountpoint secara otomatis menerjemahkan operasi ini ke dalam panggilan API objek S3, memberikan aplikasi Anda akses ke penyimpanan elastis serta throughput Amazon S3 melalui antarmuka file. Untuk informasi selengkapnya, lihat Pasang bucket Amazon S3 sebagai sistem file lokal.
- Aplikasi Multi-Wilayah — Titik Akses Multi-Wilayah Amazon S3 menyediakan titik akhir global yang dapat digunakan aplikasi untuk memenuhi permintaan dari bucket tujuan umum S3 yang terletak di beberapa tempat. Wilayah AWS Anda dapat menggunakan Titik Akses Multi-Wilayah untuk membangun aplikasi multi-Wilayah dengan arsitektur yang sama dengan yang digunakan di satu Wilayah, dan kemudian menjalankan aplikasi tersebut di mana saja di seluruh dunia. Alih-alih mengirim permintaan melalui internet publik, Titik Akses Multi-Wilayah menyediakan ketahanan jaringan bawaan dengan percepatan permintaan berbasis internet ke Amazon S3. Untuk informasi selengkapnya, lihat Mengelola lalu lintas Multi-wilayah dengan Titik Akses Multi-Wilayah.

- Secure Shell (SSH) File Transfer Protocol (SFTP) — Jika Anda mencoba mentransfer data sensitif melalui internet dengan aman, Anda dapat menggunakan server berkemampuan SFTP dengan bucket tujuan umum Amazon S3 Anda. AWS SFTP adalah protokol jaringan yang mendukung fungsionalitas keamanan dan otentikasi penuh SSH. Dengan protokol ini, Anda memiliki kontrol halus atas identitas pengguna, izin, dan kunci. Atau, Anda dapat menggunakan kebijakan IAM untuk mengelola akses. Untuk mengaitkan server berkemampuan SFTP dengan bucket Amazon S3 Anda, pastikan untuk membuat server berkemampuan SFTP terlebih dahulu. Kemudian, Anda mengatur akun pengguna, dan mengaitkan server dengan bucket tujuan umum Amazon S3. Untuk panduan proses ini, lihat AWS Transfer for SFTP - Layanan SFTP yang Dikelola Sepenuhnya untuk Amazon S3 di Blog.AWS

Konsol Amazon S3

Konsol adalah antarmuka pengguna berbasis web untuk mengelola Amazon S3 AWS dan sumber daya. Dengan konsol Amazon S3, Anda dapat dengan mudah mengakses bucket dan memodifikasi properti bucket. Anda juga dapat melakukan sebagian besar operasi bucket menggunakan UI konsol, tanpa harus menulis kode apa pun.

Jika Anda telah mendaftar Akun AWS, Anda dapat mengakses konsol Amazon S3 dengan masuk ke konsol Amazon S3 dan memilih S3 dari halaman beranda konsol Amazon S3. Anda juga dapat menggunakan tautan ini untuk langsung mengakses https://console.aws.amazon.com/s3/.

AWS CLI

Anda dapat menggunakan AWS CLI untuk mengeluarkan perintah atau membangun skrip di baris perintah sistem Anda untuk melakukan tugas AWS (termasuk S3). Misalnya, jika Anda perlu mengakses beberapa bucket, Anda dapat menghemat waktu dengan menggunakan AWS CLI untuk mengotomatiskan tugas umum dan berulang. Kemampuan naskah dan pengulangan untuk tindakan umum sering menjadi pertimbangan seiring bertumbuhnya organisasi.

AWS CLIIni menyediakan perintah untuk serangkaian luas Layanan AWS. AWS CLI Ini didukung di Windows, macOS, dan Linux. Untuk memulai, lihat Panduan Pengguna AWS Command Line Interface . Untuk informasi selengkapnya tentang perintah untuk Amazon S3, lihat s3api dan s3control di AWS CLI Referensi Perintah.

AWS SDKs

AWS menyediakan SDKs (perangkat pengembangan perangkat lunak) yang terdiri dari pustaka dan kode sampel untuk berbagai bahasa dan platform pemrograman (Java, Python, Ruby, .NET, iOS,

Android, dan sebagainya). AWS SDKs Menyediakan cara mudah untuk membuat akses terprogram ke S3 dan. AWS Amazon S3 adalah layanan REST. Anda dapat mengirim permintaan ke Amazon S3 menggunakan pustaka AWS SDK, yang membungkus API REST Amazon S3 yang mendasarinya dan menyederhanakan tugas pemrograman Anda. Misalnya, SDKs mengurus tugas-tugas seperti menghitung tanda tangan, menandatangani permintaan secara kriptografis, mengelola kesalahan, dan mencoba ulang permintaan secara otomatis. Untuk informasi tentang AWS SDKs, termasuk cara mengunduh dan menginstalnya, lihat Alat untuk AWS.

Setiap interaksi dengan Amazon S3 diautentikasi atau dilakukan secara anonim. Jika Anda menggunakan AWS SDKs, pustaka menghitung tanda tangan untuk otentikasi dari kunci yang Anda berikan. Untuk informasi selengkapnya tentang cara membuat permintaan ke Amazon S3, lihat Membuat permintaan.

API REST Amazon S3

Arsitektur Amazon S3 dirancang untuk menjadi bahasa pemrograman-netral, menggunakan antarmuka yang didukung AWS untuk menyimpan dan mengambil objek. Anda dapat mengakses S3 dan AWS secara terprogram menggunakan API REST Amazon S3. API REST adalah antarmuka HTTP ke Amazon S3. Dengan API REST, Anda menggunakan permintaan HTTP standar untuk membuat, dan menghapus bucket dan objek.

Untuk menggunakan API REST, Anda dapat menggunakan toolkit yang mendukung HTTP. Anda bahkan dapat menggunakan peramban untuk mengambil objek, selama objek tersebut dapat dibaca secara anonim.

API REST menggunakan header HTTP standar dan kode status, sehingga peramban dan toolkit standar bekerja sesuai harapan. Di beberapa area, kami telah menambahkan fungsi ke HTTP (misalnya, kami menambahkan header untuk mendukung kontrol akses). Dalam kasus ini, kami telah melakukan yang terbaik untuk menambahkan fungsionalitas baru dengan cara yang cocok dengan cara penggunaan HTTP standar.

Jika Anda melakukan panggilan API REST langsung di aplikasi Anda, Anda harus menulis kode untuk menghitung tanda tangan dan menambahkannya ke permintaan. Untuk informasi selengkapnya tentang cara membuat permintaan ke Amazon S3, lihat Membuat permintaan di Referensi API Amazon S3.

Hosting virtual dari ember tujuan umum

Hosting virtual adalah praktik melayani beberapa situs web dari server web tunggal. Salah satu cara untuk membedakan situs dalam permintaan API REST Amazon S3 Anda adalah dengan

menggunakan nama host yang jelas dari URI permintaan, bukan hanya bagian nama jalur dari URI. Permintaan REST Amazon S3 biasa menentukan bucket dengan menggunakan komponen yang dibatasi garis miring pertama dari jalur Request-URI. Sebagai gantinya, Anda dapat menggunakan hosting virtual Amazon S3 untuk menangani bucket tujuan umum dalam panggilan REST API dengan menggunakan header HTTP`Host`. Dalam praktiknya, Amazon S3 menafsirkan `Host` sebagai arti bahwa sebagian besar bucket secara otomatis dapat diakses untuk jenis permintaan terbatas di `https://`*`bucket-name`*`.s3.`*`region-code`*`.amazonaws.com`. Untuk daftar lengkap titik akhir dan Wilayah Amazon S3, lihat Titik akhir dan kuota Amazon S3 di Referensi Umum Amazon Web Services.

Hosting virtual juga memiliki manfaat lain. Dengan menamai bucket Anda setelah nama domain terdaftar Anda dan dengan membuat nama DNS alias untuk Amazon S3, Anda dapat sepenuhnya menyesuaikan URL sumber daya Amazon S3, sebagai contoh, `http://my.`*`bucket-name`*`.com/`. Anda juga dapat menerbitkan ke “direktori root” dari server virtual bucket Anda. Kemampuan ini bisa menjadi penting, karena banyak aplikasi yang ada mencari file di lokasi standar ini. Misalnya, `favicon.ico`, `robots.txt`, dan `crossdomain.xml` semuanya diharapkan dapat ditemukan di root.

> **Important**
>
> Saat Anda menggunakan bucket tujuan umum bergaya virtual dengan SSL, sertifikat wildcard SSL hanya cocok dengan bucket yang tidak berisi dots (). `.` Untuk mengatasi batasan ini, gunakan HTTP atau tulis logika verifikasi sertifikat Anda sendiri. Untuk informasi selengkapnya, lihat Rencana Penghentian Jalur Amazon S3 di Blog Berita AWS .

Topik

- Permintaan Cara Jalur
- Permintaan cara hosting virtual
- Spesifikasi bucket header Host HTTP
- Contoh
- Menyesuaikan Amazon URLs S3 dengan catatan CNAME
- Cara mengaitkan nama host dengan bucket Amazon S3
- Keterbatasan
- Kompatibilitas mundur

Permintaan Cara Jalur

Saat ini, Amazon S3 mendukung akses URL dengan gaya hosting virtual dan gaya jalur di seluruh Wilayah AWS. Namun, gaya jalur URLs akan dihentikan di masa depan. Untuk informasi selengkapnya, lihat catatan Penting berikut ini.

Di Amazon S3, gaya jalur URLs menggunakan format berikut:

```
https://s3.region-code.amazonaws.com/bucket-name/key-name
```

Misalnya, jika Anda membuat bucket dengan nama `amzn-s3-demo-bucket1` di Wilayah AS Barat (Oregon), dan Anda ingin mengakses objek `puppy.jpg` dalam bucket tersebut, Anda dapat menggunakan URL cara jalur berikut:

```
https://s3.us-west-2.amazonaws.com/amzn-s3-demo-bucket1/puppy.jpg
```

> **Important**
>
> Pembaruan (23 September 2020) - Untuk memastikan bahwa pelanggan memiliki waktu yang mereka butuhkan untuk beralih ke gaya host virtual URLs, kami telah memutuskan untuk menunda penghentian gaya jalur. URLs Untuk informasi lebih lanjut, lihat Rencana Penghentian Jalur Amazon S3-Cerita Seluruhnya di Blog Berita AWS .

> **Warning**
>
> Saat menghosting konten situs web yang akan diakses dari browser web, hindari menggunakan gaya jalur URLs, yang mungkin mengganggu model keamanan asal browser yang sama. Untuk meng-host konten situs web, kami menyarankan Anda menggunakan titik akhir situs web S3 atau distribusi. CloudFront Untuk informasi selengkapnya, lihat Titik akhir situs web dan Menerapkan aplikasi satu halaman berbasis React ke Amazon S3 dan di Pola Panduan Perspektif. CloudFront AWS

Permintaan cara hosting virtual

Dalam URI dengan cara hosting virtual, nama bucket adalah bagian dari nama domain di URL.

Gaya host virtual Amazon S3 menggunakan format berikut: URLs

```
https://bucket-name.s3.region-code.amazonaws.com/key-name
```

Dalam contoh ini, `amzn-s3-demo-bucket1` adalah nama bucket, AS Barat (Oregon) adalah Wilayah, dan `puppy.png` adalah nama kuncinya:

```
https://amzn-s3-demo-bucket1.s3.us-west-2.amazonaws.com/puppy.png
```

Spesifikasi bucket header **`Host`** HTTP

Selama permintaan `GET` Anda tidak menggunakan titik akhir SSL, Anda dapat menentukan bucket untuk permintaan menggunakan header `Host` HTTP. Header `Host` dalam permintaan REST diinterpretasikan sebagai berikut:

- Jika header `Host` dihilangkan atau nilainya `s3.region-code.amazonaws.com`, bucket untuk permintaan akan menjadi komponen yang dibatasi garis miring pertama dari Request-URI, dan kunci untuk permintaan tersebut adalah sisa dari Request-URI. Ini adalah metode biasa, seperti yang digambarkan oleh contoh pertama dan kedua di bagian ini. Menghapus header `Host` hanya valid untuk permintaan HTTP 1.0.
- Atau, jika nilai header `Host` berakhir di `.s3.region-code.amazonaws.com`, nama bucket adalah komponen utama nilai header `Host` hingga `.s3.region-code.amazonaws.com`. Kunci untuk permintaan adalah Request-URI. Interpretasi ini memaparkan bucket sebagai subdomain `.s3.region-code.amazonaws.com`, sebagaimana digambarkan oleh contoh ketiga dan keempat dalam bagian ini.
- Jika tidak, bucket permintaan adalah nilai huruf kecil dari header `Host`, dan kunci untuk permintaan adalah Request-URI. Interpretasi ini berguna ketika Anda telah mendaftarkan nama DNS yang sama dengan nama bucket Anda dan telah mengonfigurasikan nama tersebut menjadi nama kanonik (CNAME) alias untuk Amazon S3. Prosedur untuk mendaftarkan nama domain dan mengonfigurasi catatan DNS CNAME berada di luar cakupan panduan ini, tetapi hasilnya diilustrasikan oleh contoh akhir di bagian ini.

Contoh

Bagian ini memberikan contoh URLs dan permintaan.

Example — Gaya jalan dan permintaan URLs

Contoh ini menggunakan hal berikut:

- Nama Bucket - `example.com`
- Wilayah AS Timur (Virginia Utara)
- Nama kunci - `homepage.html`

URL-nya adalah sebagai berikut:

```
http://s3.us-east-1.amazonaws.com/example.com/homepage.html
```

Permintaan tersebut adalah sebagai berikut:

```
GET /example.com/homepage.html HTTP/1.1
Host: s3.us-east-1.amazonaws.com
```

Permintaan dengan HTTP 1.0 dan menghilangkan header `Host` adalah sebagai berikut:

```
GET /example.com/homepage.html HTTP/1.0
```

Untuk informasi tentang nama yang kompatibel dengan DNS, lihat Batasan. Untuk informasi lebih lanjut tentang kunci, lihat Kunci.

Example — Gaya dan permintaan yang di-host virtual URLs

Contoh ini menggunakan hal berikut:

- Nama Bucket - `amzn-s3-demo-bucket1`
- Wilayah - Eropa (Irlandia)
- Nama kunci - `homepage.html`

URL-nya adalah sebagai berikut:

```
http://amzn-s3-demo-bucket1.s3.eu-west-1.amazonaws.com/homepage.html
```

Permintaan tersebut adalah sebagai berikut:

```
GET /homepage.html HTTP/1.1
Host: amzn-s3-demo-bucket1.s3.eu-west-1.amazonaws.com
```

Example – Metode alias CNAME

Untuk menggunakan metode ini, Anda harus mengonfigurasi nama DNS Anda sebagai CNAME alias untuk *bucket-name*.s3.us-east-1.amazonaws.com. Untuk informasi selengkapnya, lihat Menyesuaikan Amazon URLs S3 dengan catatan CNAME.

Contoh ini menggunakan hal berikut:

- Nama Bucket - example.com
- Nama kunci - homepage.html

URL-nya adalah sebagai berikut:

```
http://www.example.com/homepage.html
```

Contohnya adalah sebagai berikut:

```
GET /homepage.html HTTP/1.1
Host: www.example.com
```

Menyesuaikan Amazon URLs S3 dengan catatan CNAME

Tergantung pada kebutuhan Anda, Anda mungkin tidak ingin s3.*region-code*.amazonaws.com muncul di situs web atau layanan Anda. Sebagai contoh, jika Anda meng-hosting gambar situs web di Amazon S3, Anda mungkin lebih memilih http://images.example.com/ daripada http://images.example.com.s3.us-east-1.amazonaws.com/. Setiap bucket dengan nama yang kompatibel dengan DNS dapat direferensikan sebagai berikut: http://*BucketName*.s3.*Region*.amazonaws.com/[*Filename*], sebagai contoh, http://images.example.com.s3.us-east-1.amazonaws.com/mydog.jpg. Dengan menggunakan CNAME, Anda dapat memetakan images.example.com ke nama host Amazon S3 agar URL sebelumnya dapat menjadi http://images.example.com/mydog.jpg.

Nama bucket Anda harus sama dengan CNAME. Misalnya, jika Anda membuat CNAME untuk memetakan images.example.com ke images.example.com.s3.us-east-1.amazonaws.com, maka http://images.example.com/filename dan http://images.example.com.s3.us-east-1.amazonaws.com/filename akan sama.

Catatan DNS CNAME harus membuat alias nama domain Anda dengan nama host dengan cara hosting virtual yang sesuai. Misalnya, jika nama bucket dan nama domain Anda adalah

`images.example.com` dan bucket Anda berada di Wilayah AS Timur (Virginia Utara), catatan CNAME harus alias ke `images.example.com.s3.us-east-1.amazonaws.com`.

```
images.example.com CNAME    images.example.com.s3.us-east-1.amazonaws.com.
```

Amazon S3 menggunakan nama host untuk menentukan nama bucket. Jadi CNAME dan nama bucket harus sama. Sebagai contoh, anggaplah Anda telah mengonfigurasi `www.example.com` sebagai CNAME untuk `www.example.com.s3.us-east-1.amazonaws.com`. Saat Anda mengakses `http://www.example.com`, Amazon S3 menerima permintaan yang serupa dengan yang berikut ini:

Example

```
GET / HTTP/1.1
Host: www.example.com
Date: date
Authorization: signatureValue
```

Amazon S3 hanya melihat nama host asli `www.example.com` dan tidak menyadari pemetaan CNAME yang digunakan untuk menyelesaikan permintaan.

Anda dapat menggunakan titik akhir Amazon S3 apa pun dalam alias CNAME. Misalnya, `s3.ap-southeast-1.amazonaws.com` dapat digunakan dalam alias CNAME. Untuk informasi selengkapnya tentang titik akhir, lihat Meminta titik akhir di Referensi API Amazon S3. Untuk membuat situs web statis menggunakan domain khusus, lihat Tutorial: Mengonfigurasi situs web statis menggunakan domain kustom yang terdaftar di Route 53.

Important

Saat menggunakan custom URLs with CNAMEs, Anda harus memastikan ada bucket yang cocok untuk catatan CNAME atau alias yang Anda konfigurasikan. Misalnya, jika Anda membuat entri DNS untuk `www.example.com` dan `login.example.com` memublikasikan konten web menggunakan S3, Anda harus membuat bucket `www.example.com` dan `login.example.com`.

Saat catatan CNAME atau alias dikonfigurasi dengan menunjuk ke titik akhir S3 tanpa bucket yang cocok, AWS pengguna mana pun dapat membuat bucket tersebut dan memublikasikan konten di bawah alias yang dikonfigurasi, meskipun kepemilikannya tidak sama.

Untuk alasan yang sama, sebaiknya Anda mengubah atau menghapus CNAME atau alias yang sesuai saat menghapus bucket.

Cara mengaitkan nama host dengan bucket Amazon S3

Cara mengaitkan nama host dengan bucket Amazon S3 dengan menggunakan alias CNAME

1. Pilih nama host milik domain yang Anda kontrol.

   Contoh ini menggunakan subdomain `images` dari domain `example.com`.
2. Buat bucket yang sesuai dengan nama host.

   Dalam contoh ini, nama host dan bucket adalah `images.example.com`. Nama bucket harus sama persis dengan nama host.
3. Buat catatan DNS CNAME yang mendefinisikan nama host sebagai alias untuk bucket Amazon S3.

   Sebagai contoh:

   ```
   images.example.com CNAME images.example.com.s3.us-west-2.amazonaws.com
   ```

   Important

   Untuk alasan perutean permintaan, catatan DNS CNAME harus ditentukan persis seperti yang ditunjukkan dalam contoh sebelumnya. Jika tidak, maka catatan akan terlihat beroperasi dengan benar, tetapi pada akhirnya akan menghasilkan perilaku yang tidak dapat diprediksi.

   Prosedur untuk mengonfigurasi catatan DNS CNAME tergantung pada server DNS atau penyedia DNS Anda. Untuk informasi khusus, lihat dokumentasi server Anda atau hubungi penyedia Anda.

Keterbatasan

SOAP APIs for tidak tersedia untuk pelanggan baru, dan mendekati End of Life (EOL) pada 31 Agustus 2025. Kami menyarankan Anda menggunakan REST API atau AWS SDKs.

Kompatibilitas mundur

Bagian berikut mencakup berbagai aspek kompatibilitas mundur Amazon S3 yang terkait dengan permintaan URL dengan cara jalur dan cara hosting virtual.

Titik akhir legasi

Beberapa Wilayah mendukung titik akhir legasi. Anda mungkin melihat titik akhir ini di log atau AWS CloudTrail log akses server Anda. Untuk informasi lebih lanjut, tinjau informasi berikut. Untuk daftar lengkap titik akhir dan Wilayah Amazon S3, lihat Titik akhir dan kuota Amazon S3 di Referensi Umum Amazon Web Services.

Important

Meskipun Anda mungkin melihat titik akhir legasi di log Anda, kami menyarankan Anda untuk selalu menggunakan sintaks titik akhir standar untuk mengakses bucket Anda.
Gaya host virtual Amazon S3 menggunakan format berikut: URLs

```
https://bucket-name.s3.region-code.amazonaws.com/key-name
```

Di Amazon S3, gaya jalur URLs menggunakan format berikut:

```
https://s3.region-code.amazonaws.com/bucket-name/key-name
```

s3-Wilayah

Beberapa Wilayah Amazon S3 yang lebih lama mendukung titik akhir yang berisi tanda hubung (`-`) antara `s3` dan kode Wilayah (misalnya, `s3-us-west-2`), daripda titik (misalnya, `s3.us-west-2`). Jika bucket Anda berada di salah satu Wilayah ini, Anda mungkin melihat format titik akhir berikut dalam log akses server Anda atau log CloudTrail:

```
https://bucket-name.s3-region-code.amazonaws.com
```

Dalam contoh ini, nama bucket adalah amzn-s3-demo-bucket1 dan Region adalah US West (Oregon):

```
https://amzn-s3-demo-bucket1.s3-us-west-2.amazonaws.com
```

Titik akhir global legasi

Untuk beberapa Wilayah, Anda dapat menggunakan titik akhir global legasi untuk membuat permintaan yang tidak menentukan titik akhir spesifik Wilayah. Titik akhir global legasi adalah sebagai berikut:

```
bucket-name.s3.amazonaws.com
```

Di log atau CloudTrail log akses server, Anda mungkin melihat permintaan yang menggunakan titik akhir global lama. Dalam contoh ini, nama bucket adalah `amzn-s3-demo-bucket1` dan titik akhir global legasi adalah:

```
https://amzn-s3-demo-bucket1.s3.amazonaws.com
```

Permintaan dengan cara hosting virtual untuk AS Timur (Virginia Utara)

Permintaan yang dibuat dengan titik akhir global legasi secara default ke Wilayah AS Timur (Virginia Utara). Oleh karena itu, titik akhir global legasi terkadang digunakan sebagai pengganti titik akhir Wilayah untuk AS Timur (Virginia Utara). Jika Anda membuat bucket di AS Timur (Virginia Utara) dan menggunakan titik akhir global, Amazon S3 mengirimkan permintaan Anda ke Wilayah ini secara default.

Permintaan dengan cara hosting virtual untuk Wilayah Lain

Titik akhir global legasi juga digunakan untuk permintaan dengan cara hosting virtual di Wilayah lain yang didukung. Jika Anda membuat bucket di Wilayah yang diluncurkan sebelum 20 Maret 2019, dan menggunakan titik akhir global legasi, Amazon S3 memperbarui catatan DNS untuk mengubah rute permintaan ke lokasi yang benar, yang mungkin memakan waktu. Sementara itu, aturan standar berlaku, dan permintaan dengan cara hosting virtual Anda dimasukkan ke Wilayah AS Timur (Virginia Utara). Amazon S3 kemudian mengalihkannya kembali dengan HTTP 307 Temporary Redirect ke Wilayah yang benar.

Untuk bucket S3 di Wilayah yang diluncurkan setelah 20 Maret 2019, server DNS tidak merutekan permintaan Anda langsung ke Wilayah AWS tempat bucket Anda berada. Sebagai gantinya, server mengembalikan pesan kesalahan HTTP 400 Bad Request. Untuk informasi selengkapnya, lihat Membuat permintaan di Referensi API Amazon S3.

Permintaan cara jalur

Untuk Wilayah AS Timur (Virginia Utara), Anda dapat menggunakan titik akhir global legasi untuk permintaan cara jalur.

Untuk semua Wilayah lainnya, sintaks cara jalur mengharuskan Anda menggunakan titik akhir spesifik-Wilayah saat mencoba mengakses bucket. Jika Anda mencoba mengakses bucket dengan titik akhir global lama atau titik akhir lain yang berbeda dari yang untuk Wilayah tempat bucket berada, Anda akan menerima kesalahan Pengalihan Permanen kode respons HTTP 301 dan pesan yang menunjukkan URI yang benar untuk sumber daya Anda. Misalnya, jika Anda menggunakan `https://s3.amazonaws.com/bucket-name` bucket yang dibuat di Wilayah AS Barat (Oregon), Anda akan menerima kesalahan Pengalihan Permanen HTTP 301.

Membuat ember tujuan umum

Untuk mengunggah data Anda ke Amazon S3, Anda harus terlebih dahulu membuat bucket tujuan umum Amazon S3 di salah satu. Wilayah AWS Akun AWS yang membuat bucket yang memilikinya. Saat Anda membuat bucket, Anda harus memilih nama bucket dan Wilayah. Selama proses pembuatan, Anda dapat memilih opsi manajemen penyimpanan lain untuk bucket secara opsional.

> **Important**
>
> Setelah membuat bucket, Anda tidak dapat mengubah nama bucket, pemilik bucket, atau Region. Untuk informasi selengkapnya tentang bucket, lihat the section called “Peraturan penamaan”.

Secara default, Anda dapat membuat hingga 10.000 ember tujuan umum per Akun AWS. Untuk meminta peningkatan kuota untuk bucket tujuan umum, kunjungi konsol Service Quotas.

Anda dapat menyimpan berapa pun jumlah objek dalam sebuah bucket. Untuk daftar pembatasan dan batasan yang terkait dengan bucket tujuan umum Amazon S3, lihat. Kuota, batasan, dan batasan bucket tujuan umum

Pengaturan bucket tujuan umum

Saat membuat bucket tujuan umum, Anda dapat menggunakan setelan berikut untuk mengontrol berbagai aspek perilaku bucket:

- Kepemilikan Objek S3 — Kepemilikan Objek S3 adalah setelan tingkat ember Amazon S3 yang dapat Anda gunakan untuk mengontrol kepemilikan objek yang diunggah ke bucket Anda dan

untuk menonaktifkan atau mengaktifkan daftar kontrol akses (). ACLs Secara default, Kepemilikan Objek disetel ke setelan diberlakukan pemilik Bucket, dan semuanya ACLs dinonaktifkan. Dengan ACLs dinonaktifkan, pemilik bucket memiliki setiap objek di bucket dan mengelola akses ke data secara eksklusif dengan menggunakan kebijakan. Untuk informasi selengkapnya, lihat Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.

- Kunci Objek S3 - Kunci Objek S3 dapat membantu mencegah objek Amazon S3 dihapus atau ditimpa untuk jangka waktu tertentu atau tanpa batas waktu. Object Lock menggunakan model write-once-read-many(WORM) untuk menyimpan objek. Anda dapat menggunakan Object Lock untuk membantu memenuhi persyaratan peraturan yang memerlukan penyimpanan WORM, atau untuk menambahkan lapisan perlindungan lain terhadap perubahan atau penghapusan objek. Untuk informasi selengkapnya, lihat the section called “Mengunci objek”.

Setelah membuat bucket tujuan umum, atau saat membuat bucket tujuan umum menggunakan konsol Amazon S3, Anda juga dapat menggunakan pengaturan berikut untuk mengontrol aspek lain dari perilaku bucket Anda:

- S3 Block Public Access — Fitur S3 Block Public Access menyediakan pengaturan untuk titik akses, bucket, dan akun untuk membantu Anda mengelola akses publik ke sumber daya Amazon S3. Secara bawaan, bucket baru, titik akses, dan objek baru tidak mengizinkan akses publik. Namun, pengguna dapat memodifikasi kebijakan bucket, kebijakan titik akses, atau izin objek untuk memungkinkan akses publik. Pengaturan S3 Blokir Akses Publik menolak kebijakan-kebijakan dan izin tersebut sehingga Anda dapat membatasi akses publik ke sumber daya. Untuk informasi selengkapnya, lihat the section called “Memblokir akses publik”.
- S3 Versioning — Versioning adalah cara menyimpan beberapa varian objek dalam bucket yang sama. Anda dapat menggunakan Penentuan Versi untuk menyimpan, mengambil, dan memulihkan setiap versi dari setiap objek yang disimpan dalam bucket Anda. Dengan Penentuan Versi, Anda dapat dengan mudah memulihkan dari tindakan pengguna yang tidak diinginkan, serta kegagalan aplikasi. Secara default, pembuatan versi dinonaktifkan untuk bucket. Untuk informasi selengkapnya, lihat the section called “Mempertahankan beberapa versi objek”.
- Enkripsi default — Anda dapat mengatur jenis enkripsi default untuk semua objek di bucket Anda. Enkripsi di sisi server dengan Amazon S3 managed keys (SSE-S3) adalah tingkat dasar konfigurasi enkripsi untuk setiap bucket di Amazon S3. Semua objek baru yang diunggah ke bucket S3 secara otomatis dienkripsi dengan SSE-S3 sebagai tingkat dasar enkripsi. Jika Anda ingin menggunakan jenis enkripsi default yang berbeda, Anda dapat menentukan enkripsi sisi server dengan AWS Key Management Service () kunci (SSE-KMS AWS KMS), enkripsi sisi server dua lapis dengan kunci (DSSE-KMS), atau enkripsi sisi server dengan AWS KMS kunci yang

disediakan pelanggan (SSE-C) untuk mengenkripsi data Anda. Untuk informasi selengkapnya, lihat Mengatur perilaku enkripsi sisi server default untuk bucket Amazon S3.

Anda dapat menggunakan konsol Amazon S3, Amazon S3 REST API AWS Command Line Interface ,AWS CLI(), AWS SDKs atau untuk membuat bucket tujuan umum. Untuk informasi selengkapnya tentang izin yang diperlukan untuk membuat bucket tujuan umum, lihat CreateBucketdi Referensi API Amazon Simple Storage Service.

Jika Anda mengalami masalah saat membuat bucket Amazon S3, lihat Bagaimana cara memecahkan masalah kesalahan saat membuat bucket Amazon S3? pada AWS re:Post.

Menggunakan konsol S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi di bagian atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah tempat Anda ingin membuat ember.

   > **Note**
   >
   > - Setelah membuat bucket, Anda tidak dapat mengubah Region.
   > - Untuk meminimalkan latensi dan biaya serta memenuhi persyaratan regulasi, pilih Wilayah yang dekat dengan Anda. Objek yang disimpan di Wilayah tidak pernah keluar dari Wilayah kecuali Anda secara tegas mentransfer atau mereplikasinya ke Wilayah lain. Untuk daftar Amazon S3 Wilayah AWS, lihat Layanan AWS titik akhir di. Referensi Umum Amazon Web Services

3. Di panel navigasi kiri, pilih Bucket tujuan umum.
4. Pilih Buat bucket. Halaman Buat bucket terbuka.
5. Untuk Nama bucket, masukkan nama untuk bucket Anda.

   Nama bucket harus:

   - Unik dalam partisi. Partisi adalah pengelompokan Wilayah. AWS saat ini memiliki tiga partisi: `aws` (Kawasan komersial), `aws-cn` (Wilayah Tiongkok), dan `aws-us-gov` (AWS GovCloud (US) Regions).
   - Panjangnya antara 3 hingga 63 karakter.

- Hanya terdiri dari huruf kecil, angka, titik (.), dan tanda hubung (). - Untuk kompatibilitas terbaik, sebaiknya hindari penggunaan period (.) dalam nama bucket, kecuali bucket yang hanya digunakan untuk hosting situs statis.
- Dimulai dan diakhiri dengan huruf atau angka.
- Untuk daftar lengkap aturan penamaan ember, lihat. Aturan penamaan bucket tujuan umum

> **Important**
>
> - Setelah membuat bucket, Anda tidak dapat mengubah namanya.
> - Jangan sertakan informasi sensitif dalam nama bucket. Nama bucket terlihat dalam URLs yang menunjuk objek dalam bucket.

6. (Opsional) Di bawah Konfigurasi umum, Anda dapat memilih untuk menyalin setelan bucket yang ada ke bucket baru Anda. Jika Anda tidak ingin menyalin pengaturan bucket yang ada, lewati ke langkah berikutnya.

> **Note**
>
> Opsi ini:
>
> - Tidak tersedia di AWS CLI dan hanya tersedia di konsol Amazon S3
> - Tidak menyalin kebijakan bucket dari bucket yang ada ke bucket baru

Untuk menyalin setelan bucket yang ada, di bagian Salin setelan dari bucket yang ada, pilih Pilih bucket. Jendela Select bucket terbuka. Temukan bucket dengan pengaturan yang ingin Anda salin, lalu pilih Pilih bucket. Jendela Choose bucket ditutup, dan jendela Create bucket terbuka kembali.

Di bawah Salin pengaturan dari bucket yang ada, Anda sekarang melihat nama bucket yang Anda pilih. Pengaturan bucket baru Anda sekarang cocok dengan pengaturan bucket yang Anda pilih. Jika Anda ingin menghapus pengaturan yang disalin, pilih Pulihkan default. Tinjau setelan bucket yang tersisa di halaman Buat bucket. Jika Anda tidak ingin membuat perubahan apa pun, Anda dapat melompat ke langkah terakhir.

7. Di bawah Kepemilikan Objek, untuk menonaktifkan atau mengaktifkan ACLs dan mengontrol kepemilikan objek yang diunggah di bucket Anda, pilih salah satu setelan berikut:

ACLs dinonaktifkan

- Pemilik bucket diberlakukan (default) — ACLs dinonaktifkan, dan pemilik bucket secara otomatis memiliki dan memiliki kontrol penuh atas setiap objek di bucket tujuan umum. ACLs tidak lagi memengaruhi izin akses ke data di bucket tujuan umum S3. Bucket menggunakan kebijakan secara eksklusif untuk menentukan kontrol akses.

  Secara default, ACLs dinonaktifkan. Mayoritas kasus penggunaan modern di Amazon S3 tidak lagi memerlukan penggunaan. ACLs Kami menyarankan agar Anda tetap ACLs dinonaktifkan, kecuali dalam keadaan yang tidak biasa di mana Anda harus mengontrol akses untuk setiap objek secara individual. Untuk informasi selengkapnya, lihat Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.

ACLs diaktifkan

- Pemilik bucket yang dipilih—Pemilik bucket memiliki dan diberikan kendali penuh atas objek baru yang ditulis akun lain ke bucket dengan ACL `bucket-owner-full-control` yang dibatasi.

  Jika Anda menerapkan pengaturan Pemilik bucket yang dipilih, agar semua unggahan Amazon S3 menyertakan ACL `bucket-owner-full-control` yang terekam, Anda dapat menambahkan kebijakan bucket yang hanya mengizinkan unggahan objek yang menggunakan ACL ini.
- Penulis objek — Akun AWS Yang mengunggah objek memiliki objek, memiliki kontrol penuh atasnya, dan dapat memberikan pengguna lain akses ke sana melalui. ACLs

> **Note**
>
> Pengaturan default-nya adalah Pemilik Bucket yang diberlakukan. Untuk menerapkan pengaturan default dan tetap ACLs dinonaktifkan, hanya `s3:CreateBucket` izin yang diperlukan. Untuk mengaktifkan ACLs, Anda harus memiliki `s3:PutBucketOwnershipControls` izin.

8. Di bawah Pengaturan Blokir Akses Publik untuk bucket ini, pilih pengaturan Blokir Akses Publik yang ingin Anda terapkan ke bucket.

Secara default, semua pengaturan Blokir Akses Publik untuk bucket direktori diaktifkan. Kami menyarankan Anda tetap mengaktifkan semua pengaturan, kecuali Anda tahu bahwa Anda perlu menonaktifkan satu atau beberapa pengaturan untuk kasus penggunaan spesifik Anda. Untuk informasi lebih lanjut tentang pemblokiran akses publik, lihat Melakukan blok akses publik ke penyimpanan Amazon S3 Anda.

> **Note**
>
> Untuk mengaktifkan semua pengaturan Blokir Akses Publik, hanya izin `s3:CreateBucket` yang diperlukan. Untuk mematikan pengaturan Blokir Akses Publik, Anda harus memiliki izin `s3:PutBucketPublicAccessBlock`.

9. (Opsional) Secara default, Bucket Versioning dinonaktifkan. Penentuan Versi adalah cara menyimpan beberapa varian objek dalam bucket yang sama. Anda dapat menggunakan Penentuan Versi untuk menyimpan, mengambil, dan memulihkan setiap versi dari setiap objek yang disimpan dalam bucket Anda. Dengan versioning, Anda dapat lebih mudah memulihkan dari tindakan pengguna yang tidak diinginkan dan kegagalan aplikasi. Untuk informasi selengkapnya tentang penentuan versi, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.

   Untuk mengaktifkan pembuatan versi di bucket, pilih Aktifkan.

10. (Opsional) Di bawah Tanda, Anda dapat memilih untuk menambahkan tanda ke bucket Anda. Dengan alokasi AWS biaya, Anda dapat menggunakan tag bucket untuk membubuhi keterangan penagihan penggunaan bucket. Tanda adalah pasangan nilai kunci yang mewakili label yang Anda tetapkan ke bucket. Untuk informasi selengkapnya, lihat the section called “Menggunakan tanda alokasi biaya”.

    Untuk menambahkan tanda bucket, masukkan Kunci dan secara opsional Nilai, lalu pilih Tambahkan Tanda.

11. Untuk mengonfigurasi enkripsi Default, di bawah Jenis enkripsi, pilih salah satu dari berikut ini:

    - Enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3)
    - Enkripsi sisi server dengan AWS Key Management Service kunci (SSE-KMS)
    - Enkripsi sisi server dua lapis dengan kunci () (DSSE-KMS AWS Key Management Service)AWS KMS

> **Important**
>
> Jika Anda menggunakan opsi SSE-KMS atau DSSE-KMS untuk konfigurasi enkripsi default Anda, Anda tunduk pada kuota permintaan per detik (RPS) sebesar. AWS KMSUntuk informasi selengkapnya tentang AWS KMS kuota dan cara meminta kenaikan kuota, lihat Kuota di Panduan Pengembang AWS Key Management Service .

Bucket dan objek baru dienkripsi dengan menggunakan enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3) sebagai tingkat dasar konfigurasi enkripsi. Untuk informasi selengkapnya tentang enkripsi default, lihat Mengatur perilaku enkripsi sisi server default untuk bucket Amazon S3. Untuk informasi selengkapnya tentang SSE-S3, lihat. Menggunakan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3)

Untuk informasi selengkapnya tentang penggunaan enkripsi sisi server untuk mengenkripsi data Anda, lihat. the section called “Enkripsi data”

12. Jika Anda memilih enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3) atau enkripsi sisi server dual-layer dengan kunci () (DSSE-KMS), lakukan hal berikut: AWS Key Management Service AWS KMS

    a. Di bawah AWS KMS kunci, tentukan kunci KMS Anda dengan salah satu cara berikut ini:

    - Untuk memilih dari daftar kunci KMS yang tersedia, pilih Pilih dari Anda AWS KMS keys, dan pilih kunci KMS Anda dari daftar kunci yang tersedia.

      Kunci Kunci yang dikelola AWS (`aws/s3`) dan kunci terkelola pelanggan Anda muncul dalam daftar ini. Untuk informasi selengkapnya tentang CMK, lihat Kunci pelanggan dan AWS kunci di AWS Key Management Service Panduan Pengembang.

    - Untuk memasukkan ARN kunci KMS, pilih Masukkan AWS KMS key ARN, dan masukkan ARN kunci KMS Anda di bidang yang muncul.
    - Untuk membuat kunci terkelola pelanggan baru di AWS KMS konsol, pilih Buat kunci KMS.

      Untuk informasi selengkapnya tentang membuat AWS KMS key, lihat Membuat kunci di Panduan AWS Key Management Service Pengembang.

> **Important**
>
> Anda hanya dapat menggunakan tombol KMS yang tersedia Wilayah AWS sama dengan bucket. Konsol Amazon S3 hanya mencantumkan kunci 100 KMS pertama di Wilayah yang sama dengan bucket. Untuk menggunakan kunci KMS yang tidak terdaftar, Anda harus memasukkan ARN kunci KMS Anda. Jika Anda ingin menggunakan kunci KMS yang dimiliki oleh akun yang berbeda, Anda harus terlebih dahulu memiliki izin untuk menggunakan kunci tersebut, dan kemudian Anda harus memasukkan ARN kunci KMS. Untuk informasi selengkapnya tentang izin lintas akun untuk kunci KMS, lihat Membuat kunci KMS yang dapat digunakan akun lain di Panduan Pengembang AWS Key Management Service . Untuk informasi lebih lanjut tentang SSE-KMS, lihat Menentukan enkripsi di sisi server dengan AWS KMS (SSE-KMS). Untuk informasi lebih lanjut tentang DSSE-KMS, lihat the section called “Enkripsi di sisi server dua lapis (DSSE-KMS)”.
>
> Saat Anda menggunakan enkripsi sisi server AWS KMS key untuk Amazon S3, Anda harus memilih kunci KMS enkripsi simetris. Amazon S3 hanya mendukung kunci KMS enkripsi simetris dan tidak mendukung kunci KMS asimetris. Untuk informasi selengkapnya, lihat Mengidentifikasi tombol KMS simetris dan asimetris dalam Panduan Pengembang AWS Key Management Service .

b. Saat mengonfigurasi bucket untuk menggunakan enkripsi default dengan SSE-KMS, Anda juga dapat menggunakan S3 Bucket Keys. S3 Bucket Keys menurunkan biaya enkripsi dengan mengurangi lalu lintas permintaan dari Amazon S3 ke. AWS KMS Untuk informasi selengkapnya, lihat Mengurangi biaya SSE-KMS dengan Kunci Bucket Amazon S3. Kunci Bucket S3 tidak didukung untuk DSSE-KMS.

   Secara default, Kunci Bucket S3 diaktifkan di konsol Amazon S3. Sebaiknya biarkan Kunci Bucket S3 diaktifkan untuk menurunkan biaya Anda. Untuk menonaktifkan Kunci Bucket S3 untuk bucket Anda, di bawah Kunci Bucket, pilih Nonaktifkan.

13. (Opsional) Kunci Objek S3 membantu melindungi objek baru agar tidak dihapus atau ditimpa. Untuk informasi selengkapnya, lihat Mengunci objek dengan Object Lock. Jika Anda ingin mengaktifkan S3 Object Lock, lakukan hal berikut:

    a. Pilih Pengaturan lanjutan.

> **Important**
>
> Mengaktifkan Object Lock secara otomatis memungkinkan pembuatan versi untuk bucket. Setelah mengaktifkan dan berhasil membuat bucket, Anda juga harus mengonfigurasi pengaturan penyimpanan default Object Lock dan penahanan legal pada tab Properties bucket.

b. Jika Anda ingin mengaktifkan Kunci Objek, pilih Aktifkan, baca peringatan yang muncul, lalu setujui.

> **Note**
>
> Untuk membuat bucket yang diaktifkan Object Lock, Anda harus memiliki izin berikut:`s3:CreateBucket`,`s3:PutBucketVersioning`, dan`s3:PutBucketObjectLockConfiguration`.

14. Pilih Buat bucket.

Menggunakan AWS SDKs

Ketika Anda menggunakan AWS SDKs untuk membuat bucket tujuan umum, Anda harus membuat klien dan kemudian menggunakan klien untuk mengirim permintaan untuk membuat bucket. Sebagai praktik terbaik, Anda harus membuat klien dan bucket di Wilayah AWS yang sama. Jika Anda tidak menentukan Wilayah saat membuat klien atau bucket, Amazon S3 menggunakan Wilayah default AS Timur (Virginia Utara). Jika Anda ingin membatasi pembuatan bucket ke yang spesifik Wilayah AWS, gunakan kunci LocationConstraintkondisi.

Untuk membuat klien untuk mengakses titik akhir dual-stack, Anda harus menentukan. Wilayah AWSUntuk informasi selengkapnya, lihat Menggunakan titik akhir tumpukan ganda Amazon S3 di Referensi API Amazon S3. Untuk daftar yang tersedia Wilayah AWS, lihat titik akhir Amazon Simple Storage Service dan kuota di. Referensi Umum AWS

Saat Anda membuat klien, Wilayah akan memetakan ke titik akhir spesifik Wilayah. Klien menggunakan titik akhir ini untuk berkomunikasi dengan Amazon S3: `s3.region.amazonaws.com`. Jika Wilayah Anda diluncurkan setelah 20 Maret 2019, klien dan bucket Anda harus berada di dalam Wilayah yang sama. Namun, Anda dapat menggunakan klien di Wilayah AS Timur (Virginia Utara)

untuk membuat bucket di setiap Wilayah yang diluncurkan sebelum 20 Maret 2019. Untuk informasi selengkapnya, lihat Titik akhir legasi.

Contoh kode AWS SDK ini melakukan tugas-tugas berikut:

- Membuat klien dengan secara eksplisit menentukan Wilayah AWS–Dalam contoh, klien menggunakan titik akhir `s3.us-west-2.amazonaws.com` untuk berkomunikasi dengan Amazon S3. Anda dapat menentukan Wilayah AWS. Untuk daftar Wilayah AWS, lihat Wilayah dan titik akhir dalam Referensi AWS Umum.
- Kirim permintaan pembuatan bucket dengan menentukan hanya nama bucket–Klien mengirimkan permintaan ke Amazon S3 untuk membuat bucket di Wilayah tempat Anda membuat klien.
- Mengambil informasi tentang lokasi bucket — Amazon S3 menyimpan informasi lokasi bucket di subsumber daya lokasi yang terkait dengan bucket.

Untuk contoh dan contoh AWS SDK tambahan dalam bahasa lain, lihat Menggunakan CreateBucket dengan AWS SDK atau CLI di Referensi API Layanan Penyimpanan Sederhana Amazon.

Java

Example Buat bucket yang menggunakan pengenal unik global (GUID) di nama bucket

Contoh berikut menunjukkan kepada Anda cara membuat dengan GUID di akhir nama bucket di Wilayah AS Timur (Virginia Utara) (`us-east-1`) dengan menggunakan. AWS SDK untuk Java Untuk menggunakan contoh ini, ganti *`user input placeholders`*dengan informasi Anda sendiri. Untuk informasi tentang lainnya AWS SDKs, lihat Alat untuk Dibangun AWS.

```
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.Bucket;
import com.amazonaws.services.s3.model.CreateBucketRequest;

import java.util.List;
import java.util.UUID;

public class CreateBucketWithUUID {
    public static void main(String[] args) {
        final AmazonS3 s3 =
 AmazonS3ClientBuilder.standard().withRegion(Regions.US_EAST_1).build();
```

```
        String bucketName = "amzn-s3-demo-bucket" +
 UUID.randomUUID().toString().replace("-", "");
        CreateBucketRequest createRequest = new CreateBucketRequest(bucketName);
        System.out.println(bucketName);
        s3.createBucket(createRequest);
    }
}
```

Example Buat ember tujuan umum

Contoh ini menunjukkan cara membuat bucket Amazon S3 dengan menggunakan. AWS SDK untuk Java Untuk petunjuk cara membuat dan menguji sampel yang berfungsi, lihat Panduan AWS SDK for Java 2.x Pengembang.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.CreateBucketRequest;
import com.amazonaws.services.s3.model.GetBucketLocationRequest;

import java.io.IOException;

public class CreateBucket2 {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            if (!s3Client.doesBucketExistV2(bucketName)) {
                // Because the CreateBucketRequest object doesn't specify a region,
 the
                // bucket is created in the region specified in the client.
                s3Client.createBucket(new CreateBucketRequest(bucketName));
```

```
                // Verify that the bucket was created by retrieving it and checking
 its
                // location.
                String bucketLocation = s3Client.getBucketLocation(new
 GetBucketLocationRequest(bucketName));
                System.out.println("Bucket location: " + bucketLocation);
            }
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

Untuk informasi tentang cara membuat dan menguji sampel yang berfungsi, lihat Referensi API AWS SDK for .NET Versi 3.

Example

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using Amazon.S3.Util;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class CreateBucketTest
    {
        private const string bucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
```

```
        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            CreateBucketAsync().Wait();
        }

        static async Task CreateBucketAsync()
        {
            try
            {
                if (!(await AmazonS3Util.DoesS3BucketExistAsync(s3Client,
bucketName)))
                {
                    var putBucketRequest = new PutBucketRequest
                    {
                        BucketName = bucketName,
                        UseClientRegion = true
                    };

                    PutBucketResponse putBucketResponse = await
s3Client.PutBucketAsync(putBucketRequest);
                }
                // Retrieve the bucket location.
                string bucketLocation = await FindBucketLocationAsync(s3Client);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
        }
        static async Task<string> FindBucketLocationAsync(IAmazonS3 client)
        {
            string bucketLocation;
            var request = new GetBucketLocationRequest()
            {
                BucketName = bucketName
            };
```

```
            GetBucketLocationResponse response = await
 client.GetBucketLocationAsync(request);
            bucketLocation = response.Location.ToString();
            return bucketLocation;
        }
    }
}
```

Ruby

Untuk informasi tentang cara membuat dan menguji sampel yang berfungsi, lihat AWS SDK for Ruby - Versi 3.

Example

```
require 'aws-sdk-s3'

# Wraps Amazon S3 bucket actions.
class BucketCreateWrapper
  attr_reader :bucket

  # @param bucket [Aws::S3::Bucket] An Amazon S3 bucket initialized with a name.
 This is a client-side object until
  #                                 create is called.
  def initialize(bucket)
    @bucket = bucket
  end

  # Creates an Amazon S3 bucket in the specified AWS Region.
  #
  # @param region [String] The Region where the bucket is created.
  # @return [Boolean] True when the bucket is created; otherwise, false.
  def create?(region)
    @bucket.create(create_bucket_configuration: { location_constraint: region })
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't create bucket. Here's why: #{e.message}"
    false
  end

  # Gets the Region where the bucket is located.
  #
  # @return [String] The location of the bucket.
```

```
  def location
    if @bucket.nil?
      'None. You must create a bucket before you can get its location!'
    else
      @bucket.client.get_bucket_location(bucket: @bucket.name).location_constraint
    end
  rescue Aws::Errors::ServiceError => e
    "Couldn't get the location of #{@bucket.name}. Here's why: #{e.message}"
  end
end

# Example usage:
def run_demo
  region = "us-west-2"
  wrapper = BucketCreateWrapper.new(Aws::S3::Bucket.new("amzn-s3-demo-bucket-
#{Random.uuid}"))
  return unless wrapper.create?(region)

  puts "Created bucket #{wrapper.bucket.name}."
  puts "Your bucket's region is: #{wrapper.location}"
end

run_demo if $PROGRAM_NAME == __FILE__
```

Menggunakan AWS CLI

AWS CLI Contoh berikut membuat bucket tujuan umum di Wilayah Wilayah () AS Barat (California) dengan contoh nama bucket yang menggunakan pengenal unik global (GUID). `us-west-1` Untuk menggunakan perintah contoh ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3api create-bucket \
    --bucket amzn-s3-demo-bucket1$(uuidgen | tr -d - | tr '[:upper:]' '[:lower:]' ) \
    --region us-west-1 \
    --create-bucket-configuration LocationConstraint=us-west-1
```

Untuk informasi selengkapnya dan contoh tambahan, lihat create-bucketdi AWS CLI Command Reference.

Melihat properti untuk bucket tujuan umum S3

Anda dapat melihat properti untuk bucket Amazon S3 apa pun yang Anda miliki. Pengaturan ini meliputi:

- Bucket Versioning — Simpan beberapa versi objek dalam satu bucket tujuan umum dengan menggunakan pembuatan versi. Secara default, Penentuan Versi dinonaktifkan untuk bucket yang baru. Untuk informasi tentang Penentuan Versi, lihat Mengaktifkan Penentuan Versi pada bucket.
- Tag — Dengan alokasi AWS biaya, Anda dapat menggunakan tag bucket untuk membubuhi keterangan penagihan penggunaan bucket tujuan umum. Tanda adalah pasangan nilai kunci yang mewakili label yang Anda tetapkan ke bucket. Untuk informasi selengkapnya, lihat Menggunakan tanda alokasi biaya bucket S3.
- Enkripsi default–Mengaktifkan enkripsi default memberikan Anda enkripsi di sisi server otomatis. Amazon S3 mengenkripsi sebuah objek sebelum menyimpannya ke disk, dan mendekripsi objek tersebut saat Anda mengunduhnya. Untuk informasi selengkapnya, lihat Mengatur perilaku enkripsi sisi server default untuk bucket Amazon S3.
- Pencatatan akses server — Dapatkan catatan terperinci untuk permintaan yang dibuat ke bucket tujuan umum Anda dengan pencatatan akses server. Secara default, Amazon S3 tidak mengumpulkan log akses server. Untuk informasi tentang mengaktifkan pencatatan akses server, lihatMengaktifkan pencatatan akses server Amazon S3.
- AWS CloudTrail peristiwa data - Gunakan CloudTrail untuk mencatat peristiwa data. Secara default, jejak tidak mencatat peristiwa data. Biaya tambahan berlaku untuk peristiwa data. Untuk informasi lebih lanjut, lihat Peristiwa Pencatatan Data untuk Pelacakan dalam AWS CloudTrail Panduan Pengguna.
- Pemberitahuan peristiwa — Aktifkan peristiwa bucket tujuan umum Amazon S3 tertentu untuk mengirim pesan notifikasi ke tujuan kapan pun peristiwa terjadi. Untuk informasi selengkapnya, lihat Mengaktifkan dan mengonfigurasi notifikasi peristiwa menggunakan konsol Amazon S3.
- Transfer acceleration–Memungkinkan pentransferan file yang cepat, mudah, dan aman melalui jarak jauh antara klien Anda dan bucket S3. Untuk informasi lebih lanjut tentang Transfer Acceleration, lihat Mengaktifkan dan menggunakan S3 Transfer Acceleration.
- Kunci Objek–Gunakan Kunci Objek S3 untuk mencegah objek terhapus atau ditimpa selama jangka waktu tertentu, atau tanpa batas waktu. Untuk informasi selengkapnya, lihat Mengunci objek dengan Object Lock.

- Requester Pays — Aktifkan Requester Pays jika Anda ingin pemohon (bukan pemilik bucket tujuan umum) membayar permintaan dan transfer data. Untuk informasi selengkapnya, lihat Menggunakan Requester Pays ember tujuan umum untuk transfer dan penggunaan penyimpanan.
- Hosting situs web statis–Anda dapat menghosting situs web statis di Amazon S3. Untuk informasi selengkapnya, lihat Hosting situs web statis menggunakan Amazon S3.

Anda dapat melihat properti bucket menggunakan AWS Management Console, AWS CLI, atau AWS SDKs

Menggunakan konsol S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum atau ember Direktori.
3. Dalam daftar bucket, pilih nama bucket yang ingin Anda lihat propertinya.
4. Pilih tab Properti.
5. Pada halaman Properties, Anda dapat mengonfigurasi properti di atas untuk bucket.

Menggunakan AWS CLI

Lihat properti bucket dengan AWS CLI

Perintah berikut menunjukkan bagaimana Anda dapat menggunakan daftar properti bucket tujuan umum yang berbeda. AWS CLI

Berikut ini mengembalikan set tag yang terkait dengan bucket*amzn-s3-demo-bucket1*. Untuk informasi selengkapnya tentang tag bucket, lihatMenggunakan tanda alokasi biaya bucket S3.

```
aws s3api get-bucket-tagging --bucket amzn-s3-demo-bucket1
```

Untuk informasi dan contoh selengkapnya, lihat get-bucket-tagging di AWS CLI Referensi Perintah.

Berikut ini mengembalikan status pembuatan versi bucket. *amzn-s3-demo-bucket1* Untuk informasi tentang pembuatan versi bucket, lihat. Mempertahankan beberapa versi objek dengan S3 Versioning

```
aws s3api get-bucket-versioning --bucket amzn-s3-demo-bucket1
```

Untuk informasi dan contoh selengkapnya, lihat get-bucket-versioning di AWS CLI Referensi Perintah.

Berikut ini mengembalikan konfigurasi enkripsi default untuk bucket*amzn-s3-demo-bucket1*. Secara default, semua bucket memiliki konfigurasi enkripsi default yang menggunakan enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3). Untuk informasi tentang enkripsi default bucket, lihatMengatur perilaku enkripsi sisi server default untuk bucket Amazon S3.

```
aws s3api get-bucket-encryption --bucket amzn-s3-demo-bucket1
```

Untuk informasi dan contoh selengkapnya, lihat get-bucket-encryption di AWS CLI Referensi Perintah.

Berikut ini mengembalikan konfigurasi notifikasi bucket*amzn-s3-demo-bucket1*. Untuk informasi tentang notifikasi acara bucket, lihatNotifikasi Peristiwa Amazon S3.

```
aws s3api get-bucket-notification-configuration --bucket amzn-s3-demo-bucket1
```

Untuk informasi dan contoh selengkapnya, lihat get-bucket-notification-configuration di AWS CLI Referensi Perintah.

Berikut ini mengembalikan status logging untuk bucket*amzn-s3-demo-bucket1*. Untuk informasi tentang pencatatan bucket, lihatPencatatan permintaan dengan pencatatan akses server.

```
aws s3api get-bucket-logging --bucket amzn-s3-demo-bucket1
```

Untuk informasi dan contoh selengkapnya, lihat get-bucket-logging di AWS CLI Referensi Perintah.

Menggunakan AWS SDKs

Untuk contoh cara menampilkan properti bucket tujuan umum dengan AWS SDKs, seperti pembuatan versi, tag, dan lainnya, lihat Contoh kode di Referensi API Amazon S3.

Untuk informasi umum tentang penggunaan yang berbeda AWS SDKs, lihat Mengembangkan dengan Amazon S3 menggunakan Referensi API Amazon S3 AWS SDKs di Amazon S3.

Daftar ember tujuan umum Amazon S3

Untuk mengembalikan daftar ember tujuan umum yang Anda miliki, Anda dapat menggunakannya ListBuckets. Anda dapat membuat daftar bucket Anda dengan menggunakan konsol Amazon

S3, AWS Command Line Interface the, atau. AWS SDKs Untuk `ListBuckets` permintaan yang menggunakan AWS CLI, AWS SDKs, dan Amazon S3 REST API, Akun AWS yang menggunakan kuota layanan default untuk bucket (10.000 bucket), mendukung permintaan paginasi dan unpaginasi. Terlepas dari berapa banyak bucket yang Anda miliki di akun Anda, Anda dapat membuat ukuran halaman antara 1 dan 10.000 bucket untuk mencantumkan semua bucket Anda. Untuk permintaan paginasi, `ListBuckets` permintaan mengembalikan nama bucket dan yang sesuai Wilayah AWS untuk setiap bucket. Contoh berikut AWS Command Line Interface dan AWS SDK menunjukkan cara menggunakan pagination dalam permintaan Anda`ListBuckets`. Perhatikan bahwa beberapa AWS SDKs membantu dengan pagination.

Izin

Untuk membuat daftar semua ember tujuan umum Anda, Anda harus memiliki `s3:ListAllMyBuckets` izin. Jika Anda mengalami `HTTP Access Denied (403 Forbidden)` kesalahan, lihat. Memecahkan masalah kesalahan akses ditolak (403 Forbidden) di Amazon S3

> **Important**
>
> Kami sangat menyarankan hanya menggunakan `ListBuckets` permintaan paginasi. `ListBuckets`Permintaan tanpa paginasi hanya didukung untuk Akun AWS disetel ke kuota bucket tujuan umum default sebesar 10.000. Jika Anda memiliki kuota bucket tujuan umum yang disetujui di atas 10.000, Anda harus mengirim `ListBuckets` permintaan paginasi untuk mencantumkan bucket akun Anda. Semua `ListBuckets` permintaan tanpa paginasi akan ditolak Akun AWS dengan kuota bucket tujuan umum lebih dari 10.000.

Menggunakan konsol S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Pada tab Bucket tujuan umum, Anda dapat melihat daftar bucket tujuan umum Anda.
4. Untuk menemukan bucket berdasarkan nama, masukkan nama bucket di bidang Temukan bucket dengan nama.

Menggunakan AWS CLI

Untuk menggunakan AWS CLI untuk menghasilkan daftar bucket tujuan umum, Anda dapat menggunakan `list-buckets` perintah `ls` atau. Contoh berikut menunjukkan cara membuat permintaan paginasi dan `list-buckets` permintaan tanpa `ls` paginasi. Untuk menggunakan contoh-contoh ini, ganti*`user input placeholders`*.

Example — Daftar semua bucket di akun Anda dengan menggunakan **`ls`** (unpaginated)

Contoh perintah berikut mencantumkan semua bucket tujuan umum di akun Anda dalam satu panggilan non-paginasi. Panggilan ini menampilkan daftar semua bucket di akun Anda (hingga 10.000 hasil):

```
$ aws s3 ls
```

Untuk informasi dan contoh selengkapnya, lihat Mendaftarkan bucket dan objek.

Example — Daftar semua bucket di akun Anda dengan menggunakan **`ls`** (paginated)

Contoh perintah berikut membuat satu atau beberapa panggilan paginasi untuk mencantumkan semua bucket tujuan umum di akun Anda, mengembalikan 100 bucket per halaman:

```
$ aws s3 ls --page-size 100
```

Untuk informasi dan contoh selengkapnya, lihat Mendaftarkan bucket dan objek.

Example — Daftar semua bucket di akun Anda (paginated)

Contoh berikut menyediakan `list-buckets` perintah paginasi untuk mencantumkan semua bucket tujuan umum di akun Anda. `--page-size`Opsi `--max-items` dan membatasi jumlah ember yang terdaftar hingga 100 per halaman.

```
$ aws s3api list-buckets /
    --max-items 100 /
    --page-size 100
```

Jika jumlah item output (`--max-items`) lebih sedikit dari jumlah total item yang dikembalikan oleh panggilan API yang mendasarinya, output menyertakan token lanjutan, yang ditentukan oleh

`starting-token` argumen, yang dapat Anda teruskan ke perintah berikutnya untuk mengambil set item berikutnya. Contoh berikut menunjukkan bagaimana menggunakan `starting-token` nilai yang dikembalikan oleh contoh sebelumnya. Anda dapat menentukan `starting-code` untuk mengambil 100 ember berikutnya.

```
$ aws s3api list-buckets /
    --max-items 100 /
    --page-size 100 /
    --starting-token eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxfQ==
```

Example — Daftar semua ember dalam Wilayah AWS (paginated)

Contoh perintah berikut menggunakan `--bucket-region` parameter untuk daftar hingga 100 bucket dalam akun yang berada di `us-east-2` Region. Permintaan yang dibuat ke titik akhir Regional yang berbeda dari nilai yang ditentukan dalam `--bucket-region` parameter tidak didukung. Misalnya, jika Anda ingin membatasi respons ke bucket Anda`us-east-2`, Anda harus membuat permintaan ke titik akhir di. `us-east-2`

```
$ aws s3api list-buckets /
    --region us-east-2 /
    --max-items 100 /
    --page-size 100 /
    --bucket-region us-east-2
```

Example — Buat daftar semua bucket yang dimulai dengan awalan nama bucket tertentu (paginated)

Contoh perintah berikut mencantumkan hingga 100 bucket yang memiliki nama yang dimulai dengan *`amzn-s3-demo-bucket`* awalan.

```
$ aws s3api list-buckets /
    --max-items 100 /
    --page-size 100 /
    --prefix amzn-s3-demo-bucket
```

Menggunakan AWS SDKs

Contoh berikut menunjukkan kepada Anda cara membuat daftar bucket tujuan umum Anda dengan menggunakan AWS SDKs

SDK for Python

Example — ListBuckets permintaan (paginated)

```
import boto3

s3 = boto3.client('s3')
response = s3.list_buckets(MaxBuckets=100)
```

Example — ListBuckets respon (paginated)

```
import boto3

s3 = boto3.client('s3')
response =
 s3.list_buckets(MaxBuckets=1,ContinuationToken="eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9
```

SDK for Java

```
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.Bucket;
import com.amazonaws.services.s3.model.ListBucketsPaginatedRequest;
import com.amazonaws.services.s3.model.ListBucketsPaginatedResult;

import java.util.List;

public class ListBuckets {
    public static void main(String[] args) {
        final AmazonS3 s3 =
 AmazonS3ClientBuilder.standard().withRegion(Regions.DEFAULT_REGION).build();
        ListBucketsPaginatedRequest listBucketsPaginatedRequest = new
 ListBucketsPaginatedRequest().withMaxBuckets(1);
        ListBucketsPaginatedResult listBucketsPaginatedResult =
 s3.listBuckets(listBucketsPaginatedRequest);
        List<Bucket> buckets = listBucketsPaginatedResult.getBuckets();
        System.out.println("Your Amazon S3 buckets are:");
        for (Bucket b : buckets) {
            System.out.println("* " + b.getName() + " region: " + b.getRegion());
        }
        System.out.println("continuation token: " +
 listBucketsPaginatedResult.getContinuationToken());
```

```
    }
}
```

SDK for Java 2.x

```
import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.Bucket;
import software.amazon.awssdk.services.s3.model.ListBucketsRequest;
import software.amazon.awssdk.services.s3.model.ListBucketsResponse;

import java.util.List;

public class ListBuckets {
    public static void main(String[] args) {
        // Create an S3 client
        S3Client s3 = S3Client.builder()
                .region(Region.US_EAST_1) // Replace with your preferred region
                .credentialsProvider(DefaultCredentialsProvider.create())
                .build();

        try {
            // List buckets
            ListBucketsRequest listBucketsRequest = ListBucketsRequest.builder()
                    .maxBuckets(10)
                    .build();
            ListBucketsResponse listBucketsResponse =
 s3.listBuckets(listBucketsRequest);
            List<Bucket> buckets = listBucketsResponse.buckets();

            // Print bucket names
            System.out.println("Your Amazon S3 buckets are:");
            for (Bucket bucket : buckets) {
                System.out.println(bucket.name());
                System.out.println(bucket.getBucketRegion());
            }

        } catch (Exception e) {
            System.err.println("Error listing buckets: " + e.getMessage());
            e.printStackTrace();
        } finally {
            // Close the S3 client to release resources
```

```
            s3.close();
        }
    }
}
```

SDK for Go

```
package main
import (
 "context"
 "fmt"
 "log"
 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/service/s3"
)
func main() {
 cfg, err := config.LoadDefaultConfig(context.TODO(), config.WithRegion("us-
east-2"))
 if err != nil {
  log.Fatal(err)
 }
 client := s3.NewFromConfig(cfg)
 maxBuckets := 1000
 resp, err := client.ListBuckets(context.TODO(), management
 portals3.ListBucketsInput{MaxBuckets: aws.Int32(int32(maxBuckets))})
 if err != nil {
  log.Fatal(err)
 }
 fmt.Println("S3 Buckets:")
 for _, bucket := range resp.Buckets {
     fmt.Println("- Name:", *bucket.Name)
     fmt.Println("-BucketRegion", *bucket.BucketRegion)
 }
 fmt.Println(resp.ContinuationToken == nil)
 fmt.Println(resp.Prefix == nil)
}
```

Mengosongkan ember tujuan umum

Anda dapat mengosongkan konten bucket tujuan umum menggunakan konsol Amazon S3, AWS SDKs, atau AWS Command Line Interface ()AWS CLI. Saat Anda mengosongkan ember tujuan

umum, Anda menghapus semua objek, tetapi Anda menyimpan ember. Setelah Anda mengosongkan bucket, tindakan ini tidak dapat dibatalkan. Objek yang ditambahkan ke bucket saat tindakan mengosongkan bucket sedang berlangsung akan dihapus. Semua objek (termasuk semua versi objek dan penanda hapus) di dalam bucket harus dihapus sebelum bucket itu sendiri dapat dihapus.

Saat Anda mengosongkan bucket tujuan umum yang mengaktifkan atau menangguhkan Versi S3, semua versi semua objek di bucket akan dihapus. Untuk informasi selengkapnya, lihat Bekerja dengan objek di dalam bucket dengan dukungan Penentuan Versi.

Saat mengosongkan bucket, sebaiknya Anda juga menghapus semua unggahan multipart yang tidak lengkap. Anda dapat menggunakan unggahan multipart untuk mengunggah objek yang sangat besar (hingga 5 TB) sebagai satu set bagian untuk meningkatkan throughput dan pemulihan yang lebih cepat dari masalah jaringan. Jika proses pengunggahan multibagian tidak selesai, bagian yang tidak lengkap tetap berada di bucket (dalam keadaan tidak dapat digunakan). Suku cadang yang tidak lengkap ini menimbulkan biaya penyimpanan sampai proses pengunggahan selesai, atau sampai bagian yang tidak lengkap dilepas. Untuk informasi selengkapnya, lihat Mengunggah dan menyalin objek menggunakan unggahan multipart di Amazon S3.

Sebagai praktik terbaik, kami merekomendasikan untuk mengonfigurasi aturan siklus hidup untuk objek kedaluwarsa dan unggahan multibagian yang tidak lengkap yang lebih lama dari jumlah hari tertentu. Saat Anda membuat aturan siklus hidup untuk mengakhiri unggahan multibagian yang tidak lengkap, kami merekomendasikan 7 hari sebagai titik awal yang baik. Untuk informasi selengkapnya, lihat Menyetel konfigurasi Siklus Hidup S3 pada bucket.

Kedaluwarsa siklus hidup adalah proses asinkron, jadi aturannya mungkin memerlukan waktu beberapa hari untuk dijalankan sebelum bucket Anda kosong. Setelah pertama kali Amazon S3 menjalankan aturan, semua objek yang memenuhi syarat untuk kedaluwarsa ditandai untuk dihapus. Anda tidak lagi dikenakan biaya untuk objek yang ditandai untuk dihapus. Untuk informasi selengkapnya, lihat Bagaimana saya mengosongkan bucket Amazon S3 menggunakan aturan konfigurasi siklus hidup?.

Menggunakan konsol S3

Anda dapat menggunakan konsol Amazon S3 untuk mengosongkan bucket tujuan umum, yang menghapus semua objek di bucket tanpa menghapus bucket.

Untuk mengosongkan bucket S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Di bucket list, pilih opsi di samping nama bucket yang ingin Anda kosongkan, lalu pilih Empty.
4. Pada halaman Bucket kosong, konfirmasikan bahwa Anda ingin mengosongkan bucket dengan memasukkan nama bucket ke dalam bidang teks, lalu pilih Kosongkan.
5. Pantau progres proses pengosongan bucket di halaman Bucket kosong: Status.

Menggunakan AWS CLI

Anda dapat mengosongkan bucket tujuan umum AWS CLI hanya menggunakan bucket jika bucket tidak mengaktifkan Bucket Versioning. Jika pembuatan versi tidak diaktifkan, Anda dapat menggunakan AWS CLI perintah `rm` (hapus) dengan `--recursive` parameter untuk mengosongkan bucket (atau menghapus subset objek dengan awalan nama kunci tertentu).

Perintah `rm` berikut menghapus objek yang memiliki prefiks nama kunci `doc`, misalnya, `doc/doc1` dan `doc/doc2`.

```
$ aws s3 rm s3://bucket-name/doc --recursive
```

Gunakan perintah berikut untuk menghapus semua objek tanpa menentukan prefiks.

```
$ aws s3 rm s3://bucket-name --recursive
```

Untuk informasi lebih lanjut, lihat Menggunakan perintah S3 tingkat tinggi dengan AWS CLI di AWS Command Line Interface Panduan Pengguna.

> **Note**
>
> Anda tidak dapat menghapus objek dari bucket dengan Penentuan Versi diaktifkan. Amazon S3 menambahkan penanda hapus saat Anda menghapus objek, yang dilakukan oleh perintah ini. Untuk informasi selengkapnya tentang Penentuan Versi Bucket S3, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.

Menggunakan AWS SDKs

Anda dapat menggunakan AWS SDKs untuk mengosongkan bucket tujuan umum atau menghapus subset objek yang memiliki awalan nama kunci tertentu.

Untuk contoh cara mengosongkan ember menggunakan AWS SDK untuk Java, lihatMenghapus bucket tujuan umum. Kode tersebut menghapus semua objek, terlepas dari apakah bucket telah mengaktifkan Penentuan Versi atau tidak, lalu menghapus bucket. Untuk mengosongkan bucket saja, pastikan bahwa Anda menghapus pernyataan yang menghapus bucket.

Untuk informasi selengkapnya tentang penggunaan lainnya AWS SDKs, lihat Alat untuk Amazon Web Services.

Menggunakan konfigurasi siklus aktif

Untuk mengosongkan bucket tujuan umum yang besar, sebaiknya gunakan aturan konfigurasi Siklus Hidup S3. Kedaluwarsa siklus hidup merupakan proses asinkron, sehingga aturannya mungkin memerlukan waktu beberapa hari untuk dijalankan sebelum bucket tersebut kosong. Setelah pertama kali Amazon S3 menjalankan aturan, semua objek yang memenuhi syarat untuk kedaluwarsa ditandai untuk dihapus. Anda tidak lagi dikenakan biaya untuk objek yang ditandai untuk dihapus. Untuk informasi selengkapnya, lihat Bagaimana saya mengosongkan bucket Amazon S3 menggunakan aturan konfigurasi siklus hidup?.

Jika Anda menggunakan konfigurasi siklus hidup untuk mengosongkan bucket, konfigurasi harus mencakup versi terkini, versi terdahulu, penanda hapus, dan unggahan multibagian yang tidak lengkap.

Anda dapat menambahkan aturan konfigurasi siklus hidup untuk menjadikan semua objek atau subset objek yang memiliki prefiks nama kunci tertentu menjadi kedaluwarsa. Misalnya, untuk menghapus semua objek dalam bucket, Anda dapat menyetel aturan siklus hidup untuk kedaluwarsa objek satu hari setelah pembuatannya.

Amazon S3 mendukung aturan siklus hidup bucket yang dapat Anda gunakan untuk menghentikan unggahan multibagian yang tidak selesai dalam jumlah hari tertentu setelah dimulai. Kami menyarankan Anda untuk mengonfigurasi aturan siklus hidup ini, guna meminimalkan biaya penyimpanan Anda. Untuk informasi selengkapnya, lihat Mengonfigurasi siklus hidup bucket untuk menghapus unggahan multibagian yang tidak lengkap.

Untuk informasi lebih lanjut tentang penggunaan konfigurasi siklus hidup untuk mengosongkan bucket, lihat Menyetel konfigurasi Siklus Hidup S3 pada bucket dan Mengakhiri objek.

Mengosongkan bucket tujuan umum dengan dikonfigurasi AWS CloudTrail

AWS CloudTrail melacak peristiwa data tingkat objek dalam bucket tujuan umum Amazon S3, seperti menghapus objek. Jika Anda menggunakan bucket tujuan umum sebagai tujuan untuk mencatat

CloudTrail peristiwa dan menghapus objek dari bucket yang sama, Anda mungkin membuat objek baru sambil mengosongkan bucket. Untuk mencegah hal ini, hentikan AWS CloudTrail jejak Anda. Untuk informasi selengkapnya tentang menghentikan CloudTrail jejak Anda dari peristiwa pencatatan, lihat Menonaktifkan pencatatan untuk jejak di Panduan AWS CloudTrail Pengguna.

Alternatif lain untuk menghentikan CloudTrail jejak agar tidak ditambahkan ke bucket adalah dengan menambahkan `s3:PutObject` pernyataan penolakan ke kebijakan bucket Anda. Jika Anda ingin menyimpan objek baru di bucket di lain waktu Anda harus menghapus pernyataan penolakan `s3:PutObject` ini. Untuk informasi selengkapnya, lihat Operasi objek dan Elemen kebijakan JSON IAM: Efek dalam Panduan Pengguna IAM.

Menghapus bucket tujuan umum

Anda dapat menghapus bucket tujuan umum Amazon S3 kosong. Untuk informasi tentang mengosongkan ember tujuan umum, lihat. the section called “Mengosongkan ember tujuan umum”

Anda dapat menghapus bucket dengan menggunakan konsol Amazon S3, AWS Command Line Interface (AWS CLI), the AWS SDKs, atau Amazon S3 REST API.

> **Important**
>
> Sebelum menghapus bucket tujuan umum, pertimbangkan hal berikut:
>
> - nama bucket tujuan umum unik dalam namespace global. Jika Anda menghapus bucket, ketahuilah bahwa bucket lain Akun AWS dapat menggunakan nama bucket tujuan umum yang sama untuk bucket baru dan karenanya berpotensi menerima permintaan yang ditujukan untuk bucket yang dihapus. Jika Anda ingin mencegah hal ini, atau jika Anda ingin terus menggunakan nama bucket yang sama, jangan hapus bucket. Kami menyarankan Anda mengosongkan ember dan menyimpannya, dan sebagai gantinya, memblokir permintaan bucket apa pun sesuai kebutuhan. Untuk ember yang tidak lagi digunakan secara aktif, kami sarankan untuk mengosongkan ember dari semua objek untuk meminimalkan biaya sambil mempertahankan ember itu sendiri.
> - Saat Anda menghapus bucket tujuan umum, bucket mungkin tidak langsung dihapus. Sebagai gantinya, Amazon S3 mengantri ember untuk dihapus. Karena Amazon S3 didistribusikan Wilayah AWS, proses penghapusan membutuhkan waktu untuk menyebarkan sepenuhnya dan mencapai konsistensi di seluruh sistem.
> - Jika bucket menghosting situs web statis, dan Anda membuat serta mengonfigurasi zona Amazon Route 53 yang di-hosting seperti yang dijelaskan dalam Tutorial: Mengonfigurasi

situs web statis menggunakan domain kustom yang terdaftar di Route 53, Anda harus membersihkan pengaturan zona Route 53 yang di-hosting yang berkaitan dengan bucket tersebut. Untuk informasi selengkapnya, lihat Langkah 2: Menghapus Route 53 zona yang di-hosting.

- Jika bucket menerima data log dari Elastic Load Balancing (ELB), sebaiknya hentikan pengiriman log ELB ke bucket sebelum menghapusnya. Setelah Anda menghapus bucket, jika pengguna lain membuat bucket menggunakan nama yang sama, data log Anda berpotensi dikirim ke bucket tersebut. Untuk informasi tentang log akses ELB, lihat Log akses untuk Classic Load Balancer Anda di Panduan Pengguna untuk Classic Load Balancers dan Access log untuk Application Load Balancer Anda di Panduan Pengguna untuk Application Load Balancers.

Pemecahan Masalah

Jika Anda tidak dapat menghapus bucket tujuan umum Amazon S3, pertimbangkan hal berikut:

- Pastikan ember kosong — Anda dapat menghapus ember hanya jika mereka tidak memiliki objek di dalamnya. Pastikan ember kosong. Untuk informasi tentang mengosongkan ember, lihat. the section called “Mengosongkan ember tujuan umum”
- Pastikan tidak ada titik akses yang terlampir — Anda dapat menghapus bucket hanya jika mereka tidak memiliki Titik Akses S3 atau Titik Akses Multi-Wilayah yang dilampirkan dalam akun yang sama. Sebelum menghapus bucket, hapus titik akses akun yang sama yang dilampirkan ke bucket.
- Pastikan Anda memiliki **`s3:DeleteBucket`** izin — Jika Anda tidak dapat menghapus bucket, bekerjalah dengan administrator IAM Anda untuk mengonfirmasi bahwa Anda memiliki `s3:DeleteBucket` izin. Untuk informasi tentang cara melihat atau memperbarui izin IAM, lihat Mengubah izin untuk pengguna IAM di Panduan Pengguna IAM. Untuk informasi pemecahan masalah, lihat the section called “Memecahkan masalah kesalahan akses ditolak (403 Forbidden)”.
- Periksa **`s3:DeleteBucket Deny`** pernyataan dalam kebijakan kontrol AWS Organizations layanan (SCPs) dan kebijakan kontrol sumber daya (RCPs) — SCPs dan RCPs dapat menolak izin hapus pada bucket. Untuk informasi selengkapnya, lihat kebijakan kontrol layanan dan kebijakan kontrol sumber daya di Panduan AWS Organizations Pengguna.
- Periksa **`s3:DeleteBucket Deny`** pernyataan dalam kebijakan bucket Anda — Jika Anda memiliki `s3:DeleteBucket` izin di pengguna IAM atau kebijakan peran dan Anda tidak dapat menghapus bucket, kebijakan bucket mungkin menyertakan `Deny` pernyataan untuk. `s3:DeleteBucket` Bucket yang dibuat oleh AWS Elastic Beanstalk memiliki kebijakan yang berisi

pernyataan ini secara default. Sebelum Anda dapat menghapus bucket, Anda harus menghapus pernyataan ini, atau kebijakan bucket tersebut.

Prasyarat

Sebelum Anda dapat menghapus ember tujuan umum, Anda harus mengosongkannya. Untuk informasi tentang mengosongkan ember, lihat. the section called “Mengosongkan ember tujuan umum”

Menggunakan konsol S3

Untuk menghapus S3 bucket

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Di daftar bucket, pilih tombol opsi di sebelah nama bucket yang ingin Anda hapus, lalu pilih Hapus di bagian atas halaman.
4. Di halaman Hapus bucket, konfirmasikan bahwa Anda ingin menghapus bucket dengan memasukkan nama bucket ke dalam bidang teks, lalu pilih Hapus bucket.

   > **Note**
   >
   > Jika bucket berisi objek apa pun, kosongkan bucket sebelum menghapusnya dengan memilih tombol Empty bucket di peringatan kesalahan This bucket is not empty dan ikuti petunjuk di halaman Ember kosong. Lalu kembali ke halaman Hapus bucket dan hapus bucket tersebut.

5. Untuk memverifikasi bahwa Anda telah menghapus bucket, buka daftar bucket tujuan umum dan masukkan nama bucket yang Anda hapus. Jika bucket tidak dapat ditemukan, penghapusan Anda berhasil.

Menggunakan AWS SDK for Java

Contoh berikut menunjukkan cara mengosongkan dan menghapus bucket tujuan umum dengan menggunakan AWS SDK untuk Java. Kode pertama-tama menghapus semua objek di ember tujuan umum, dan kemudian menghapus ember.

Untuk contoh dalam bahasa lain, lihat Menggunakan DeleteBucket dengan AWS SDK atau CLI di Referensi API Amazon Simple Storage Service. Untuk informasi tentang penggunaan lainnya AWS SDKs, lihat Alat untuk Amazon Web Services.

Java

JavaContoh berikut menghapus bucket yang berisi objek. Contoh ini menghapus semua objek, lalu menghapus bucket. Contoh ini berfungsi untuk bucket dengan atau tanpa mengaktifkan Penentuan Versi.

> **Note**
>
> Untuk bucket tanpa Penentuan Versi diaktifkan, Anda dapat menghapus semua objek secara langsung dan kemudian menghapus bucket. Untuk bucket dengan Penentuan Versi diaktifkan, Anda harus menghapus semua versi objek sebelum menghapus bucket.

Untuk petunjuk cara membuat dan menguji sampel yang berfungsi, lihat Panduan AWS SDK for Java 2.x Pengembang.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.util.Iterator;

public class DeleteBucket2 {

    public static void main(String[] args) {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();
```

```
            // Delete all objects from the bucket. This is sufficient
            // for unversioned buckets. For versioned buckets, when you attempt to
delete
            // objects, Amazon S3 inserts
            // delete markers for all objects, but doesn't delete the object
versions.
            // To delete objects from versioned buckets, delete all of the object
versions
            // before deleting
            // the bucket (see below for an example).
            ObjectListing objectListing = s3Client.listObjects(bucketName);
            while (true) {
                Iterator<S3ObjectSummary> objIter =
objectListing.getObjectSummaries().iterator();
                while (objIter.hasNext()) {
                    s3Client.deleteObject(bucketName, objIter.next().getKey());
                }

                // If the bucket contains many objects, the listObjects() call
                // might not return all of the objects in the first listing. Check
to
                // see whether the listing was truncated. If so, retrieve the next
page of
                // objects
                // and delete them.
                if (objectListing.isTruncated()) {
                    objectListing = s3Client.listNextBatchOfObjects(objectListing);
                } else {
                    break;
                }
            }

            // Delete all object versions (required for versioned buckets).
            VersionListing versionList = s3Client.listVersions(new
ListVersionsRequest().withBucketName(bucketName));
            while (true) {
                Iterator<S3VersionSummary> versionIter =
versionList.getVersionSummaries().iterator();
                while (versionIter.hasNext()) {
                    S3VersionSummary vs = versionIter.next();
                    s3Client.deleteVersion(bucketName, vs.getKey(),
vs.getVersionId());
                }
```

```
                if (versionList.isTruncated()) {
                    versionList = s3Client.listNextBatchOfVersions(versionList);
                } else {
                    break;
                }
            }

            // After all objects and object versions are deleted, delete the bucket.
            s3Client.deleteBucket(bucketName);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
 couldn't
            // parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Menggunakan AWS CLI

Anda dapat menghapus bucket tujuan umum yang berisi objek AWS CLI jika bucket tidak mengaktifkan versi. Saat Anda menghapus bucket yang berisi objek, semua objek dalam bucket akan dihapus secara permanen, termasuk objek yang telah dialihkan ke kelas penyimpanan S3 Glacier Flexible Retrieval.

Jika bucket Anda tidak mengaktifkan versi, Anda dapat menggunakan AWS CLI perintah `rb` (hapus bucket) dengan `--force` parameter untuk menghapus bucket dan semua objek di dalamnya. Perintah ini menghapus semua objek terlebih dahulu dan kemudian menghapus ember.

Jika pembuatan versi diaktifkan, menggunakan `rb` perintah dengan `--force` parameter tidak akan menghapus objek berversi, sehingga penghapusan bucket gagal karena bucket tidak kosong. Untuk informasi selengkapnya tentang menghapus objek berversi, lihat Menghapus versi objek.

Untuk menggunakan perintah berikut, ganti *`amzn-s3-demo-bucket`* dengan nama bucket yang ingin Anda hapus:

```
$ aws s3 rb s3://amzn-s3-demo-bucket --force
```

Untuk informasi selengkapnya, lihat Menggunakan Perintah S3 Tingkat Tinggi dengan AWS Command Line Interface Panduan Pengguna. AWS Command Line Interface

Pasang bucket Amazon S3 sebagai sistem file lokal

Mountpoint untuk Amazon S3 adalah klien file open source throughput tinggi untuk memasang bucket Amazon S3 sebagai sistem file lokal. Dengan Mountpoint, aplikasi Anda dapat mengakses objek yang disimpan di Amazon S3 melalui operasi sistem file, seperti membuka dan membaca. Mountpoint secara otomatis menerjemahkan operasi ini ke dalam panggilan API objek S3, memberikan aplikasi Anda akses ke penyimpanan elastis serta throughput Amazon S3 melalui antarmuka file.

Mountpoint untuk Amazon S3 tersedia untuk penggunaan produksi pada aplikasi baca-berat skala besar Anda: data lake, pelatihan pembelajaran mesin, rendering gambar, simulasi kendaraan otonom, ekstrak, transformasi, dan muat (ETL), dan banyak lagi.

Mountpoint mendukung operasi sistem file dasar, dan dapat membaca file berukuran hingga 5 TB. Mountpoint dapat mendaftarkan dan membaca file yang ada, dan dapat membuat file yang baru. Mountpoint tidak dapat memodifikasi file yang ada atau menghapus direktori, dan tidak mendukung tautan simbolis atau penguncian file. Mountpoint ideal untuk aplikasi yang tidak memerlukan semua fitur sistem file bersama dan izin bergaya POSIX, namun memerlukan throughput elastis Amazon S3 untuk membaca dan menulis set data S3 yang besar. Untuk detailnya, lihat Perilaku sistem file Mountpoint pada GitHub. Untuk beban kerja yang memerlukan dukungan POSIX penuh, kami merekomendasikan Amazon FSx untuk Lustre dan dukungannya untuk menautkan bucket Amazon S3.

Mountpoint untuk Amazon S3 hanya tersedia untuk sistem operasi Linux. Anda dapat menggunakan Mountpoint untuk mengakses Objek S3 di semua kelas penyimpanan kecuali S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive, S3 Intelligent-Tiering Archive Access Tier, dan S3 Intelligent-Tiering Deep Archive Access Tier.

Topik

- Instalasi Mountpoint
- Mengkonfigurasi dan menggunakan Mountpoint
- Pemecahan Masalah Mountpoint

Instalasi Mountpoint

Anda dapat mengunduh dan menginstal paket Mountpoint untuk Amazon S3 dengan menggunakan baris perintah. Petunjuk untuk mengunduh dan menginstal Mountpoint bervariasi, tergantung pada sistem operasi Linux yang Anda gunakan.

Topik

- Distribusi berbasis RPM (Amazon Linux, Fedora, CentOS, RHEL)
- Distribusi berbasis DEB (Debian, Ubuntu)
- Distribusi Linux lainnya
- Memverifikasi tanda tangan paket Mountpoint untuk Amazon S3

Distribusi berbasis RPM (Amazon Linux, Fedora, CentOS, RHEL)

1. Salin URL unduhan untuk arsitektur Anda.

   x86_64:

   ```
   https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.rpm
   ```

   ARM64 (Graviton):

   ```
   https://s3.amazonaws.com/mountpoint-s3-release/latest/arm64/mount-s3.rpm
   ```

2. Unduh paket Mountpoint untuk Amazon S3. Ganti *download-link* dengan URL unduhan yang sesuai dari langkah sebelumnya.

   ```
   wget download-link
   ```

3. (Opsional) Verifikasi keaslian dan integritas file yang diunduh. Pertama, salin URL tanda tangan untuk arsitektur Anda.

   x86_64:

   ```
   https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.rpm.asc
   ```

   ARM64 (Graviton):

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/arm64/mount-s3.rpm.asc
```

Selanjutnya, lihat Memverifikasi tanda tangan paket Mountpoint untuk Amazon S3.

4. Instal paket menggunakan perintah berikut ini:

```
sudo yum install ./mount-s3.rpm
```

5. Verifikasi bahwa Mountpoint berhasil diinstal dengan memasukkan perintah berikut:

```
mount-s3 --version
```

Anda akan melihat output yang serupa dengan yang berikut:

```
mount-s3 1.3.1
```

Distribusi berbasis DEB (Debian, Ubuntu)

1. Salin URL unduhan untuk arsitektur Anda.

x86_64:

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.deb
```

ARM64 (Graviton):

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/arm64/mount-s3.deb
```

2. Unduh paket Mountpoint untuk Amazon S3. Ganti *download-link* dengan URL unduhan yang sesuai dari langkah sebelumnya.

```
wget download-link
```

3. (Opsional) Verifikasi keaslian dan integritas file yang diunduh. Pertama, salin URL tanda tangan untuk arsitektur Anda.

x86_64:

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.deb.asc
```

ARM64 (Graviton):

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/arm64/mount-s3.deb.asc
```

Selanjutnya, lihat Memverifikasi tanda tangan paket Mountpoint untuk Amazon S3.

4. Instal paket menggunakan perintah berikut ini:

```
sudo apt-get install ./mount-s3.deb
```

5. Konfirmasikan apakah Mountpoint untuk Amazon S3 berhasil diinstal dengan menjalankan perintah berikut ini:

```
mount-s3 --version
```

Anda akan melihat output yang serupa dengan yang berikut:

```
mount-s3 1.3.1
```

Distribusi Linux lainnya

1. Konsultasikan dokumentasi sistem operasi Anda untuk menginstal paket FUSE dan `libfuse2` yang diperlukan.
2. Salin URL unduhan untuk arsitektur Anda.

   x86_64:

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.tar.gz
```

   ARM64 (Graviton):

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/arm64/mount-s3.tar.gz
```

3. Unduh paket Mountpoint untuk Amazon S3. Ganti *`download-link`* dengan URL unduhan yang sesuai dari langkah sebelumnya.

```
wget download-link
```

4. (Opsional) Verifikasi keaslian dan integritas file yang diunduh. Pertama, salin URL tanda tangan untuk arsitektur Anda.

   x86_64:

   ```
   https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.tar.gz.asc
   ```

   ARM64 (Graviton):

   ```
   https://s3.amazonaws.com/mountpoint-s3-release/latest/arm64/mount-s3.tar.gz.asc
   ```

   Selanjutnya, lihat Memverifikasi tanda tangan paket Mountpoint untuk Amazon S3.

5. Instal paket menggunakan perintah berikut ini:

   ```
   sudo mkdir -p /opt/aws/mountpoint-s3 && sudo tar -C /opt/aws/mountpoint-s3 -xzf ./
   mount-s3.tar.gz
   ```

6. Tambahkan biner `mount-s3` ke variabel lingkungan `PATH` Anda. Dalam file `$HOME/.profile` Anda, tambahkan baris berikut:

   ```
   export PATH=$PATH:/opt/aws/mountpoint-s3/bin
   ```

   Simpan file `.profile`, dan jalankan perintah berikut ini:

   ```
   source $HOME/.profile
   ```

7. Konfirmasikan apakah Mountpoint untuk Amazon S3 berhasil diinstal dengan menjalankan perintah berikut ini:

   ```
   mount-s3 --version
   ```

   Anda akan melihat output yang serupa dengan yang berikut:

   ```
   mount-s3 1.3.1
   ```

Memverifikasi tanda tangan paket Mountpoint untuk Amazon S3

1. Instal GnuPG (perintah gpg). Diperlukan untuk memverifikasi keaslian dan integritas paket Mountpoint untuk Amazon S3 yang diunduh. GnuPGdiinstal secara default di Amazon Linux Amazon Machine Images (AMIs). Setelah Anda menginstal GnuPG, lanjutkan ke langkah 2.
2. Unduh kunci publik Mountpoint dengan menjalankan perintah berikut ini:

   ```
   wget https://s3.amazonaws.com/mountpoint-s3-release/public_keys/KEYS
   ```

3. Impor kunci publik Mountpoint ke dalam keyring Anda dengan menjalankan perintah berikut:

   ```
   gpg --import KEYS
   ```

4. Verifikasi sidik jari kunci publik Mountpoint dengan menjalankan perintah berikut:

   ```
   gpg --fingerprint mountpoint-s3@amazon.com
   ```

   Konfirmasikan bahwa string sidik jari yang ditampilkan cocok dengan yang berikut:

   ```
   673F E406 1506 BB46 9A0E  F857 BE39 7A52 B086 DA5A
   ```

   Jika string sidik jarinya tidak cocok, hentikan penginstalan Mountpoint, dan hubungi AWS Dukungan.
5. Unduh file tanda tangan paket. Ganti *signature-link* dengan tautan tanda tangan yang sesuai dari bagian sebelumnya.

   ```
   wget signature-link
   ```

6. Verifikasi tanda tangan paket yang diunduh dengan menjalankan perintah berikut. Ganti *signature-filename* dengan nama file dari langkah sebelumnya.

   ```
   gpg --verify signature-filename
   ```

   Misalnya, pada distribusi berbasis RPM, termasuk Amazon Linux, masukkan perintah berikut:

   ```
   gpg --verify mount-s3.rpm.asc
   ```

7. Outputnya harus menyertakan frasa `Good signature`. Jika output menyertakan frasa `BAD signature`, unduh ulang file paket Mountpoint dan ulangi langkah-langkah ini. Jika masalahnya berlanjut, hentikan penginstalan Mountpoint, dan hubungi AWS Dukungan.

   Outputnya mungkin termasuk peringatan tentang tanda tangan tepercaya. Ini tidak menunjukkan masalah. Ini hanya berarti bahwa Anda belum memverifikasi kunci publik Mountpoint secara independen.

Mengkonfigurasi dan menggunakan Mountpoint

Untuk menggunakan Mountpoint untuk Amazon S3, host Anda memerlukan kredensi yang AWS valid dengan akses ke bucket atau bucket Amazon S3 yang ingin Anda pasang. Untuk berbagai macam cara mengautentikasi, lihat Kredensial AWS Mountpoint pada GitHub.

Misalnya, Anda dapat membuat pengguna dan peran baru AWS Identity and Access Management (IAM) untuk tujuan ini. Pastikan bahwa peran ini memiliki akses ke bucket, atau bucket-bucket yang ingin Anda pasang. Anda dapat meneruskan peran IAM ke EC2 instans Amazon Anda dengan profil instans.

Topik

- Menggunakan Mountpoint untuk Amazon S3
- Mengonfigurasi caching di Mountpoint

Menggunakan Mountpoint untuk Amazon S3

Gunakan Mountpoint untuk Amazon S3 untuk melakukan hal berikut ini:

1. Pasang ember Amazon S3 Anda.

   a. Anda dapat memasang bucket Amazon S3 secara manual dengan menggunakan perintah. `mount-s3`

      Dalam contoh berikut, ganti *`amzn-s3-demo-bucket`* dengan nama bucket S3 Anda, dan ganti *`~/mnt`* dengan direktori di host tempat Anda ingin bucket S3 Anda dipasang.

      ```
      mkdir ~/mnt
      mount-s3 amzn-s3-demo-bucket ~/mnt
      ```

Karena klien Mountpoint berjalan di latar belakang secara default, *~/mnt* direktori sekarang memberi Anda akses ke objek di bucket Amazon S3 Anda.

b. Atau, Anda dapat mengonfigurasi pemasangan otomatis bucket Amazon S3 saat instance dinyalakan atau di-boot ulang.

Untuk EC2 instans Amazon yang ada atau yang sedang berjalan, cari `fstab` file di `/etc/fstab` direktori sistem Linux Anda. Kemudian, tambahkan baris ke `fstab` file Anda. Misalnya, untuk me-mount *amzn-s3-demo-bucket* menggunakan awalan `example-prefix/` ke jalur sytem Anda`/mnt/mountpoint`, lihat yang berikut ini. Untuk menggunakan contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri.

```
s3://amzn-s3-demo-bucket/example-prefix/ /mnt/mountpoint mount-s3
 _netdev,nosuid,nodev,nofail,rw 0 0
```

Lihat tabel berikut untuk penjelasan tentang opsi yang digunakan dalam contoh.

Opsi	Deskripsi
`_netdev`	Menentukan bahwa filesystem membutuhkan jaringan untuk mount.
`nosuid`	Menentukan bahwa filesystem tidak dapat berisi file ID pengguna set.
`nodev`	Menentukan bahwa filesystem tidak dapat berisi perangkat khusus.
`nofail`	Menentukan bahwa kegagalan untuk me-mount filesystem harus tetap memungkinkan sistem untuk boot.
`rw`	Menentukan bahwa titik mount dibuat dengan izin baca dan tulis. Atau, gunakan hanya `ro` untuk membaca.

Untuk EC2 instans Amazon baru, Anda dapat memodifikasi data pengguna pada EC2 template Amazon dan mengatur `fstab` file sebagai berikut. Untuk menggunakan contoh berikut, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
#!/bin/bash -e
MP_RPM=$(mktemp --suffix=.rpm)
curl https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.rpm
 > $MP_RPM
yum install -y $MP_RPM
rm $MP_RPM

MNT_PATH=/mnt/mountpoint
echo "s3://amzn-s3-demo-bucket/ ${MNT_PATH} mount-s3
 _netdev,nosuid,nodev,rw,allow-other,nofail" >> /etc/fstab
mkdir $MNT_PATH

systemctl daemon-reload
mount -a
```

2. Akses objek di bucket Amazon S3 Anda melalui Mountpoint.

   Setelah memasang bucket secara lokal, Anda dapat menggunakan perintah umum Linux, seperti `cat` atau `ls`, untuk bekerja dengan Objek S3. Mountpoint untuk Amazon S3 menafsirkan kunci di bucket Amazon S3 Anda sebagai jalur sistem file dengan memisahkannya pada karakter garis miring () maju. / Misalnya, jika Anda memiliki kunci objek `Data/2023-01-01.csv` di dalam bucket Anda, Anda akan memiliki direktori bernama `Data` dalam sistem file Mountpoint Anda, dengan file bernama `2023-01-01.csv` di dalamnya.

   Mountpoint untuk Amazon S3 secara sengaja tidak menerapkan spesifikasi standar POSIX lengkap untuk sistem file. Mountpoint dioptimalkan untuk beban kerja yang memerlukan akses baca dan tulis throughput tinggi ke data yang disimpan di Amazon S3 melalui antarmuka sistem file, namun tidak bergantung pada fitur sistem file. Untuk informasi selengkapnya, lihat perilaku sistem file Mountpoint untuk Amazon S3 pada GitHub. Pelanggan yang membutuhkan semantik sistem file yang lebih kaya harus mempertimbangkan layanan AWS file lainnya, seperti Amazon Elastic File System (Amazon EFS) atau Amazon. FSx

3. Lepaskan bucket Amazon S3 Anda dengan menggunakan `umount` perintah. Perintah ini melepaskan bucket S3 Anda, dan keluar dari Mountpoint.

Untuk menggunakan perintah contoh berikut ini, ganti *~/mnt* dengan direktori di host tempat bucket S3 Anda dipasang.

```
umount ~/mnt
```

> **Note**
>
> Untuk mendapatkan daftar opsi untuk perintah ini, jalankan `umount --help`.

Untuk detail konfigurasi Mountpoint tambahan, lihat konfigurasi bucket Amazon S3, dan konfigurasi sistem file aktif. GitHub

Mengonfigurasi caching di Mountpoint

Mountpoint untuk Amazon S3 mendukung berbagai jenis caching data. Untuk mempercepat permintaan baca berulang, Anda dapat memilih untuk mengikuti yang berikut:

- Cache lokal — Anda dapat menggunakan cache lokal di penyimpanan EC2 instans Amazon atau volume Amazon Elastic Block Store. Jika Anda berulang kali membaca data yang sama dari instance komputasi yang sama dan jika Anda memiliki ruang yang tidak digunakan di penyimpanan instans lokal untuk kumpulan data yang berulang kali dibaca, Anda harus memilih untuk masuk ke cache lokal.
- Cache bersama - Anda dapat menggunakan cache bersama di S3 Express One Zone. Jika Anda berulang kali membaca objek kecil dari beberapa instance komputasi atau jika Anda tidak tahu ukuran kumpulan data yang Anda baca berulang kali dan ingin mendapatkan keuntungan dari elastisitas ukuran cache, Anda harus ikut serta dalam cache bersama. Setelah Anda ikut serta, Mountpoint mempertahankan objek dengan ukuran hingga satu megabyte dalam bucket direktori yang menggunakan S3 Express One Zone.
- Gabungan cache lokal dan bersama - Jika Anda memiliki ruang yang tidak digunakan di cache lokal Anda tetapi juga menginginkan cache bersama di beberapa instance, Anda dapat memilih untuk menggunakan cache lokal dan cache bersama.

Caching di Mountpoint sangat ideal untuk kasus penggunaan di mana Anda berulang kali membaca data yang sama yang tidak berubah selama beberapa kali pembacaan. Misalnya, Anda dapat

menggunakan caching dengan tugas pelatihan machine learning yang perlu membaca set data pelatihan beberapa kali untuk meningkatkan akurasi model.

Untuk informasi selengkapnya tentang cara mengkonfigurasi caching di Mountpoint, lihat contoh berikut.

Topik

- Cache lokal
- Cache bersama
- Gabungan cache lokal dan bersama

Cache lokal

Anda dapat memilih untuk masuk ke cache lokal dengan `--cache CACHE_PATH` bendera. Dalam contoh berikut, ganti *CACHE_PATH* dengan filepath ke direktori tempat Anda ingin menyimpan data Anda. Ganti *amzn-s3-demo-bucket* dengan nama bucket Amazon S3 Anda, dan ganti *~/mnt* dengan direktori di host tempat Anda ingin bucket S3 Anda dipasang.

```
mkdir ~/mnt
mount-s3 --cache CACHE_PATH amzn-s3-demo-bucket ~/mnt
```

Saat Anda memilih caching lokal saat memasang bucket Amazon S3, Mountpoint membuat sub-direktori kosong di lokasi cache yang dikonfigurasi, jika sub-direktori tersebut belum ada. Saat pertama kali memasang bucket dan saat Anda melepas pemasangan, Mountpoint menghapus konten cache lokal.

> **Important**
>
> Jika Anda mengaktifkan caching lokal, Mountpoint akan mempertahankan konten objek yang tidak terenkripsi dari bucket Amazon S3 yang dipasang di lokasi cache lokal yang disediakan saat pemasangan. Untuk melindungi data Anda, Anda harus membatasi akses ke lokasi cache data dengan menggunakan mekanisme kontrol akses sistem file.

Cache bersama

Jika Anda berulang kali membaca objek kecil (hingga 1 MB) dari beberapa instance komputasi atau ukuran kumpulan data yang berulang kali Anda baca sering melebihi ukuran cache lokal Anda, Anda

harus menggunakan cache bersama di S3 Express One Zone. Saat Anda membaca data yang sama berulang kali dari beberapa instans, ini meningkatkan latensi dengan menghindari permintaan berlebihan ke bucket Amazon S3 yang dipasang.

Setelah Anda memilih cache bersama, Anda membayar data yang di-cache di bucket direktori Anda di S3 Express One Zone. Anda juga membayar permintaan yang dibuat terhadap data Anda di bucket direktori di S3 Express One Zone. Untuk informasi selengkapnya, lihat Harga Amazon S3. Mountpoint tidak pernah menghapus objek yang di-cache dari bucket direktori. Untuk mengelola biaya penyimpanan, Anda harus menyiapkan kebijakan Siklus Hidup di bucket direktori agar Amazon S3 kedaluwarsa data cache di S3 Express One Zone setelah jangka waktu yang ditentukan. Untuk informasi selengkapnya, lihat Mountpoint untuk konfigurasi caching Amazon S3 aktif. GitHub

Untuk ikut serta dalam cache di S3 Express One Zone saat Anda memasang bucket Amazon S3 ke instance komputasi, gunakan `--cache-xz` flag dan tentukan bucket direktori sebagai lokasi cache Anda. Dalam contoh berikut, ganti*user input placeholders*.

```
mount-s3 amzn-s3-demo-bucket ~/mnt --cache-xz amzn-s3-demo-bucket--usw2-az1--x-s3
```

Gabungan cache lokal dan bersama

Jika Anda memiliki ruang yang tidak digunakan pada instans Anda tetapi Anda juga ingin menggunakan cache bersama di beberapa instance, Anda dapat memilih untuk menggunakan cache lokal dan cache bersama. Dengan konfigurasi caching ini, Anda dapat menghindari permintaan baca yang berlebihan dari instance yang sama ke cache bersama di bucket direktori saat data yang diperlukan di-cache di penyimpanan lokal. Ini dapat mengurangi biaya permintaan dan meningkatkan kinerja.

Untuk ikut serta dalam cache lokal dan cache bersama saat memasang bucket Amazon S3, Anda menentukan kedua lokasi cache dengan menggunakan `--cache` dan `--cache-xz` flag. Untuk menggunakan contoh berikut untuk memilih cache lokal dan bersama, ganti file*user input placeholders*.

```
mount -s3 amzn-s3-demo-bucket ~/mnt --cache /path/to/mountpoint/cache --cache -xz amzn-s3-demo-bucket--usw2-az1--x-s3
```

Untuk informasi lebih lanjut, konfigurasi caching Mountpoint untuk Amazon S3 aktif. GitHub

> **Important**
>
> Jika Anda mengaktifkan cache bersama, Mountpoint akan menyalin konten objek dari bucket Amazon S3 yang terpasang ke bucket direktori S3 yang Anda berikan sebagai lokasi cache bersama, sehingga dapat diakses oleh pemanggil mana pun yang memiliki akses ke bucket direktori S3. Untuk melindungi data cache Anda, Anda harus mengikuti Praktik Terbaik Keamanan untuk Amazon S3 untuk memastikan bahwa bucket Anda menggunakan kebijakan yang benar dan tidak dapat diakses publik. Anda harus menggunakan bucket direktori yang didedikasikan untuk cache bersama Mountpoint dan memberikan akses hanya ke klien Mountpoint.

Pemecahan Masalah Mountpoint

Mountpoint untuk Amazon S3 didukung oleh. Dukungan Jika Anda membutuhkan bantuan, hubungi AWS Dukungan Pusat.

Anda juga dapat meninjau dan mengirimkan Masalah Mountpoint di GitHub.

Jika Anda menemukan potensi masalah keamanan dalam proyek ini, kami meminta Anda untuk memberi tahu Keamanan AWS melalui halaman pelaporan kerentanan kami. Jangan membuat pelaporannya di masalah GitHub publik.

Jika aplikasi Anda berperilaku tidak terduga dengan Mountpoint, Anda dapat memeriksa informasi log untuk mendiagnosis masalahnya.

Pencatatan log

Secara default, Mountpoint mengeluarkan informasi log tingkat keparahan tinggi ke syslog.

Untuk melihat log pada sebagian besar distribusi Linux modern, termasuk Amazon Linux, jalankan perintah `journald` berikut:

```
journalctl -e SYSLOG_IDENTIFIER=mount-s3
```

Pada sistem Linux lain, entri `syslog` kemungkinan akan ditulis ke file seperti `/var/log/syslog`.

Anda dapat menggunakan log ini untuk memecahkan masalah aplikasi Anda. Misalnya, jika aplikasi Anda mencoba menimpa file yang sudah ada, operasi tersebut gagal, dan Anda akan melihat baris yang mirip dengan berikut ini di dalam log:

```
[WARN] open{req=12 ino=2}: mountpoint_s3::fuse: open failed: inode error: inode 2 (full
 key "README.md") is not writable
```

Untuk informasi selengkapnya, lihat Pencatatan log Mountpoint untuk Amazon S3 pada GitHub.

Bekerja dengan Browser Penyimpanan untuk Amazon S3

Browser Penyimpanan untuk S3 adalah komponen open source yang dapat Anda tambahkan ke aplikasi web Anda untuk menyediakan pengguna akhir Anda dengan antarmuka grafis sederhana untuk data yang disimpan di Amazon S3. Dengan Browser Penyimpanan untuk S3, Anda dapat memberikan akses kepada pengguna akhir resmi untuk menelusuri, mengunduh, mengunggah, menyalin, dan menghapus data di S3 langsung dari aplikasi Anda sendiri.

Browser Penyimpanan untuk S3 mendukung operasi berikut untuk file:`LIST`,,`GET`,`PUT`, `COPY`UPLOAD, dan`DELETE`. Untuk memberikan transfer data throughput tinggi, Storage Browser untuk S3 hanya menampilkan data yang diizinkan oleh pengguna akhir Anda untuk mengakses dan mengoptimalkan permintaan unggahan. Browser Penyimpanan juga mengoptimalkan kinerja untuk waktu muat yang lebih cepat, menghitung checksum data yang diunggah pengguna akhir Anda, dan menerima objek setelah mengonfirmasi bahwa integritas data Anda dipertahankan (dalam perjalanan) melalui internet publik. Anda dapat mengontrol akses ke data berdasarkan identitas pengguna akhir menggunakan layanan AWS keamanan dan identitas, atau layanan terkelola Anda sendiri. Anda juga dapat menyesuaikan Storage Browser agar sesuai dengan desain dan branding aplikasi yang ada.

Storage Browser untuk S3 hanya tersedia untuk aplikasi web dan intranet pada framework React. Anda dapat menggunakan Storage Browser untuk mengakses objek Amazon S3 di semua kelas penyimpanan kecuali S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive, S3 Intelligent-Tiering Archive Access tier, dan tingkat S3 Intelligent-Tiering Deep Archive Access.

Browser Penyimpanan untuk S3 tersedia untuk digunakan dengan aplikasi web Anda di pustaka AWS Amplify React. Untuk informasi selengkapnya tentang Browser Penyimpanan, lihat Browser Penyimpanan untuk S3.

Topik

- Menggunakan Browser Penyimpanan untuk S3
- Instalasi Browser Penyimpanan untuk S3
- Menyiapkan Browser Penyimpanan untuk S3
- Mengkonfigurasi Browser Penyimpanan untuk S3

- Pemecahan Masalah Browser Penyimpanan untuk S3

Menggunakan Browser Penyimpanan untuk S3

Di Browser Penyimpanan untuk S3, lokasi adalah bucket atau awalan tujuan umum S3, yang Anda berikan akses kepada pengguna akhir untuk menggunakan Hibah Akses S3, kebijakan IAM, atau layanan otorisasi terkelola Anda sendiri. Setelah Anda mengizinkan pengguna akhir Anda untuk mengakses lokasi tertentu, mereka dapat bekerja dengan data apa pun di lokasi tersebut.

Browser Penyimpanan untuk antarmuka pengguna S3 memiliki empat tampilan utama:

- Halaman rumah: Halaman beranda mencantumkan lokasi S3 yang dapat diakses pengguna Anda, serta izin Anda untuk masing-masing lokasi. Ini adalah tampilan awal untuk pengguna yang menunjukkan sumber daya S3 tingkat root yang dapat diakses oleh pengguna akhir Anda dan izin (READ/WRITE/READWRITE) untuk setiap lokasi S3.
- Detail lokasi: Tampilan ini memungkinkan pengguna untuk menelusuri file dan folder di S3, dan mengunggah atau mengunduh file. (Perhatikan bahwa di Browser Penyimpanan untuk S3, objek dikenal sebagai file, dan awalan dan ember dikenal sebagai folder.)
- Tindakan lokasi: Setelah pengguna memilih tindakan (seperti Unggah) di Browser Penyimpanan, itu membuka tampilan lain dari lokasi file.
- Elipsis vertikal: Ikon elipsis vertikal membuka daftar tindakan drop-down.

Saat menggunakan Browser Penyimpanan untuk S3, perhatikan batasan berikut:

- Folder yang dimulai atau diakhiri dengan titik (.) tidak didukung.
- Hibah Akses S3 dengan izin WRITE saja tidak didukung.
- Browser Penyimpanan untuk S3 mendukung PUT operasi untuk file berukuran hingga 160 GB.
- Browser Penyimpanan untuk S3 hanya mendukung COPY operasi untuk file yang lebih kecil dari 5 GB. Jika ukuran file melebihi 5 GB, Browser Penyimpanan gagal permintaan.

Instalasi Browser Penyimpanan untuk S3

Cara tercepat untuk memulai dengan Storage Browser adalah dengan mengkloning salah satu proyek sampel. GitHub Contoh proyek ini dapat membantu Anda menerapkan aplikasi web siap produksi untuk Browser Penyimpanan dengan integrasi AWS layanan yang telah

ditetapkan sebelumnya AWS Identity and Access Management sehingga Anda dapat dengan cepat menghubungkan pengguna akhir yang berwenang ke data di S3.

Untuk informasi selengkapnya, lihat Mulai cepat di Amplify Dev Center.

Menginstal Browser Penyimpanan untuk S3 dari GitHub

Atau, Anda dapat menginstal Browser Penyimpanan untuk S3 dari versi terbaru `aws-amplify/ui-react-storage` dan `aws-amplify` paket di `aws-amplify` GitHubrepositori untuk mulai mengintegrasikan Browser Penyimpanan ke dalam aplikasi yang ada. Saat menginstal Storage Browser untuk S3, pastikan untuk menambahkan dependensi berikut ke file Anda: `package.json`

```
"dependencies": {
    "aws-amplify/ui-react-storage": "latest",
    "aws-amplify": "latest",
  }
```

Atau, Anda dapat menambahkan dependensi menggunakan Node Package Manager (NPM):

```
npm i --save @aws-amplify/ui-react-storage aws-amplify
```

Menyiapkan Browser Penyimpanan untuk S3

Untuk menghubungkan pengguna akhir dengan lokasi Amazon S3, Anda harus terlebih dahulu menyiapkan metode otentikasi dan otorisasi. Ada tiga metode untuk mengatur metode otentikasi dan otorisasi dengan Storage Browser:

- Metode 1: Mengelola akses data untuk pelanggan Anda dan mitra pihak ketiga
- Metode 2: Mengelola akses data untuk prinsipal IAM Anda untuk akun Anda AWS
- Metode 3: Mengelola akses data dalam skala besar

Metode 1: Mengelola akses data untuk pelanggan Anda dan mitra pihak ketiga

Dengan metode ini, Anda dapat menggunakan AWS Amplify Auth untuk mengelola kontrol akses dan keamanan untuk file. Metode ini sangat ideal ketika Anda ingin menghubungkan pelanggan Anda atau mitra pihak ketiga dengan data di S3. Dengan opsi ini, pelanggan Anda dapat mengautentikasi menggunakan penyedia identitas sosial atau perusahaan.

Anda memberikan kredensil IAM kepada pengguna akhir dan mitra pihak ketiga menggunakan AWS Amplify Auth dengan bucket S3 yang dikonfigurasi untuk menggunakan Amplify Storage. AWS Amplify Auth dibangun di Amazon Cognito, identitas pelanggan yang dikelola sepenuhnya dan layanan manajemen akses tempat Anda dapat mengautentikasi dan mengotorisasi pengguna dari direktori pengguna atau direktori perusahaan bawaan, atau dari penyedia identitas konsumen. Model otorisasi Amplify mendefinisikan awalan mana yang dapat diakses oleh pengguna yang diautentikasi saat ini. Untuk informasi selengkapnya tentang cara mengatur otorisasi AWS Amplify, lihat Menyiapkan penyimpanan.

Untuk menginisialisasi komponen dengan metode otentikasi dan penyimpanan Amplify, tambahkan cuplikan kode berikut ke aplikasi web Anda:

```
import {
  createAmplifyAuthAdapter,
  createStorageBrowser,
} from '@aws-amplify/ui-react-storage/browser';
import "@aws-amplify/ui-react-storage/styles.css";

import config from './amplify_outputs.json';

Amplify.configure(config);

export const { StorageBrowser } = createStorageBrowser({
  config: createAmplifyAuthAdapter(),
});
```

Metode 2: Mengelola akses data untuk prinsipal IAM Anda untuk akun Anda AWS

Jika Anda ingin mengelola akses untuk prinsipal IAM atau Akun AWS langsung, Anda dapat membuat peran IAM yang memiliki izin untuk menjalankan operasi API S3. GetDataAccess Untuk mengaturnya, Anda harus membuat instance S3 Access Grants untuk memetakan izin untuk bucket dan awalan tujuan umum S3 ke identitas IAM yang ditentukan. Komponen Storage Browser (yang harus dipanggil di sisi klien setelah mendapatkan kredensyal IAM) kemudian akan memanggil operasi ListCallerAccessGrantsS3 API untuk mengambil hibah yang tersedia untuk pemohon identitas dan mengisi lokasi dalam komponen. Setelah Anda mendapatkan `s3:GetDataAccess` izin, kredensyal tersebut kemudian digunakan oleh komponen Storage Browser untuk meminta akses data ke S3.

```
import {
  createManagedAuthAdapter,
```

```
  createStorageBrowser,
} from '@aws-amplify/ui-react-storage/browser';
import "@aws-amplify/ui-react-storage/styles.css";

export const { StorageBrowser } = createStorageBrowser({
  config: createManagedAuthAdapter({
    credentialsProvider: async (options?: { forceRefresh?: boolean }) => {
      // return your credentials object
      return {
        credentials: {
          accessKeyId: 'my-access-key-id',
          secretAccessKey: 'my-secret-access-key',
          sessionToken: 'my-session-token',
          expiration: new Date()
        },
      }
    },
    // AWS `region` and `accountId`
    region: '',
    accountId: '',
    // call `onAuthStateChange` when end user auth state changes
    // to clear sensitive data from the `StorageBrowser` state
    registerAuthListener: (onAuthStateChange) => {},
  })
});
```

Metode 3: Mengelola akses data dalam skala besar

Jika Anda ingin mengaitkan instans Hibah Akses S3 ke Pusat Identitas IAM Anda untuk solusi yang lebih skalabel (seperti menyediakan akses data ke seluruh perusahaan Anda), Anda dapat meminta data dari Amazon S3 atas nama pengguna yang diautentikasi saat ini. Misalnya, Anda dapat memberikan grup pengguna di direktori perusahaan Anda akses ke data Anda di S3. Pendekatan ini memungkinkan Anda mengelola izin Hibah Akses S3 secara terpusat untuk pengguna dan grup Anda, termasuk yang dihosting di penyedia eksternal seperti Microsoft Entra, Okta, dan lainnya.

Saat menggunakan metode ini, integrasi dengan Pusat Identitas IAM memungkinkan Anda untuk menggunakan direktori pengguna yang ada. Manfaat lain dari propagasi identitas tepercaya IAM Identity Center adalah bahwa setiap peristiwa AWS CloudTrail data untuk Amazon S3 berisi referensi langsung ke identitas pengguna akhir yang mengakses data S3.

Jika Anda memiliki aplikasi yang mendukung OAuth 2.0 dan pengguna Anda memerlukan akses dari aplikasi ini ke AWS layanan, kami sarankan Anda menggunakan propagasi identitas tepercaya.

Dengan propagasi identitas tepercaya, pengguna dapat masuk ke aplikasi, dan aplikasi itu dapat meneruskan identitas pengguna dalam permintaan apa pun yang mengakses data dalam AWS layanan. Aplikasi ini berinteraksi dengan IAM Identity Center atas nama pengguna yang diautentikasi. Untuk informasi selengkapnya, lihat Menggunakan propagasi identitas tepercaya dengan aplikasi yang dikelola pelanggan.

Propagasi identitas tepercaya adalah AWS IAM Identity Center fitur yang Layanan AWS dapat digunakan oleh administrator terhubung untuk memberikan dan mengaudit akses ke data layanan. Akses ke data ini didasarkan pada atribut pengguna seperti asosiasi grup. Menyiapkan propagasi identitas tepercaya memerlukan kolaborasi antara administrator yang terhubung Layanan AWS dan administrator Pusat Identitas IAM. Untuk informasi lebih lanjut, lihat Prasyarat dan pertimbangan.

Pengaturan

Untuk mengatur otentikasi Browser Penyimpanan dalam propagasi identitas tepercaya AWS Management Console menggunakan S3 Access Grants dan IAM Identity Center, aplikasi Anda harus meminta data dari Amazon S3 atas nama pengguna yang diautentikasi saat ini. Dengan pendekatan ini, Anda dapat memberi pengguna atau grup pengguna dari direktori perusahaan Anda akses langsung ke bucket, awalan, atau objek S3 Anda. Ini berarti bahwa aplikasi Anda tidak perlu memetakan pengguna ke prinsipal IAM.

Alur kerja berikut menguraikan langkah-langkah untuk menyiapkan Browser Penyimpanan untuk S3, menggunakan IAM Identity Center dan S3 Access Grants:

Langkah-langkah	Deskripsi
1	Aktifkan Pusat Identitas IAM untuk Anda AWS Organizations
2	Konfigurasikan federasi Pusat AWS Identity and Access Management Identitas
3	Tambahkan penerbit token tepercaya di konsol Pusat AWS Identity and Access Management Identitas Penerbit token tepercaya mewakili penyedia identitas eksternal (iDP) Anda dalam IAM Identity Center, memungkinkannya mengenali token identitas untuk pengguna yang diautentikasi aplikasi Anda.

Langkah-langkah	Deskripsi
4	Buat peran IAM untuk `bootstrap` aplikasi dan `identity bearer`
5	Buat dan konfigurasikan aplikasi Anda di IAM Identity Center Aplikasi ini berinteraksi dengan IAM Identity Center atas nama pengguna yang diautentikasi.
6	Tambahkan S3 Access Grants sebagai aplikasi tepercaya untuk propagasi identitas Langkah ini menghubungkan aplikasi Anda ke S3 Access Grants, sehingga dapat membuat permintaan ke S3 Access Grants atas nama pengguna yang diautentikasi.
7	Membuat hibah untuk pengguna atau grup Langkah ini menyinkronkan pengguna dari AWS Identity and Access Management Identity Center dengan System for Cross-domain Identity Management (SCIM). SCIM menjaga identitas IAM Identity Center Anda tetap sinkron dengan identitas dari penyedia identitas Anda (iDP).
8	Buat Browser Penyimpanan Anda untuk komponen S3

Aktifkan Pusat Identitas IAM untuk Anda AWS Organizations

Untuk mengaktifkan Pusat Identitas IAM untuk Anda AWS Organizations, lakukan langkah-langkah berikut:

1. Masuk ke AWS Management Console, menggunakan salah satu metode ini:
    1. Baru di AWS (pengguna root) - Masuk sebagai pemilik akun dengan memilih pengguna Root dan memasukkan alamat Akun AWS email Anda. Di laman berikutnya, masukkan kata sandi.
    2. Sudah menggunakan AWS (kredensyal IAM) - Masuk menggunakan kredensyal IAM Anda dengan izin administratif.
2. Buka konsol Pusat Identitas IAM.

3. Di bawah Aktifkan Pusat Identitas IAM, pilih Aktifkan.

> **Note**
>
> IAM Identity Center membutuhkan pengaturan. AWS Organizations Jika Anda belum menyiapkan organisasi, Anda dapat memilih untuk AWS membuatnya untuk Anda. Pilih Buat AWS organisasi untuk menyelesaikan proses ini.

4. Pilih Aktifkan dengan AWS Organizations.
5. Pilih Lanjutkan.
6. (Opsional) Tambahkan tag apa pun yang ingin Anda kaitkan dengan instance organisasi ini.
7. (Opsional) Konfigurasikan administrasi yang didelegasikan.

> **Note**
>
> Jika Anda menggunakan lingkungan multi-akun, kami sarankan Anda mengonfigurasi administrasi yang didelegasikan. Dengan administrasi yang didelegasikan, Anda dapat membatasi jumlah orang yang memerlukan akses ke akun manajemen di AWS Organizations. Untuk informasi selengkapnya, lihat Administrasi yang didelegasikan.

8. Pilih Simpan.

AWS Organizations secara otomatis mengirimkan email verifikasi ke alamat yang terkait dengan akun manajemen Anda. Mungkin ada waktu tunda sebelum Anda menerima email verifikasi. Pastikan untuk memverifikasi alamat email Anda dalam waktu 24 jam, sebelum email verifikasi Anda kedaluwarsa.

Konfigurasikan federasi Pusat AWS Identity and Access Management Identitas

Untuk menggunakan Browser Penyimpanan untuk S3 dengan pengguna direktori perusahaan, Anda harus mengonfigurasi Pusat Identitas IAM untuk menggunakan penyedia identitas eksternal (iDP). Anda dapat menggunakan penyedia identitas pilihan Anda. Namun, ketahuilah bahwa setiap penyedia identitas menggunakan pengaturan konfigurasi yang berbeda. Untuk tutorial tentang menggunakan penyedia identitas eksternal yang berbeda, lihat tutorial sumber Pusat Identitas IAM.

> **Note**
>
> Pastikan untuk merekam URL penerbit dan atribut audiens dari penyedia identitas yang telah Anda konfigurasi karena Anda harus merujuknya di langkah selanjutnya. Jika Anda tidak memiliki akses atau izin yang diperlukan untuk mengonfigurasi iDP, Anda mungkin perlu menghubungi administrator iDP eksternal untuk mendapatkannya.

Tambahkan penerbit token tepercaya di konsol Pusat AWS Identity and Access Management Identitas

Penerbit token tepercaya mewakili penyedia identitas eksternal Anda di Pusat AWS Identity and Access Management Identitas, dan mengenali token untuk pengguna yang diautentikasi aplikasi Anda. Pemilik akun instans Pusat Identitas IAM di Anda AWS Organizations harus melakukan langkah-langkah ini. Langkah-langkah ini dapat dilakukan baik di konsol IAM Identity Center, atau secara terprogram.

Untuk menambahkan penerbit token tepercaya di konsol Pusat AWS Identity and Access Management Identitas, lakukan langkah-langkah berikut:

1. Buka konsol Pusat Identitas IAM.
2. Pilih Pengaturan.
3. Pilih tab Otentikasi.
4. Buka bagian Penerbit token tepercaya, dan isi detail berikut:
    1. Di bawah URL Penerbit, masukkan URL IDP eksternal yang berfungsi sebagai penerbit token tepercaya. Anda mungkin perlu menghubungi administrator iDP eksternal untuk mendapatkan informasi ini. Untuk informasi selengkapnya, lihat Menggunakan aplikasi dengan penerbit token tepercaya.
    2. Di bawah nama penerbit token tepercaya, masukkan nama untuk penerbit token tepercaya. Nama ini akan muncul dalam daftar penerbit token tepercaya yang dapat Anda pilih di Langkah 8, ketika sumber daya aplikasi dikonfigurasi untuk propagasi identitas.
5. Perbarui atribut Peta Anda ke atribut aplikasi pilihan Anda, di mana setiap atribut penyedia identitas dipetakan ke atribut Pusat Identitas IAM. Misalnya, Anda mungkin ingin memetakan atribut aplikasi `email` ke atribut `email` pengguna IAM Identity Center. Untuk melihat daftar atribut pengguna yang diizinkan di Pusat Identitas IAM, lihat tabel di Pemetaan atribut untuk direktori AWS Microsoft AD Terkelola.

6. (Opsional) Jika Anda ingin menambahkan tag sumber daya, masukkan pasangan kunci dan nilai. Untuk menambahkan beberapa tag sumber daya, pilih Tambahkan tag baru untuk menghasilkan entri baru dan masukkan pasangan kunci dan nilai.
7. Pilih Buat penerbit token tepercaya.
8. Setelah Anda selesai membuat penerbit token tepercaya, hubungi administrator aplikasi untuk memberi tahu mereka nama penerbit token tepercaya, sehingga mereka dapat mengonfirmasi bahwa penerbit token tepercaya terlihat di konsol yang berlaku.
9. Pastikan administrator aplikasi memilih penerbit token tepercaya ini di konsol yang berlaku untuk mengaktifkan akses pengguna ke aplikasi dari aplikasi yang dikonfigurasi untuk propagasi identitas tepercaya.

Buat peran IAM untuk **`bootstrap`** aplikasi dan **`identity bearer`**

Untuk memastikan bahwa `bootstrap` aplikasi dan `identity bearer` pengguna dapat bekerja dengan baik satu sama lain, pastikan untuk membuat dua peran IAM. Satu peran IAM diperlukan untuk `bootstrap` aplikasi dan peran IAM lainnya harus digunakan untuk pembawa identitas, atau pengguna akhir yang mengakses aplikasi web yang meminta akses melalui S3 Access Grants. `bootstrap`Aplikasi menerima token yang dikeluarkan oleh penyedia identitas dan memanggil `CreateTokenWithIAM` API, menukar token ini dengan token yang dikeluarkan oleh Pusat Identitas.

Buat peran IAM, sepertibootstrap-role, dengan izin seperti berikut ini. Contoh berikut kebijakan IAM memberikan izin `bootstrap-role` untuk melakukan pertukaran token:

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Action": [
            "sso-oauth:CreateTokenWithIAM",
        ],
        "Resource": "arn:${Partition}:sso::${AccountId}:application/${InstanceId}/
${ApplicationId}",
        "Effect": "Allow"
    },
    {
        "Action": [
            "sts:AssumeRole",
            "sts:SetContext"
        ],
        "Resource": "arn:aws:iam::${AccountId}:role/identity-bearer-role",
```

```
        "Effect": "Allow"
    }]
}
```

Kemudian, buat peran IAM kedua (seperti`identity-bearer-role`), yang digunakan broker identitas untuk menghasilkan kredensi IAM. Kredensi IAM yang dikembalikan dari broker identitas ke aplikasi web digunakan oleh Storage Browser untuk komponen S3 untuk memungkinkan akses ke data S3:

```
{
    "Action": [
        "s3:GetDataAccess",
        "s3:ListCallerAccessGrants",
    ],
    "Resource": "arn:${Partition}:s3:${Region}:${Account}:access-grants/default",
    "Effect": "Allow"
}
```

Peran IAM (`identity-bearer-role`) ini harus menggunakan kebijakan kepercayaan dengan pernyataan berikut:

```
{
   "Effect": "Allow",
   "Principal": {
      "AWS": "arn:${Partition}:iam::${Account}:role/${RoleNameWithPath}"
   },
   "Action": [
       "sts:AssumeRole",
       "sts:SetContext"
   ]
}
```

Buat dan konfigurasikan aplikasi Anda di IAM Identity Center

> **Note**
>
> Sebelum memulai, pastikan bahwa Anda telah membuat peran IAM yang diperlukan dari langkah sebelumnya. Anda harus menentukan salah satu peran IAM ini dalam langkah ini.

Untuk membuat dan mengkonfigurasi aplikasi yang dikelola pelanggan di AWS IAM Identity Center, lakukan langkah-langkah berikut:

1. Buka konsol Pusat Identitas IAM.
2. Pilih Aplikasi.
3. Pilih tab yang dikelola Pelanggan.
4. Pilih Tambahkan aplikasi.
5. Pada halaman Pilih jenis aplikasi, di bawah preferensi Pengaturan, pilih Saya memiliki aplikasi yang ingin saya atur.
6. Di bawah Jenis aplikasi, pilih OAuth2.0.
7. Pilih Berikutnya. Halaman Tentukan aplikasi ditampilkan.
8. Di bawah bagian Nama dan deskripsi aplikasi, masukkan nama Tampilan untuk aplikasi, seperti**`storage-browser-oauth`**.
9. Masukkan Deskripsi. Deskripsi aplikasi muncul di konsol IAM Identity Center dan permintaan API, tetapi tidak di portal AWS akses.
10. Di bawah Metode penugasan pengguna dan grup, pilih Jangan memerlukan tugas. Opsi ini memungkinkan semua pengguna dan grup IAM Identity Center resmi mengakses aplikasi ini.
11. Di bawah portal AWS akses, masukkan URL aplikasi tempat pengguna dapat mengakses aplikasi.
12. (Opsional) Jika Anda ingin menambahkan tag sumber daya, masukkan pasangan kunci dan nilai. Untuk menambahkan beberapa tag sumber daya, pilih Tambahkan tag baru untuk menghasilkan entri baru dan masukkan pasangan kunci dan nilai.
13. Pilih Berikutnya. Halaman Tentukan otentikasi ditampilkan.
14. Di bawah Otentikasi dengan penerbit token tepercaya, gunakan kotak centang untuk memilih penerbit token tepercaya yang sebelumnya Anda buat.
15. Di bawah Konfigurasi penerbit token tepercaya yang dipilih, masukkan klaim aud. Klaim aud mengidentifikasi audiens JSON Web Token (JWT), dan itu adalah nama yang digunakan penerbit token tepercaya untuk mengidentifikasi aplikasi ini.

> **Note**
>
> Anda mungkin perlu menghubungi administrator iDP eksternal untuk mendapatkan informasi ini.

16. Pilih Berikutnya. Halaman Tentukan kredensial otentikasi ditampilkan.
17. Di bawah Metode konfigurasi, pilih Masukkan satu atau beberapa peran IAM.
18. Di bawah Masukkan peran IAM, tambahkan peran IAM atau Nama Sumber Daya Amazon (ARN) untuk token pembawa identitas. Anda harus memasukkan peran IAM yang Anda buat dari langkah sebelumnya untuk aplikasi broker identitas (misalnya,**`bootstrap-role`**).
19. Pilih Berikutnya.
20. Pada halaman Tinjau dan konfigurasi, tinjau detail konfigurasi aplikasi Anda. Jika Anda perlu mengubah salah satu pengaturan, pilih Edit untuk bagian yang ingin Anda edit dan buat perubahan.
21. Pilih Kirim. Halaman detail aplikasi yang baru saja Anda tambahkan ditampilkan.

Setelah menyiapkan aplikasi, pengguna dapat mengakses aplikasi Anda dari dalam portal AWS akses mereka berdasarkan set izin yang telah Anda buat dan akses pengguna yang telah Anda tetapkan.

Tambahkan S3 Access Grants sebagai aplikasi tepercaya untuk propagasi identitas

Setelah menyiapkan aplikasi yang dikelola pelanggan, Anda harus menentukan S3 Access Grants untuk propagasi identitas. S3 Access Grants memberikan kredensi bagi pengguna untuk mengakses data Amazon S3. Saat Anda masuk ke aplikasi yang dikelola pelanggan, S3 Access Grants akan meneruskan identitas pengguna Anda ke aplikasi tepercaya.

Prasyarat: Pastikan Anda telah menyiapkan S3 Access Grants (seperti membuat instance S3 Access Grants dan mendaftarkan lokasi) sebelum mengikuti langkah-langkah ini. Untuk informasi selengkapnya, lihat Memulai Hibah Akses S3.

Untuk menambahkan S3 Access Grants untuk propagasi identitas ke aplikasi yang dikelola pelanggan Anda, lakukan langkah-langkah berikut:

1. Buka konsol Pusat Identitas IAM.
2. Pilih Aplikasi.
3. Pilih tab yang dikelola Pelanggan.
4. Dalam daftar aplikasi terkelola Pelanggan, pilih aplikasi OAuth 2.0 yang ingin Anda lakukan untuk memulai permintaan akses. Ini adalah aplikasi yang akan digunakan pengguna Anda.
5. Pada halaman Detail, di bawah Aplikasi tepercaya untuk propagasi identitas, pilih Tentukan aplikasi tepercaya.

6. Di bawah Jenis pengaturan, pilih Aplikasi individual dan tentukan akses, lalu pilih Berikutnya.
7. Pada halaman Select Service, pilih S3 Access Grants. S3 Access Grants memiliki aplikasi yang dapat Anda gunakan untuk menentukan aplikasi web Anda sendiri untuk propagasi identitas tepercaya.
8. Pilih Berikutnya. Anda akan memilih aplikasi Anda di langkah berikutnya.
9. Pada halaman Pilih aplikasi, pilih Aplikasi individual, pilih kotak centang untuk setiap aplikasi yang dapat menerima permintaan akses, lalu pilih Berikutnya.
10. Pada halaman Konfigurasi akses, di bawah metode Konfigurasi, pilih salah satu dari berikut ini:

    - Pilih akses per aplikasi — Pilih opsi ini untuk mengonfigurasi tingkat akses yang berbeda untuk setiap aplikasi. Pilih aplikasi yang ingin Anda konfigurasikan tingkat aksesnya, lalu pilih Edit akses. Di Tingkat akses untuk diterapkan, ubah tingkat akses sesuai kebutuhan, lalu pilih Simpan perubahan.
    - Terapkan tingkat akses yang sama ke semua aplikasi — Pilih opsi ini jika Anda tidak perlu mengonfigurasi tingkat akses per aplikasi.
11. Pilih Berikutnya.
12. Pada halaman konfigurasi Tinjauan, tinjau pilihan yang Anda buat.

    > **Note**
    >
    > Anda akan ingin memastikan `s3:access_grants:read_write` izin diberikan untuk pengguna Anda. Izin ini memungkinkan pengguna Anda untuk mengambil kredensil untuk mengakses Amazon S3. Pastikan untuk menggunakan kebijakan IAM yang Anda buat sebelumnya, atau S3 Access Grants, untuk membatasi akses ke operasi penulisan.
13. Untuk membuat perubahan, pilih Edit untuk bagian konfigurasi yang ingin Anda ubah. Kemudian, buat perubahan yang diperlukan dan pilih Simpan perubahan.
14. Pilih aplikasi Trust untuk menambahkan aplikasi tepercaya untuk propagasi identitas.

Membuat hibah untuk pengguna atau grup

Pada langkah ini, Anda menggunakan IAM Identity Center untuk menyediakan pengguna Anda. Anda dapat menggunakan SCIM untuk penyediaan otomatis atau manual pengguna dan grup. SCIM menjaga identitas IAM Identity Center Anda tetap sinkron dengan identitas dari penyedia identitas Anda (iDP). Ini termasuk penyediaan, pembaruan, dan penonaktifan pengguna antara IDP dan Pusat Identitas IAM Anda.

> **Note**
>
> Langkah ini diperlukan karena ketika S3 Access Grants digunakan dengan IAM Identity Center, pengguna IAM Identity Center lokal tidak digunakan. Sebagai gantinya, pengguna harus disinkronkan dari penyedia identitas dengan IAM Identity Center.

Untuk menyinkronkan pengguna dari penyedia identitas Anda dengan IAM Identity Center, lakukan langkah-langkah berikut:

1. Aktifkan penyediaan otomatis.
2. Hasilkan token akses.

Untuk contoh cara mengatur penyedia identitas dengan IAM Identity Center untuk kasus penggunaan spesifik Anda, lihat tutorial sumber Identitas Pusat Identitas IAM.

Buat Browser Penyimpanan Anda untuk komponen S3

Setelah menyiapkan instans IAM Identity Center dan membuat hibah di S3 Access Grants, buka aplikasi React Anda. Dalam aplikasi React, gunakan `createManagedAuthAdapter` untuk mengatur aturan otorisasi. Anda harus menyediakan penyedia kredensi untuk mengembalikan kredensil yang Anda peroleh dari IAM Identity Center. Anda kemudian dapat memanggil `createStorageBrowser` untuk menginisialisasi Browser Penyimpanan untuk komponen S3:

```
import {
    createManagedAuthAdapter,
    createStorageBrowser,
} from '@aws-amplify/ui-react-storage/browser';
import '@aws-amplify/ui-react-storage/styles.css';

export const { StorageBrowser } = createStorageBrowser({
   config: createManagedAuthAdapter({
    credentialsProvider: async (options?: { forceRefresh?: boolean }) => {
      // return your credentials object
      return {
        credentials: {
          accessKeyId: 'my-access-key-id',
          secretAccessKey: 'my-secret-access-key',
          sessionToken: 'my-session-token',
          expiration: new Date(),
```

```
        },
      }
    },
    // AWS `region` and `accountId` of the S3 Access Grants Instance.
    region: '',
    accountId: '',
    // call `onAuthStateChange` when end user auth state changes
    // to clear sensitive data from the `StorageBrowser` state
    registerAuthListener: (onAuthStateChange) => {},
  })
});
```

Kemudian, buat mekanisme untuk menukar token web JSON (JWTs) dari aplikasi web Anda dengan kredensyal IAM dari IAM Identity Center. Untuk informasi selengkapnya tentang cara menukar JWT, lihat sumber daya berikut:

- Cara mengembangkan aplikasi data yang dihadapi pengguna dengan IAM Identity Center dan S3 Access Grants posting di Storage Blog AWS
- Menskalakan akses data dengan posting S3 Access Grants di Storage Blog AWS
- Lokakarya Hibah Akses S3 di AWS studio lokakarya
- Lokakarya Hibah Akses S3 tentang GitHub

Kemudian, siapkan titik akhir API untuk menangani permintaan untuk mengambil kredensil. Untuk memvalidasi pertukaran token web JSON (JWT), lakukan langkah-langkah berikut:

1. Ambil token web JSON dari header otorisasi untuk permintaan masuk.
2. Validasi token menggunakan kunci publik dari URL set kunci web JSON (JWKS) yang ditentukan.
3. Verifikasi kedaluwarsa token, penerbit, subjek, dan klaim audiens.

Untuk menukar token web JSON penyedia identitas dengan kredensil AWS IAM, lakukan langkah-langkah berikut:

> **Tip**
>
> Pastikan untuk menghindari mencatat informasi sensitif apa pun. Kami menyarankan Anda menggunakan kontrol penanganan kesalahan untuk otorisasi yang hilang, token

kedaluwarsa, dan pengecualian lainnya. Untuk informasi selengkapnya, lihat posting Menerapkan pola penanganan AWS Lambda kesalahan di AWS Compute Blog.

1. Verifikasi bahwa parameter Izin dan Lingkup yang diperlukan disediakan dalam permintaan.
2. Gunakan CreateTokenWithIAMAPI untuk menukar token web JSON dengan token IAM Identity Center.

> **Note**
>
> Setelah token web iDP JSON digunakan, token tersebut tidak dapat digunakan lagi. Token baru harus digunakan untuk bertukar dengan IAM Identity Center.

3. Gunakan operasi AssumeRoleAPI untuk mengambil peran sementara menggunakan token IAM Identity Center. Pastikan untuk menggunakan peran pembawa identitas, juga dikenal sebagai peran yang membawa konteks identitas (misalnya,**`identity-bearer-role`**) untuk meminta kredensialnya.
4. Kembalikan kredensi IAM ke aplikasi web.

> **Note**
>
> Pastikan Anda telah menyiapkan mekanisme logging yang tepat. Respons dikembalikan dalam format JSON standar dengan kode status HTTP yang sesuai.

Mengkonfigurasi Browser Penyimpanan untuk S3

Untuk mengizinkan Browser Penyimpanan untuk akses S3 ke bucket S3, komponen Storage Browser membuat panggilan REST API ke Amazon S3. Secara default, berbagi sumber daya lintas asal (CORS) tidak diaktifkan pada bucket S3. Akibatnya, Anda harus mengaktifkan CORS untuk setiap bucket S3 tempat Storage Browser mengakses data.

Misalnya, untuk mengaktifkan CORS di bucket S3, Anda dapat memperbarui kebijakan CORS seperti ini:

```
[
    {
        "ID": "S3CORSRuleId1",
```

```
        "AllowedHeaders": [
            "*"
        ],
        "AllowedMethods": [
            "GET",
            "HEAD",
            "PUT",
            "POST",
            "DELETE"
        ],
        "AllowedOrigins": [
            "*"
        ],
        "ExposeHeaders": [
            "last-modified",
            "content-type",
            "content-length",
            "etag",
            "x-amz-version-id",
            "x-amz-request-id",
            "x-amz-id-2",
            "x-amz-cf-id",
            "x-amz-storage-class",
            "date",
            "access-control-expose-headers"
        ],
        "MaxAgeSeconds": 3000
    }
]
```

Pemecahan Masalah Browser Penyimpanan untuk S3

Jika Anda mengalami masalah dengan Browser Penyimpanan untuk S3, pastikan untuk meninjau tips pemecahan masalah berikut:

- Hindari mencoba menggunakan token yang sama (`idTokenatauaccessToken`) untuk beberapa permintaan. Token tidak dapat digunakan kembali. Ini akan mengakibatkan kegagalan permintaan.
- Pastikan bahwa kredensi IAM yang Anda berikan ke komponen Browser Penyimpanan menyertakan izin untuk menjalankan operasi. `s3:GetDataAccess` Jika tidak, pengguna akhir Anda tidak akan dapat mengakses data Anda.

Atau, Anda dapat memeriksa sumber daya berikut:

- Browser Penyimpanan untuk S3 didukung oleh AWS Support. Jika Anda membutuhkan bantuan, hubungi AWS Support Center.
- Jika Anda mengalami masalah dengan Browser Penyimpanan untuk S3 atau ingin mengirimkan umpan balik, kunjungi halaman GitHub Amplify.
- Jika Anda menemukan potensi masalah keamanan dalam proyek ini, Anda dapat memberi tahu AWS Keamanan melalui halaman Pelaporan AWS Kerentanan.

Mengonfigurasi transfer file yang cepat dan aman menggunakan Amazon S3 Transfer Acceleration

Amazon S3 Transfer Acceleration adalah fitur tingkat ember yang memungkinkan transfer file yang cepat, mudah, dan aman dalam jarak jauh antara klien Anda dan bucket tujuan umum S3. Transfer Acceleration dirancang untuk mengoptimalkan kecepatan transfer dari seluruh dunia ke dalam bucket tujuan umum S3. Transfer Acceleration memanfaatkan lokasi edge yang didistribusikan secara global di Amazon CloudFront. Saat datanya tiba di lokasi edge, data tersebut diarahkan ke Amazon S3 melalui jalur jaringan yang dioptimalkan.

Saat Anda menggunakan Transfer Acceleration, biaya transfer data tambahan mungkin berlaku. Untuk informasi selengkapnya tentang harga, lihat Harga Amazon S3.

Mengapa menggunakan Percepatan Transfer?

Anda mungkin ingin menggunakan Transfer Acceleration pada bucket tujuan umum karena berbagai alasan:

- Pelanggan Anda mengunggah ke ember tujuan umum terpusat dari seluruh dunia.
- Anda mentransfer ukuran data gigabyte ke terabyte secara teratur di seluruh benua.
- Anda tidak dapat menggunakan semua bandwidth yang tersedia melalui internet saat mengunggah ke Amazon S3.

Untuk informasi selengkapnya tentang kapan menggunakan Akselerasi Transfer, lihat Amazon S3 FAQs.

Persyaratan untuk menggunakan Transfer Acceleration

Hal berikut ini diperlukan saat Anda menggunakan Transfer Acceleration pada bucket S3:

- Transfer Acceleration hanya didukung pada permintaan bergaya hosting virtual. Untuk informasi selengkapnya tentang permintaan gaya yang dihosting virtual, lihat Membuat permintaan menggunakan REST API di Referensi API Amazon S3.
- Nama bucket yang digunakan untuk Transfer Acceleration harus mematuhi DNS, dan tidak boleh mengandung titik (".").
- Transfer Acceleration harus diaktifkan pada bucket. Untuk informasi selengkapnya, lihat Mengaktifkan dan menggunakan S3 Transfer Acceleration.

  Setelah Anda mengaktifkan Transfer Acceleration pada bucket, mungkin diperlukan waktu hingga 20 menit sebelum kecepatan transfer data ke bucket tersebut meningkat.

  Note

  Transfer Acceleration saat ini tidak didukung untuk bucket yang terletak di Wilayah berikut:

  - Asia Pasifik (Tokyo) (ap-northeast-1)
  - Asia Pasifik (Seoul) (ap-northeast-2)
  - Asia Pasifik (Mumbai) (ap-south-1)
  - Asia Pasifik (Singapura) (ap-southeast-1)
  - Asia Pasifik (Sydney) (ap-southeast-2)
  - Kanada (Pusat) (ca-central-1)
  - Eropa (Frankfurt) (eu-central-1)
  - Eropa (Irlandia) (eu-west-1)
  - Eropa (London) (eu-west-2)
  - Eropa (Paris) (eu-west-3)
  - Amerika Selatan (Sao Paulo) (sa-east-1)
  - AS Timur (Virginia Utara) (us-east-1)
  - AS Timur (Ohio) (us-east-2)
  - AS Barat (California Utara) (us-west-1)
  - AS Barat (Oregon) (us-west-2)

- Untuk mengakses bucket yang diaktifkan untuk Transfer Acceleration, Anda harus menggunakan titik akhir *bucket-name*.s3-accelerate.amazonaws.com. Atau, gunakan titik akhir tumpukan ganda *bucket-name*.s3-accelerate.dualstack.amazonaws.com untuk terhubung ke bucket yang diaktifkan. IPv6 Anda dapat terus menggunakan titik akhir reguler untuk transfer data standar.
- Anda harus menjadi pemilik bucket untuk mengatur kondisi akselerasi transfer. Pemilik bucket dapat menetapkan izin kepada pengguna lain untuk memungkinkan mereka mengatur kondisi akselerasi pada bucket. `s3:PutAccelerateConfiguration` mengizinkan pengguna untuk mengaktifkan atau menonaktifkan Transfer Acceleration pada bucket. `s3:GetAccelerateConfiguration`Izin ini memungkinkan pengguna untuk mengembalikan status Akselerasi Transfer bucket, yaitu salah satu atau `Enabled` `Suspended`.

Bagian berikut ini menjelaskan bagaimana caranya untuk memulai dan menggunakan Amazon S3 Transfer Acceleration untuk mentransfer data.

Topik

- Memulai Amazon S3 Transfer Acceleration
- Mengaktifkan dan menggunakan S3 Transfer Acceleration
- Menggunakan Alat Perbandingan Kecepatan Amazon S3 Transfer Acceleration

Memulai Amazon S3 Transfer Acceleration

Amazon S3 Transfer Acceleration memungkinkan transfer file yang cepat, mudah, dan aman melalui jarak jauh antara klien Anda dan bucket S3. Transfer Acceleration menggunakan lokasi edge yang didistribusikan secara global di Amazon CloudFront. Saat datanya tiba di lokasi edge, data diarahkan ke Amazon S3 melalui jalur jaringan yang dioptimalkan.

Untuk mulai menggunakan Amazon S3 Transfer Acceleration, lakukan langkah-langkah berikut:

1. Aktifkan Transfer Acceleration pada bucket

   Anda dapat mengaktifkan Transfer Acceleration pada bucket dengan cara berikut ini:

   - Gunakan konsol Amazon S3.
   - Gunakan PutBucketAccelerateConfigurationoperasi REST API.

- Gunakan AWS CLI dan AWS SDKs. Untuk informasi selengkapnya, lihat Mengembangkan dengan Amazon S3 menggunakan Referensi API Amazon S3 AWS SDKs di Amazon S3.

Untuk informasi selengkapnya, lihat Mengaktifkan dan menggunakan S3 Transfer Acceleration.

> **Note**
>
> Agar bucket berfungsi dengan akselerasi transfer, nama bucket harus sesuai dengan persyaratan penamaan DNS dan tidak boleh berisi period ().

2. Transfer data ke dan dari bucket yang diaktifkan dengan akselerasi

Gunakan salah satu nama domain `s3-accelerate` endpoint berikut:

- Untuk mengakses bucket yang mendukung akselerasi, gunakan *`bucket-name`*`.s3-accelerate.amazonaws.com`.
- Untuk mengakses bucket over berkemampuan akselerasi IPv6, gunakan. *`bucket-name`*`.s3-accelerate.dualstack.amazonaws.com`

  Titik akhir tumpukan ganda Amazon S3 mendukung permintaan ke bucket S3 secara berulang-ulang. IPv6 IPv4 Titik akhir dual-stack Transfer Acceleration hanya menggunakan jenis nama titik akhir yang dihosting bergaya virtual. Untuk informasi selengkapnya, lihat Membuat permintaan ke Amazon S3 IPv6 di Referensi API Amazon S3 dan Menggunakan titik akhir tumpukan ganda Amazon S3 di Referensi API Amazon S3.

> **Note**
>
> Aplikasi transfer data Anda harus menggunakan salah satu dari dua jenis titik akhir berikut untuk mengakses bucket agar transfer data lebih cepat: `.s3-accelerate.amazonaws.com` atau `.s3-accelerate.dualstack.amazonaws.com` untuk titik akhir tumpukan ganda. Anda dapat terus menggunakan titik akhir reguler untuk transfer data standar.

Anda dapat mengarahkan objek Amazon S3 dan permintaan PUT GET objek ke nama domain `s3-accelerate` endpoint setelah Anda mengaktifkan Transfer Acceleration. Misalnya, misalkan saat ini Anda memiliki aplikasi REST API PutObjectyang menggunakan nama host *`amzn-s3-demo-bucket`*`.s3.us-east-1.amazonaws.com` dalam PUT permintaan. Untuk mempercepat PUT, Anda mengubah nama host dalam permintaan Anda ke *`amzn-s3-demo-bucket`*`.s3-`

`accelerate.amazonaws.com`. Untuk kembali menggunakan kecepatan pengunggahan standar, ubah namanya kembali ke *`amzn-s3-demo-bucket`*`.s3.us-east-1.amazonaws.com`.

Setelah Transfer Acceleration diaktifkan, mungkin memerlukan waktu hingga 20 menit untuk mewujudkan manfaat performanya. Namun, titik akhir akselerasi tersedia segera setelah Anda mengaktifkan Transfer Acceleration.

Anda dapat menggunakan titik akhir percepatan di AWS CLI, AWS SDKs, dan alat lain yang mentransfer data ke dan dari Amazon S3. Jika Anda menggunakan AWS SDKs, beberapa bahasa yang didukung menggunakan flag konfigurasi klien titik akhir percepatan sehingga Anda tidak perlu secara eksplisit menyetel titik akhir untuk Transfer Acceleration. *`bucket-name`*`.s3-accelerate.amazonaws.com` Untuk contoh cara menggunakan flag konfigurasi klien titik akhir yang dipercepat, lihat Mengaktifkan dan menggunakan S3 Transfer Acceleration.

Anda dapat menggunakan semua operasi Amazon S3 melalui titik akhir percepatan transfer kecuali untuk hal berikut ini:

- ListBuckets
- CreateBucket
- DeleteBucket

Selain itu, Amazon S3 Transfer Acceleration tidak mendukung penggunaan salinan lintas wilayah. CopyObject

Mengaktifkan dan menggunakan S3 Transfer Acceleration

Anda dapat menggunakan Amazon S3 Transfer Acceleration untuk mentransfer file dengan cepat dan aman dalam jarak yang jauh antara klien Anda dan bucket tujuan umum S3. Anda dapat mengaktifkan Transfer Acceleration menggunakan konsol S3, AWS Command Line Interface (AWS CLI), API, atau. AWS SDKs

Bagian ini memberikan contoh cara mengaktifkan Amazon S3 Transfer Acceleration pada bucket dan menggunakan titik akhir akselerasi untuk bucket yang diaktifkan.

Untuk informasi lebih lanjut tentang persyaratan Transfer Acceleration, lihat Mengonfigurasi transfer file yang cepat dan aman menggunakan Amazon S3 Transfer Acceleration.

Menggunakan konsol S3

> **Note**
>
> Jika Anda ingin membandingkan kecepatan pengunggahan yang dipercepat dan yang tidak dipercepat, buka Alat Perbandingan Kecepatan Amazon S3 Transfer Acceleration.
> Alat Perbandingan Kecepatan menggunakan unggahan multibagian untuk mentransfer file dari browser Anda ke berbagai Wilayah AWS dengan dan tanpa akselerasi transfer Amazon S3. Anda dapat membandingkan kecepatan unggahan untuk pengunggahan langsung dan mentransfer unggahan yang dipercepat oleh Wilayah.

Untuk mengaktifkan akselerasi transfer untuk bucket tujuan umum S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Dalam daftar Bucket tujuan umum, pilih nama bucket yang ingin Anda aktifkan akselerasi transfer.
4. Pilih Properti.
5. Di bagian bawah Akselerasi transfer, pilih Edit.
6. Pilih Aktifkan, dan pilih Simpan perubahan.

Untuk mengakses transfer data yang dipercepat

1. Setelah Amazon S3 mengaktifkan akselerasi transfer untuk bucket Anda, lihat tab Properti untuk bucket tersebut.
2. Di bagian bawah Akselerasi transfer, Titik akhir yang dipercepat menampilkan titik akhir akselerasi transfer untuk bucket Anda. Gunakan titik akhir ini untuk mengakses transfer data yang dipercepat ke dan dari bucket Anda.

   Jika Anda menangguhkan akselerasi transfer, titik akhir percepatan tidak lagi bekerja.

Menggunakan AWS CLI

Berikut ini adalah contoh AWS CLI perintah yang digunakan untuk Transfer Acceleration. Untuk petunjuk cara menyiapkan AWS CLI, lihat Mengembangkan dengan Amazon S3 menggunakan Referensi API Amazon S3 AWS CLI di Amazon S3.

Mengaktifkan Transfer Acceleration pada bucket

Gunakan AWS CLI put-bucket-accelerate-configurationperintah untuk mengaktifkan atau menangguhkan Transfer Acceleration pada bucket.

Contoh berikut ditetapkan `Status=Enabled` untuk mengaktifkan Transfer Acceleration pada bucket bernama*`amzn-s3-demo-bucket`*. Untuk menangguhkan Transfer Acceleration, gunakan`Status=Suspended`.

Example

```
$ aws s3api put-bucket-accelerate-configuration --bucket amzn-s3-demo-bucket --
accelerate-configuration Status=Enabled
```

Menggunakan Transfer Acceleration

Anda dapat mengarahkan semua permintaan Amazon S3 yang dibuat oleh `s3` dan `s3api` AWS CLI perintah ke titik akhir percepatan:. `s3-accelerate.amazonaws.com` Untuk melakukan ini, atur nilai konfigurasi `use_accelerate_endpoint` ke `true` dalam profil di AWS Config file Anda. Transfer Acceleration harus diaktifkan pada bucket Anda untuk menggunakan titik akhir akselerasi.

Semua permintaan dikirim menggunakan pengalamatan bucket bergaya virtual: *`amzn-s3-demo-bucket`*`.s3-accelerate.amazonaws.com`. Setiap permintaan `ListBuckets`, `CreateBucket`, dan `DeleteBucket` tidak dikirim ke titik akhir percepatan karena titik akhir tidak mendukung operasi tersebut.

Untuk informasi selengkapnya tentang `use_accelerate_endpoint`, lihat AWS CLI Konfigurasi S3 dalam AWS CLI Referensi Perintah.

Contoh berikut ini menetapkan `use_accelerate_endpoint` ke `true` di profil default.

Example

```
$ aws configure set default.s3.use_accelerate_endpoint true
```

Jika Anda ingin menggunakan titik akhir percepatan untuk beberapa AWS CLI perintah tetapi tidak yang lain, Anda dapat menggunakan salah satu dari dua metode berikut:

- Gunakan titik akhir akselerasi untuk `s3api` perintah apa pun `s3` atau dengan menyetel `--endpoint-url` parameter kehttps://s3-accelerate.amazonaws.com.
- Siapkan profil terpisah di AWS Config file Anda. Misalnya, buat satu profil yang mengatur `use_accelerate_endpoint` hingga `true` dan profil yang tidak mengatur `use_accelerate_endpoint`. Ketika Anda menjalankan perintah, tentukan profil mana yang ingin Anda gunakan, tergantung pada apakah Anda ingin menggunakan titik akhir tumpukan ganda atau tidak.

Mengunggah objek ke bucket yang diaktifkan untuk Transfer Acceleration

Contoh berikut mengunggah file ke bucket bernama *amzn-s3-demo-bucket* yang telah diaktifkan untuk Transfer Acceleration dengan menggunakan profil default yang telah dikonfigurasi untuk menggunakan endpoint percepatan.

Example

```
$ aws s3 cp file.txt s3://amzn-s3-demo-bucket/key-name --region region
```

Contoh berikut ini mengunggah file ke bucket yang diaktifkan untuk Transfer Acceleration dengan menggunakan `--endpoint-url` untuk menentukan titik akhir akselerasi.

Example

```
$ aws configure set s3.addressing_style virtual
$ aws s3 cp file.txt s3://amzn-s3-demo-bucket/key-name --region region --endpoint-url
 https://s3-accelerate.amazonaws.com
```

Menggunakan AWS SDKs

Berikut ini adalah contoh penggunaan Transfer Acceleration untuk mengunggah objek ke Amazon S3 menggunakan SDK. AWS Beberapa bahasa yang didukung AWS SDK (misalnya, Java dan.NET) menggunakan flag konfigurasi klien titik akhir percepatan sehingga Anda tidak perlu secara eksplisit menyetel titik akhir untuk Transfer Acceleration. *bucket-name*`.s3-accelerate.amazonaws.com`

Java

Example

Contoh berikut menunjukkan cara menggunakan titik akhir yang dipercepat untuk mengunggah objek ke Amazon S3. Contoh ini melakukan hal berikut:

- Membuat `AmazonS3Client` yang dikonfigurasi untuk menggunakan titik akhir yang dipercepat. Semua bucket yang diakses klien harus mengaktifkan Transfer Acceleration.
- Memungkinkan Transfer Acceleration pada bucket tertentu. Langkah ini diperlukan hanya jika bucket yang Anda tentukan belum mengaktifkan Transfer Acceleration.
- Memverifikasi bahwa akselerasi transfer diaktifkan untuk bucket tertentu.
- Mengunggah objek baru ke bucket tertentu menggunakan titik akhir akselerasi bucket.

Untuk informasi lebih lanjut tentang Transfer Acceleration, lihat Memulai Amazon S3 Transfer Acceleration. Untuk petunjuk cara membuat dan menguji sampel yang berfungsi, lihat Memulai di Panduan AWS SDK untuk Java Pengembang.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.BucketAccelerateConfiguration;
import com.amazonaws.services.s3.model.BucketAccelerateStatus;
import com.amazonaws.services.s3.model.GetBucketAccelerateConfigurationRequest;
import com.amazonaws.services.s3.model.SetBucketAccelerateConfigurationRequest;

public class TransferAcceleration {
    public static void main(String[] args) {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String keyName = "*** Key name ***";

        try {
            // Create an Amazon S3 client that is configured to use the accelerate
 endpoint.
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withRegion(clientRegion)
```

```
                    .withCredentials(new ProfileCredentialsProvider())
                    .enableAccelerateMode()
                    .build();

            // Enable Transfer Acceleration for the specified bucket.
            s3Client.setBucketAccelerateConfiguration(
                    new SetBucketAccelerateConfigurationRequest(bucketName,
                            new BucketAccelerateConfiguration(
                                    BucketAccelerateStatus.Enabled)));

            // Verify that transfer acceleration is enabled for the bucket.
            String accelerateStatus = s3Client.getBucketAccelerateConfiguration(
                    new GetBucketAccelerateConfigurationRequest(bucketName))
                    .getStatus();
            System.out.println("Bucket accelerate status: " + accelerateStatus);

            // Upload a new object using the accelerate endpoint.
            s3Client.putObject(bucketName, keyName, "Test object for transfer
 acceleration");
            System.out.println("Object \"" + keyName + "\" uploaded with transfer
 acceleration.");
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

Contoh berikut menunjukkan cara menggunakan AWS SDK untuk .NET untuk mengaktifkan Transfer Acceleration pada bucket. Untuk informasi tentang menyiapkan dan menjalankan contoh kode, lihat Memulai AWS SDK untuk .NET di Panduan AWS SDK untuk .NET Pengembang.

Example

```
using Amazon;
using Amazon.S3;
```

```
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class TransferAccelerationTest
    {
        private const string bucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            EnableAccelerationAsync().Wait();
        }

        static async Task EnableAccelerationAsync()
        {
                try
                {
                    var putRequest = new PutBucketAccelerateConfigurationRequest
                    {
                        BucketName = bucketName,
                        AccelerateConfiguration = new AccelerateConfiguration
                        {
                            Status = BucketAccelerateStatus.Enabled
                        }
                    };
                    await
 s3Client.PutBucketAccelerateConfigurationAsync(putRequest);

                    var getRequest = new GetBucketAccelerateConfigurationRequest
                    {
                        BucketName = bucketName
                    };
                    var response = await
 s3Client.GetBucketAccelerateConfigurationAsync(getRequest);

                    Console.WriteLine("Acceleration state = '{0}' ",
 response.Status);
                }
```

```
                catch (AmazonS3Exception amazonS3Exception)
                {
                    Console.WriteLine(
                        "Error occurred. Message:'{0}' when setting transfer
 acceleration",
                        amazonS3Exception.Message);
                }
        }
    }
}
```

Saat mengunggah objek ke bucket yang mengaktifkan Transfer Acceleration, Anda menentukan menggunakan titik akhir akselerasi pada saat sedang membuat klien.

```
var client = new AmazonS3Client(new AmazonS3Config
            {
                RegionEndpoint = TestRegionEndpoint,
                UseAccelerateEndpoint = true
            }
```

JavaScript

Untuk contoh mengaktifkan Transfer Acceleration menggunakan AWS SDK untuk JavaScript, lihat PutBucketAccelerateConfigurationperintah di Referensi AWS SDK untuk JavaScript API.

Python (Boto)

Untuk contoh mengaktifkan Transfer Acceleration dengan menggunakan SDK untuk Python, lihat put_bucket_accelerate_configuration di SDK for Python AWS (Boto3) API Referensi.

Other

Untuk informasi tentang menggunakan lainnya AWS SDKs, lihat Contoh Kode dan Pustaka.

Penggunaan API REST

Gunakan operasi API REST `PutBucketAccelerateConfiguration` untuk mengaktifkan konfigurasi akselerasi pada bucket yang ada.

Untuk informasi selengkapnya, lihat PutBucketAccelerateConfiguration dalam Referensi API Amazon Simple Storage Service.

Menggunakan Alat Perbandingan Kecepatan Amazon S3 Transfer Acceleration

Anda dapat menggunakan Amazon S3 Transfer Acceleration Speed Comparison tool untuk membandingkan kecepatan unggahan yang dipercepat dan tidak dipercepat di seluruh Wilayah Amazon S3. Alat Perbandingan Kecepatan menggunakan unggahan multibagian untuk mentransfer file dari browser Anda ke berbagai Wilayah Amazon S3 dengan dan tanpa menggunakan Akselerasi Transfer.

Anda dapat mengakses alat Perbandingan Kecepatan dengan menggunakan salah satu metode berikut:

- Salin URL berikut ke jendela browser Anda, ganti *region* dengan Wilayah AWS yang Anda gunakan (misalnya,us-west-2) dan *amzn-s3-demo-bucket* dengan nama bucket yang ingin Anda evaluasi:

  ```
  https://s3-accelerate-speedtest.s3-accelerate.amazonaws.com/en/
  accelerate-speed-comparsion.html?region=region&origBucketName=amzn-s3-
  demo-bucket
  ```

  Untuk daftar titik akhir dan Wilayah Amazon S3, lihat Titik akhir dan kuota Amazon S3 di Referensi Umum AWS.
- Gunakan konsol Amazon S3.

Menggunakan Requester Pays ember tujuan umum untuk transfer dan penggunaan penyimpanan

Secara umum, pemilik bucket membayar semua biaya penyimpanan dan transfer data Amazon S3 yang terkait dengan bucket mereka. Namun, Anda dapat mengonfigurasi bucket tujuan umum menjadi bucket Requester Pays. Dengan bucket Pembayaran Pemohon, pemohon alih-alih pemilik bucket membayar biaya permintaan dan data unduhan dari bucket. Pemilik bucket selalu membayar biaya penyimpanan data.

Biasanya, Anda mengonfigurasi bucket agar menjadi Pembayaran Pemohon ketika Anda ingin berbagi data, tetapi tidak mengeluarkan biaya yang terkait dengan orang lain yang mengakses data tersebut. Sebagai contoh, Anda dapat menggunakan bucket Pembayaran Pemohon saat

menyediakan set data besar, seperti direktori kode zip, data referensi, informasi geospasial, atau data perayapan web.

> **Important**
>
> Jika Anda mengaktifkan Requester Pays pada bucket tujuan umum, akses anonim ke bucket tersebut tidak diperbolehkan.

Anda harus mengautentikasi semua permintaan yang melibatkan bucket Pembayaran Pemohon. Autentikasi permintaan memungkinkan Amazon S3 untuk mengidentifikasi dan mengenakan biaya atas penggunaan mereka atas bucket Pembayaran Pemohon.

Ketika pemohon mengambil peran AWS Identity and Access Management (IAM) sebelum membuat permintaan mereka, akun tempat peran tersebut dibebankan untuk permintaan tersebut. Untuk informasi selengkapnya tentang peran IAM, lihat Peran IAM dalam Panduan Pengguna IAM.

Setelah Anda mengonfigurasi bucket untuk menjadi bucket Requester Pays, pemohon harus menunjukkan bahwa mereka memahami bahwa mereka akan dikenakan biaya untuk permintaan dan untuk pengunduhan data. Untuk menunjukkan bahwa mereka menerima tagihan, pemohon harus menyertakan `x-amz-request-payer` sebagai header dalam permintaan API mereka untuk permintaan DELETE, GET, HEAD, POST, dan PUT, atau menambahkan `RequestPayer` parameter dalam permintaan REST mereka. Untuk permintaan CLI, pemohon dapat menggunakan parameter. `--request-payer`

Example — Menggunakan Requester Pays saat menghapus objek

Untuk menggunakan contoh DeleteObjectVersionAPI berikut, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
DELETE /Key+?versionId=VersionId HTTP/1.1
Host: Bucket.s3.amazonaws.com
x-amz-mfa: MFA
x-amz-request-payer: RequestPayer
x-amz-bypass-governance-retention: BypassGovernanceRetention
x-amz-expected-bucket-owner: ExpectedBucketOwner
```

Jika pemohon mengembalikan objek dengan menggunakan RestoreObjectAPI, Requester Pays didukung selama `x-amz-request-payer` header atau `RequestPayer` parameter ada dalam

permintaan; namun, pemohon hanya membayar biaya permintaan. Pemilik ember membayar biaya pengambilan.

Bucket Pembayaran Pemohon tidak mendukung hal berikut:

- Permintaan anonim
- Permintaan SOAP
- Menggunakan bucket Pembayaran Pemohon sebagai bucket tujuan untuk pencatatan log pengguna akhir, atau sebaliknya. Namun, Anda dapat mengaktifkan pembuatan log pengguna akhir pada bucket Pembayaran Pemohon jika bucket targetnya bukan merupakan bucket Pembayaran Pemohon.

Cara kerja Pembayaran Pemohon

Biaya untuk permintaan Pembayaran Pemohon yang berhasil sangatlah sederhana: pemohon membayar transfer data dan permintaan tersebut; pemilik bucket membayar penyimpanan datanya. Namun, pemilik bucket dibebankan untuk permintaan berdasarkan ketentuan berikut:

- Permintaan mengembalikan kesalahan `AccessDenied` (HTTP`403 Forbidden`) dan permintaan dimulai di dalam AWS akun atau AWS organisasi individu pemilik bucket.
- Permintaan tersebut adalah permintaan SOAP.

Untuk informasi selengkapnya tentang Pembayaran Pemohon, lihat topik berikut.

Topik

- Mengonfigurasi Pembayaran Pemohon pada bucket
- Mengambil konfigurasi requestPayment menggunakan API REST
- Mengunduh objek dari ember Requester Pays

Mengonfigurasi Pembayaran Pemohon pada bucket

Anda mengonfigurasikan bucket Amazon S3 untuk menjadi bucket Pembayaran Pemohon, sehingga pemohon membayar biaya permintaan dan pengunduhan data alih-alih pemilik bucket yang melakukannya.

Bagian ini menyediakan contoh bagaimana mengonfigurasi Pembayaran Pemohon pada bucket Amazon S3 menggunakan konsol tersebut dan API REST.

Menggunakan konsol S3

Untuk mengaktifkan Requester Pays untuk bucket tujuan umum S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Dalam daftar Bucket tujuan umum, pilih nama bucket yang ingin Anda aktifkan Requester Pays for.
4. Pilih Properti.
5. Pada Pembayaran Pemohon, pilih Edit.
6. Pilih Aktifkan, dan pilih Simpan perubahan.

   Amazon S3 memungkinkan Pembayaran Pemohon untuk bucket Anda, dan menampilkan Ikhtisar bucket. Pada Pembayaran Pemohon, Anda melihat Diaktifkan.

Penggunaan API REST

Hanya pemilik bucket yang dapat mengatur nilai konfigurasi `RequestPaymentConfiguration.payer` dari bucket menjadi `BucketOwner` (default) atau `Requester`. Mengatur sumber daya `requestPayment` bersifat opsional. Secara default, bucket bukan merupakan bucket Pembayaran Pemohon.

Untuk mengembalikan bucket Pembayaran Pemohon ke bucket biasa, Anda menggunakan nilai `BucketOwner`. Biasanya, Anda akan menggunakan `BucketOwner` ketika mengunggah data ke bucket Amazon S3, dan kemudian Anda akan menetapkan nilai untuk `Requester` sebelum menerbitkan objek di dalam bucket.

Untuk mengatur requestPayment

- Gunakan permintaan PUT untuk mengatur nilai `Payer` ke `Requester` pada bucket tertentu.

```
PUT ?requestPayment HTTP/1.1
Host: [BucketName].s3.amazonaws.com
Content-Length: 173
```

```
Date: Wed, 01 Mar 2009 12:00:00 GMT
Authorization: AWS [Signature]

<RequestPaymentConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<Payer>Requester</Payer>
</RequestPaymentConfiguration>
```

Jika permintaannya berhasil, Amazon S3 mengembalikan tanggapan yang serupa dengan yang berikut ini.

```
HTTP/1.1 200 OK
x-amz-id-2: [id]
x-amz-request-id: [request_id]
Date: Wed, 01 Mar 2009 12:00:00 GMT
Content-Length: 0
Connection: close
Server: AmazonS3
x-amz-request-charged:requester
```

Anda dapat mengatur Pembayaran Pemohon hanya pada tingkatan bucket. Anda tidak dapat mengatur Pembayaran Pemohon untuk objek tertentu di dalam bucket.

Anda dapat mengonfigurasi bucket agar menjadi `BucketOwner` atau `Requester` kapan pun. Namun, mungkin beberapa menit dibutuhkan sebelum nilai konfigurasi barunya mulai berlaku.

> **Note**
>
> Pemilik bucket yang memberikan presigned URLs harus mempertimbangkan dengan cermat sebelum mengonfigurasi bucket menjadi Requester Pays, terutama jika URL memiliki masa pakai yang lama. Pemilik bucket dibebankan setiap kali pemohon menggunakan URL yang telah ditandatangani sebelumnya yang menggunakan kredensial pemilik bucket.

Mengambil konfigurasi requestPayment menggunakan API REST

Anda dapat menentukan nilai `Payer` yang ditetapkan di bucket dengan meminta sumber daya `requestPayment`.

Untuk mengembalikan sumber daya requestPayment

- Gunakan permintaan GET untuk mendapatkan sumber daya `requestPayment`, seperti yang ditunjukkan pada permintaan berikut ini.

```
GET ?requestPayment HTTP/1.1
Host: [BucketName].s3.amazonaws.com
Date: Wed, 01 Mar 2009 12:00:00 GMT
Authorization: AWS [Signature]
```

Jika permintaannya berhasil, Amazon S3 mengembalikan tanggapan yang serupa dengan yang berikut ini.

```
HTTP/1.1 200 OK
x-amz-id-2: [id]
x-amz-request-id: [request_id]
Date: Wed, 01 Mar 2009 12:00:00 GMT
Content-Type: [type]
Content-Length: [length]
Connection: close
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<RequestPaymentConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<Payer>Requester</Payer>
</RequestPaymentConfiguration>
```

Jawaban ini menunjukkan bahwa nilai `payer` ditetapkan menjadi `Requester`.

Mengunduh objek dari ember Requester Pays

Karena pemohon dikenai biaya untuk mengunduh data melalui bucket Pembayaran Pemohon, permintaan harus mengandung parameter kustom, `x-amz-request-payer`, yang mengonfirmasi bahwa pemohon mengetahui bahwa mereka akan dikenakan biaya untuk pengunduhan tersebut. Untuk mengakses objek di bucket Pembayaran Pemohon, permintaan harus mencakup salah satu dari hal berikut.

- Untuk permintaan DELETE, GET, HEAD, POST, dan PUT, sertakan `x-amz-request-payer : requester` di header
- Untuk ditandatangani URLs, sertakan `x-amz-request-payer=requester` dalam permintaan

Jika permintaan berhasil dan pemohon dikenai biaya, responsnya menyertakan header `x-amz-request-charged:requester`. Jika `x-amz-request-payer` tidak ada dalam permintaannya, Amazon S3 mengembalikan kesalahan 403 dan menagih pemilik bucket untuk permintaan tersebut.

> **Note**
>
> Pemilik bucket tidak perlu menambahkan `x-amz-request-payer` atas permintaan mereka. Pastikan bahwa Anda telah menyertakan `x-amz-request-payer` dan nilainya dalam perhitungan tanda tangan Anda. Untuk informasi selengkapnya, lihat Menggunakan Header Otorisasi di Referensi API Amazon S3.

Penggunaan API REST

Untuk mengunduh objek dari bucket Pembayaran Pemohon

- Gunakan permintaan `GET` untuk mengunduh sebuah objek dari bucket Pemohon Membayar, seperti yang ditunjukkan dalam permintaan berikut.

```
GET / [destinationObject] HTTP/1.1
Host: [BucketName].s3.amazonaws.com
x-amz-request-payer : requester
Date: Wed, 01 Mar 2009 12:00:00 GMT
Authorization: AWS [Signature]
```

Jika permintaan GET berhasil dan pemohon dikenai biaya, responsnya menyertakan `x-amz-request-charged:requester`.

Amazon S3 dapat mengembalikan kesalahan `Access Denied` pada permintaan yang mencoba untuk mendapatkan objek dari bucket Pembayaran Pemohon. Untuk informasi selengkapnya, lihat Respons Kesalahan dalam Referensi API Amazon Simple Storage Service.

Menggunakan AWS CLI

Untuk mengunduh objek dari bucket Requester Pays menggunakan AWS CLI, Anda tentukan `--request-payer requester` sebagai bagian dari `get-object` permintaan Anda. Untuk informasi selengkapnya, lihat get-object dalam Referensi AWS CLI .

Bekerja dengan objek di Amazon S3

Untuk menyimpan data Anda di Amazon S3, Anda bekerja dengan sumber daya yang dikenal sebagai bucket dan objek. Bucket adalah kontainer untuk objek. Objek adalah file data dan metadata apa pun yang mendeskripsikan file tersebut.

Untuk menyimpan objek di Amazon S3, Anda membuat bucket dan kemudian mengunggah objek tersebut ke dalam bucket. Saat objek berada di dalam bucket, Anda bisa membuka, mengunduh, dan menyalinnya. Bila Anda tidak lagi membutuhkan objek atau bucket, Anda bisa membersihkan sumber daya ini.

> **Note**
>
> Untuk informasi selengkapnya tentang menggunakan kelas penyimpanan Amazon S3 Express One Zone dengan bucket direktori, lihat S3 Express One Zone dan Bekerja dengan bucket direktori.

> **Important**
>
> Di konsol Amazon S3, saat Anda memilih Buka atau Unduh Sebagai objek, operasi ini membuat presigned. URLs Selama lima menit, objek Anda akan dapat diakses oleh siapa saja yang memiliki akses ke presigned URLs ini. Untuk informasi selengkapnya tentang presigned URLs, lihat Menggunakan URL yang telah ditetapkan sebelumnya.

Dengan Amazon S3, Anda hanya membayar sesuai penggunaan Anda. Untuk informasi selengkapnya tentang fitur dan harga Amazon S3, lihat Amazon S3. Jika Anda adalah pelanggan baru Amazon S3, Anda dapat memulai Amazon S3 secara gratis. Untuk informasi selengkapnya, lihat AWS Tingkat Gratis.

Topik

- Gambaran umum objek Amazon S3
- Menamai objek Amazon S3
- Bekerja dengan metadata objek
- Mengunggah Objek

- Tambahkan prasyarat ke operasi S3 dengan permintaan bersyarat
- Menyalin, memindahkan, dan mengganti nama objek
- Mengunggah objek
- Memeriksa integritas objek di Amazon S3
- Menghapus objek Amazon S3
- Mengatur, membuat daftar, dan bekerja dengan objek Anda
- Unduh dan unggah objek dengan presigned URLs
- Mengubah objek dengan S3 Lambda Objek
- Melakukan operasi objek secara massal dengan Operasi Batch
- Kueri data di tempat dengan Amazon S3 Select

Gambaran umum objek Amazon S3

Amazon S3 adalah penyimpanan objek yang menggunakan nilai kunci yang unik untuk menyimpan berapa pun objek yang Anda inginkan. Anda dapat menyimpan objek ini di dalam satu bucket atau lebih, dan setiap objek dapat berukuran hingga 5 TB. Sebuah objek terdiri atas beberapa hal berikut ini:

Kunci

Nama yang Anda tetapkan ke sebuah objek. Anda harus menggunakan kunci objek tersebut untuk mengambil objeknya. Untuk informasi lebih lanjut, lihat Bekerja dengan objek metadata.

ID Versi

Dalam sebuah bucket, kunci dan ID versi akan mengidentifikasikan sebuah objek secara unik. ID versi adalah sebuah string yang dibuat oleh Amazon S3 saat Anda menambahkan sebuah objek ke sebuah bucket. Untuk informasi selengkapnya, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.

Nilai

Konten yang Anda simpan.

Sebuah nilai objek dapat berupa urutan byte mana pun. Objek dapat memiliki rentang ukuran yang bervariasi dari nol hingga 5 TB. Untuk informasi selengkapnya, lihat Mengunggah Objek.

Metadata

Seperangkat pasangan nama-nilai yang dapat Anda gunakan untuk menyimpan informasi terkait objek. Anda dapat menetapkan metadata, yang disebut sebagai metadata yang ditentukan pengguna, ke objek Anda dalam Amazon S3. Amazon S3 juga akan menetapkan sistem-metadata ke objek-objek ini, yang akan digunakan untuk mengelola objek. Untuk informasi lebih lanjut, lihat Bekerja dengan objek metadata.

Sub-sumber daya

Amazon S3 menggunakan mekanisme sub-sumber daya untuk menyimpan informasi tambahan terkait objek. Karena sub-sumber daya merupakan subordinat objek, sub-sumber daya selalu dikaitkan dengan beberapa entitas lain seperti objek atau bucket. Untuk informasi selengkapnya, lihat Sub-sumber daya objek.

Informasi kontrol akses

Anda dapat mengontrol akses ke objek yang Anda simpan dalam Amazon S3. Amazon S3 mendukung kontrol akses berbasis sumber daya, seperti daftar kontrol akses (ACL) dan kebijakan bucket, dan kontrol akses berbasis pengguna. Untuk informasi selengkapnya tentang kontrol akses, lihat hal berikut ini:

- Kontrol akses di Amazon S3
- Identity and Access Management untuk Amazon S3
- Mengkonfigurasi ACLs

Sumber daya Amazon S3 Anda (misalnya, bucket dan objek) secara default bersifat pribadi. Anda harus secara eksplisit memberikan izin kepada orang lain untuk mengakses sumber daya ini. Untuk informasi lebih lanjut tentang objek, lihat Berbagi objek dengan presigned URLs.

Tanda

Anda dapat menggunakan tanda untuk mengategorikan objek yang Anda simpan, untuk mengontrol akses, atau mengalokasikan biaya. Untuk informasi selengkapnya, lihat Mengategorikan penyimpanan Anda menggunakan tag.

Sub-sumber daya objek

Amazon S3 menentukan seperangkat sub-sumber daya yang berkaitan dengan bucket dan objek. Sub-sumber daya merupakan subordinat objek. Artinya, sub-sumber daya tidak dapat muncul

dengan sendirinya. Sub-sumber daya selalu dikaitkan dengan beberapa entitas lain, seperti objek atau bucket.

Tabel berikut ini mencantumkan sub-sumber daya yang terkait dengan objek Amazon S3.

Sub-sumber daya	Deskripsi
acl	Berisi daftar izin yang mengidentifikasi penerima izin serta izin yang diberikan. Saat Anda menciptakan sebuah objek, `acl` mengidentifikasi pemilik objek sebagai pemilik dengan kendali penuh atas objek. Anda dapat mengambil ACL objek atau menggantinya dengan daftar izin yang diperbarui. Setiap pembaruan terhadap ACL mengharuskan Anda untuk mengganti ACL yang ada. Untuk informasi lebih lanjut tentang ACLs, lihatGambaran umum daftar kontrol akses (ACL).

Menamai objek Amazon S3

Sebuah kunci objek (atau nama kunci) secara unik mengidentifikasi objek dalam sebuah bucket Amazon S3. Saat Anda membuat objek, Anda menentukan nama kunci. Misalnya, di konsol Amazon S3, saat Anda memilih bucket, daftar objek di bucket akan muncul. Nama-nama ini merupakan kunci objek.

Nama kunci objek terdiri dari urutan karakter Unicode yang dikodekan dalam UTF-8, dengan panjang maksimum 1.024 byte atau sekitar 1.024 karakter Latin. Di beberapa lokal, satu karakter mungkin memerlukan 2 byte untuk pengkodean. Saat memberi nama objek Anda, perhatikan hal-hal berikut:

- Nama kunci objek peka terhadap huruf besar atau kecil.
- Nama kunci objek mencakup awalan apa pun (dikenal sebagai folder di konsol). Misalnya, `Development/Projects.xls` adalah nama kunci objek lengkap dari `Projects.xls` objek yang terletak di dalam `Development` awalan (atau folder). Awalan, pembatas (`/`), dan nama objek termasuk dalam batasan 1.024 byte untuk nama kunci objek. Untuk informasi selengkapnya tentang awalan dan folder, lihat. the section called “Memilih nama kunci objek”
- Karakter tertentu mungkin memerlukan penanganan khusus ketika mereka digunakan dalam nama kunci objek. Untuk informasi selengkapnya, lihat the section called “Panduan penamaan kunci objek”.

> **Note**
>
> Nama kunci objek dengan nilai `"soap"` tidak didukung untuk virtual-hosted-style permintaan. Untuk nilai nama kunci objek di mana `"soap"` digunakan, URL gaya jalur harus digunakan sebagai gantinya.

Memilih nama kunci objek

Model data Amazon S3 bersifat struktur yang datar: Anda membuat bucket, dan bucket menyimpan objek tersebut. Tidak ada hierarki sub-bucket atau sub-folder. Namun, Anda dapat menyimpulkan hierarki logis dengan menggunakan prefiks dan pembatas nama kunci, seperti yang dilakukan oleh konsol Amazon S3. Konsol Amazon S3 menggunakan konsep folder. Untuk informasi lebih lanjut tentang cara mengedit metadata dari konsol Amazon S3, lihat Mengedit metadata objek di konsol Amazon S3.

Misalnya, bucket Anda (`admin-created`) memiliki empat objek dengan kunci objek berikut:

`Development/Projects.xls`

`Finance/statement1.pdf`

`Private/taxdocument.pdf`

`s3-dg.pdf`

Konsol menggunakan prefiks nama kunci (`Development/`,`Finance/`, and`Private/`) dan delimiter (`/`) untuk menyajikan struktur folder. `s3-dg.pdf`Kunci tidak berisi awalan yang dibatasi garis miring, sehingga objeknya muncul langsung di tingkat root bucket. Jika Anda membuka folder `Development/`, Anda melihat objek `Projects.xlsx` di dalamnya.

- Amazon S3 mendukung bucket dan objek, dan tidak memiliki hierarki. Namun, dengan menggunakan awalan dan pembatas dalam nama kunci objek, konsol Amazon S3 dan AWS SDKs dapat menyimpulkan hierarki dan memperkenalkan konsep folder.
- Konsol Amazon S3 menerapkan pembuatan folder dengan menciptakan objek nol-byte dengan nilai prefiks dan pembatas folder sebagai kuncinya. Objek folder ini tidak akan muncul dalam konsol. Jika tidak, mereka berperilaku seperti objek lain dan dapat dilihat dan dimanipulasi melalui REST API, AWS CLI, dan. AWS SDKs

Panduan penamaan kunci objek

Anda dapat menggunakan karakter UTF-8 apa pun dalam nama kunci objek. Namun, penggunaan karakter tertentu dalam nama kunci dapat menimbulkan masalah pada beberapa aplikasi dan protokol. Panduan berikut membantu Anda memaksimalkan kepatuhan terhadap DNS, karakter web-safe, parser XHTML, dan lainnya. APIs

Karakter aman

Kumpulan karakter berikut umumnya aman untuk digunakan dalam nama kunci:

Karakter alfanumerik	• 0-9 • a-z • A-Z
Karakter-karakter khusus	• Tanda seru (!) • Tanda hubung (-) • Garis bawah (_) • Titik (.) • Tanda bintang (*) • Tanda kutip tunggal () ' • Tanda kurung pembuka () (• Tanda kurung penutup ())

Berikut ini adalah contoh nama kunci objek yang valid:

- `4my-organization`
- `my.great_photos-2014/jan/myvacation.jpg`
- `videos/2014/birthday/video1.wmv`

> **Note**
>
> Jika Anda menggunakan konsol Amazon S3 untuk mengunduh objek yang memiliki nama kunci yang diakhiri dengan periode (.), periode akan dihapus dari ujung nama kunci objek

yang diunduh. Untuk mempertahankan periode di akhir nama kunci dalam objek yang diunduh, Anda harus menggunakan AWS Command Line Interface (AWS CLI), AWS SDKs, atau Amazon S3 REST API.
Selain itu, waspadai batasan prefiks berikut ini:

- Objek dengan awalan `./` harus diunggah atau diunduh dengan AWS CLI, AWS SDKs, atau REST API. Anda tidak dapat menggunakan konsol Amazon S3 untuk mengunggah objek ini.
- Kunci objek yang berisi elemen jalur relatif (misalnya,`../`) valid jika, ketika diuraikan left-to-right, jumlah kumulatif segmen jalur relatif tidak pernah melebihi jumlah elemen jalur non-relatif yang ditemui. Aturan ini berlaku untuk semua permintaan yang dibuat menggunakan konsol Amazon S3, Amazon S3 REST API AWS CLI, dan. AWS SDKs

  Misalnya:

  - `videos/2014/../../video1.wmv`valid.
  - `videos/../../video1.wmv`tidak valid.
  - `videos/../../2014/video1.wmv`tidak valid.

Karakter yang memerlukan penanganan khusus

Karakter berikut dalam nama kunci mungkin memerlukan penanganan kode tambahan dan kemungkinan besar harus dikodekan URL atau direferensikan sebagai HEX. Beberapa karakter ini adalah karakter yang tidak dapat dicetak yang mungkin tidak ditangani browser Anda, yang juga memerlukan penanganan khusus:

- Ampersand () &
- Tanda dolar ($)
- Karakter ASCII bervariasi dengan rentang hex 00–1F (desimal 0–31) dan 7F (desimal 127)
- Pada simbol (@)
- Tanda sama (=)
- Titik koma () `;`
- Garis miring (/)
- Usus besar (`:`)
- Tanda plus (+)

- Ruang — Urutan ruang yang signifikan mungkin hilang dalam beberapa kasus (terutama beberapa ruang)
- Koma () ,
- Tanda tanya (?)

Karakter-karakter yang harus dihindari

Kami menyarankan untuk tidak menggunakan karakter berikut dalam nama kunci karena penanganan karakter khusus yang signifikan, yang tidak konsisten di semua aplikasi:

- Garis miring terbalik () \
- Penjepit kiri () {
- Karakter ASCII yang tidak dapat dicetak (karakter desimal 128–255)
- Karet atau sirkumflex () ^
- Penjepit kanan () }
- Persen karakter (%)
- Aksen kubur atau backtick () `
- Braket kanan ()]
- Tanda kutip () "
- Lebih besar dari tanda (>)
- Braket kiri () [
- Tilde () ~
- Kurang dari tanda (<)
- Tanda pound (#)
- Batang vertikal atau pipa (|)

Kendala kunci objek terkait XML

Seperti yang ditentukan oleh standar XHTML pada end-of-line penanganan, semua teks XML-dinormalisasi sedemikian rupa sehingga single carriage return (ASCII code 13) dan carriage return segera diikuti oleh line feed (kode ASCII 10), juga dikenal sebagai karakter baris baru, digantikan oleh karakter feed baris tunggal. Untuk memastikan penguraian kunci objek yang benar dalam

permintaan XHTML, pengembalian carriage dan karakter khusus lainnya harus diganti dengan kode entitas XMLnya yang setara saat dimasukkan ke dalam tag XMLnya.

Berikut ini adalah daftar karakter khusus tersebut dan kode entitas XMLnya yang setara:

- Apostrof (') harus diganti dengan `'`
- Tanda kutip (") harus diganti dengan `"`
- Ampersand (&) harus diganti dengan `&`
- Kurang dari tanda (<) harus diganti dengan `<`
- Lebih besar dari tanda (>>) harus diganti dengan `>`
- Carriage return (`\r`) harus diganti dengan `
` atau `
`
- Newline (`\n`) harus diganti dengan `
` atau `
`

Example

Contoh berikut ini mengilustrasikan penggunaan kode entitas XML sebagai pengganti untuk pengangkutan kembali. `DeleteObjects`Permintaan ini menghapus objek dengan `key` parameter `/some/prefix/objectwith\rcarriagereturn` (di mana `\r` adalah carriage return).

```
<Delete xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Object>
    <Key>/some/prefix/objectwith&#13;carriagereturn</Key>
  </Object>
</Delete>
```

Bekerja dengan metadata objek

Ada dua jenis metadata objek di Amazon S3: metadata yang ditentukan sistem dan metadata yang ditentukan pengguna. Metadata yang ditentukan sistem mencakup metadata seperti tanggal pembuatan objek, ukuran, dan kelas penyimpanan. Metadata yang ditentukan pengguna adalah metadata yang dapat Anda pilih untuk disetel pada saat Anda mengunggah objek. Metadata yang ditentukan pengguna ini adalah sekumpulan pasangan nama-nilai. Untuk informasi selengkapnya, lihat the section called “Metadata objek yang ditentukan sistem” dan the section called “Metadata objek yang ditentukan pengguna”.

Saat membuat objek, Anda menentukan kunci objek (atau nama kunci), yang secara unik mengidentifikasi objek dalam bucket Amazon S3. Untuk informasi selengkapnya, lihat Menamai objek

Amazon S3. Anda juga dapat mengatur metadata yang ditentukan pengguna di Amazon S3 pada saat Anda mengunggah objek.

Setelah mengunggah objek, Anda tidak dapat mengubah metadata yang ditentukan pengguna ini. Satu-satunya cara untuk memodifikasi metadata ini adalah dengan membuat salinan objek dan mengatur metadata. Untuk informasi selengkapnya tentang mengedit metadata menggunakan konsol Amazon S3, lihat. Mengedit metadata objek di konsol Amazon S3

Kueri metadata Anda dan percepat penemuan data dengan S3 Metadata

Untuk menemukan, menyimpan, dan menanyakan metadata objek S3 dengan mudah, Anda dapat menggunakan Metadata S3. Dengan S3 Metadata, Anda dapat dengan cepat menyiapkan data untuk digunakan dalam analisis bisnis, pengambilan konten, kecerdasan buatan dan pelatihan model pembelajaran mesin (AI/ML), dan banyak lagi.

Metadata S3 mempercepat penemuan data dengan secara otomatis menangkap metadata untuk objek di bucket tujuan umum Anda dan menyimpannya dalam tabel hanya-baca dan terkelola penuh yang dapat Anda kueri. Apache Iceberg Tabel read-only ini disebut tabel metadata. Saat objek ditambahkan, diperbarui, dan dihapus dari bucket tujuan umum Anda, Metadata S3 secara otomatis menyegarkan tabel metadata yang sesuai untuk mencerminkan perubahan terbaru.

Secara default, Metadata S3 menyediakan metadata objek yang ditentukan sistem, seperti waktu pembuatan objek dan kelas penyimpanan, dan metadata khusus, seperti tag dan metadata yang ditentukan pengguna yang disertakan selama pengunggahan objek. Metadata S3 juga menyediakan metadata peristiwa, seperti ketika objek diperbarui atau dihapus, dan yang membuat permintaan. Akun AWS

Tabel metadata disimpan dalam bucket tabel S3, yang menyediakan penyimpanan yang dioptimalkan untuk data tabular. Untuk menanyakan metadata Anda, Anda dapat mengintegrasikan keranjang meja Anda dengan layanan AWS analitik, seperti Amazon Athena, Amazon Redshift, dan Amazon. QuickSight

Untuk informasi selengkapnya tentang Metadata S3, lihat. the section called “Mempercepat penemuan data”

Metadata objek yang ditentukan sistem

Untuk setiap objek yang disimpan dalam sebuah bucket, Amazon S3 akan menyimpan serangkaian metadata sistem. Amazon S3 memproses metadata sistem ini sesuai dengan kebutuhan. Misalnya,

Amazon S3 mempertahankan tanggal pembuatan objek dan metadata ukuran, menggunakan informasi ini sebagai bagian dari manajemen objek.

Terdapat dua kategori metadata sistem:

- Sistem dikendalikan — Metadata seperti tanggal pembuatan objek dikendalikan oleh sistem, yang berarti hanya Amazon S3 yang dapat mengubah nilai tanggal.
- Dikontrol pengguna–Metadata sistem lainnya, seperti kelas penyimpanan yang dikonfigurasi untuk objek dan apakah enkripsi di sisi server objek diaktifkan, adalah beberapa contoh metadata sistem yang nilai-nilainya Anda kendalikan. Jika bucket Anda dikonfigurasi sebagai situs web, terkadang Anda mungkin ingin mengarahkan permintaan halaman ke halaman lain atau URL eksternal. Dalam hal ini, halaman laman web adalah sebuah objek di dalam bucket Anda. Amazon S3 menyimpan nilai pengalihan halaman sebagai metadata sistem, yang dapat Anda kontrol.

  Saat membuat objek, Anda dapat mengonfigurasi nilai item metadata sistem ini, atau memperbarui nilai saat Anda membutuhkannya. Untuk informasi selengkapnya tentang kelas penyimpanan, lihat Memahami dan mengelola kelas penyimpanan Amazon S3.

  Amazon S3 menggunakan AWS KMS kunci untuk mengenkripsi objek Amazon S3 Anda. AWS KMS mengenkripsi hanya data objek. Checksum dan algoritma yang ditentukan disimpan sebagai bagian dari metadata objek. Jika enkripsi di sisi server diminta untuk objek, maka checksum disimpan dalam bentuk terenkripsi. Untuk informasi lebih lanjut tentang enkripsi di sisi server, lihat Melindungi data dengan enkripsi.

> **Note**
>
> Header permintaan PUT dibatasi hingga ukuran 8 KB. Dalam header permintaan PUT, metadata yang ditentukan sistem dibatasi ke dalam ukuran 2 KB. Ukuran metadata yang ditentukan sistem diukur dengan menjumlahkan jumlah byte dalam pengodean US-ASCII dari setiap kunci dan nilai.

Tabel berikut ini menyediakan daftar metadata yang ditentukan sistem, dan apakah Anda dapat memperbaruinya.

Nama	Deskripsi	Apakah pengguna dapat mengubah nilainya?
`Date`	Tanggal dan waktu saat ini.	Tidak
`Cache-Control`	Bidang header umum yang digunakan untuk menentukan kebijakan caching.	Ya
`Content-Disposition`	Informasi presentasi objek.	Ya
`Content-Length`	Ukuran objek dalam byte.	Tidak
`Content-Type`	Jenis objek.	Ya
`Last-Modified`	Tanggal pembuatan objek atau tanggal modifikasi terakhir, mana pun yang terbaru. Untuk unggahan multibagian, tanggal pembuatan objek adalah tanggal inisiasi unggahan multibagian.	Tidak
`ETag`	Tag entitas (ETag) yang mewakili versi tertentu dari suatu objek. Untuk objek yang tidak diunggah sebagai unggahan multibagian dan tidak dienkripsi atau dienkripsi oleh enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3), ini adalah intisari data. ETag MD5	Tidak
`x-amz-server-side-encryption`	Header yang menunjukkan apakah enkripsi sisi server diaktifkan untuk objek, dan apakah enkripsi tersebut menggunakan kunci AWS Key Management Service (AWS KMS) (SSE-KMS) atau menggunakan kunci enkripsi terkelola Amazon S3 (SSE-S3). Untuk informasi selengkapnya, lihat Melindungi data dengan enkripsi di sisi klien.	Ya
`x-amz-checksum-crc`	Header yang berisi checksum atau intisari objek. Biasanya, salah satu header ini akan diatur pada satu	Tidak

Nama	Deskripsi	Apakah pengguna dapat mengubah nilainya?
`64nvme` , `x-amz-checksum-crc32` , `x-amz-checksum-crc32c` , `x-amz-checksum-sha1` , `x-amz-checksum-sha256`	waktu, bergantung pada algoritma checksum yang Anda perintahkan untuk digunakan oleh Amazon S3. Untuk informasi selengkapnya tentang memilih algoritma checksum, lihat Memeriksa integritas objek di Amazon S3.	
`x-amz-checksum-type`	Tipe checksum, yang menentukan bagaimana checksum tingkat bagian digabungkan untuk membuat checksum tingkat objek untuk objek multipart.	Ya
`x-amz-version-id`	Versi objek tersebut. Saat Anda mengaktifkan Penentuan Versi pada bucket, Amazon S3 menetapkan ID versi ke objek yang ditambahkan ke bucket. Untuk informasi selengkapnya, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.	Tidak
`x-amz-delete-marker`	Penanda Boolean yang mengindikasikan apakah objek tersebut adalah penanda hapus. Penanda ini hanya digunakan dalam bucket yang mengaktifkan Penentuan Versi,	Tidak
`x-amz-storage-class`	Kelas penyimpanan yang digunakan untuk menyimpan objek tersebut. Untuk informasi selengkapnya, lihat Memahami dan mengelola kelas penyimpanan Amazon S3.	Ya

Nama	Deskripsi	Apakah pengguna dapat mengubah nilainya?
`x-amz-website-redirect-location`	Sebuah header yang mengalihkan permintaan objek terkait ke objek lain di bucket yang sama, atau ke URL eksternal. Untuk informasi selengkapnya, lihat (Opsional) Mengonfigurasi pengalihan halaman web.	Ya
`x-amz-server-side-encryption-aws-kms-key-id`	Header yang menunjukkan ID kunci KMS enkripsi AWS KMS simetris yang digunakan untuk mengenkripsi objek. Header ini hanya digunakan ketika header `x-amz-server-side-encryption` muncul, dan memiliki nilai `aws:kms`.	Ya
`x-amz-server-side-encryption-customer-algorithm`	Header yang menunjukkan apakah enkripsi di sisi server dengan kunci enkripsi yang disediakan pelanggan (SSE-C) telah diaktifkan. Untuk informasi selengkapnya, lihat Menggunakan enkripsi di sisi server dengan kunci yang disediakan pelanggan (SSE-C).	Ya
`x-amz-tagging`	Tag-set untuk objek tersebut. Kumpulan tag harus dikodekan sebagai parameter Kueri URL.	Ya

Metadata objek yang ditentukan pengguna

Saat mengunggah objek, Anda juga dapat menetapkan metadata pada objek tersebut. Anda memberikan informasi opsional ini sebagai pasangan nama-nilai (nilai-kunci) saat Anda mengirimkan permintaan `PUT` atau `POST` untuk membuat objek. Saat Anda mengunggah objek menggunakan API REST, nama metadata opsional yang ditentukan pengguna harus dimulai dengan `x-amz-meta-` untuk membedakannya dari header HTTP lainnya. Saat Anda mengambil objek menggunakan API REST, prefiks ini akan ditampilkan. Saat Anda mengunggah objek menggunakan SOAP API, prefiks tidak dibutuhkan. Saat Anda mengambil objek menggunakan SOAP API, prefiks akan dihapus, tanpa menghiraukan API apa yang Anda gunakan untuk mengunggah objek tersebut.

> **Note**
>
> SOAP APIs for tidak tersedia untuk pelanggan baru, dan mendekati End of Life (EOL) pada 31 Agustus 2025. Kami menyarankan Anda menggunakan REST API atau AWS SDKs.

Saat metadata diambil melalui API REST, Amazon S3 menggabungkan header yang memiliki nama yang sama (kasus pengabaian) ke dalam daftar yang dibatasi koma. Jika beberapa metadata berisi karakter yang tidak dapat dicetak, maka metadata tidak akan ditampilkan. Sebaliknya, header `x-amz-missing-meta` ditampilkan dengan nilai jumlah dari entri metadata yang tidak dapat dicetak. Tindakan `HeadObject` mengambil metadata dari sebuah objek tanpa mengembalikan objek itu sendiri. Operasi ini berguna jika Anda hanya tertarik pada metadata objek. Untuk menggunakan `HEAD`, Anda harus memiliki akses `READ` ke objek. Untuk informasi selengkapnya, lihat HeadObjectdi Referensi API Amazon Simple Storage Service.

Metadata yang ditentukan pengguna adalah sekumpulan pasangan nilai-kunci. Amazon S3 menyimpan kunci metadata yang ditentukan pengguna dalam huruf kecil.

Amazon S3 memungkinkan karakter Unicode arbitrer dalam nilai metadata Anda.

Untuk menghindari masalah yang terkait dengan penyajian nilai metadata ini, Anda harus menyesuaikan diri dengan menggunakan karakter US-ASCII saat menggunakan REST dan UTF-8 saat menggunakan SOAP atau unggahan berbasis browser melalui. `POST`

Saat menggunakan non-US-ASCII karakter dalam nilai metadata Anda, string Unicode yang disediakan diperiksa untuk karakter. non-US-ASCII Nilai header tersebut didekodekan karakternya sesuai RFC 2047 sebelum disimpan dan dikodekan sesuai RFC 2047 agar aman untuk email sebelum dikembalikan. Jika string hanya berisi karakter US-ASCII, string akan ditampilkan sebagaimana adanya.

Berikut adalah contohnya.

```
PUT /Key HTTP/1.1
Host: amzn-s3-demo-bucket.s3.amazonaws.com
x-amz-meta-nonascii: ÄMÄZÕÑ S3

HEAD /Key HTTP/1.1
Host: amzn-s3-demo-bucket.s3.amazonaws.com
```

```
x-amz-meta-nonascii: =?UTF-8?B?w4PChE3Dg8KEWsODwpXDg8KRIFMz?=

PUT /Key HTTP/1.1
Host: amzn-s3-demo-bucket.s3.amazonaws.com
x-amz-meta-ascii: AMAZONS3

HEAD /Key HTTP/1.1
Host: amzn-s3-demo-bucket.s3.amazonaws.com
x-amz-meta-ascii: AMAZONS3
```

> **Note**
>
> Header permintaan PUT dibatasi hingga ukuran 8 KB. Dalam header permintaan PUT, metadata yang ditentukan pengguna dibatasi hingga ke dalam ukuran 2 KB. Ukuran metadata yang ditentukan pengguna diukur dengan menjumlahkan jumlah byte dalam pengodean UTF-8 dari setiap kunci dan nilai.

Untuk informasi tentang mengubah metadata objek Anda setelah diunggah dengan membuat salinan objek, memodifikasinya, dan mengganti objek lama, atau membuat versi baru, lihat Mengedit metadata objek di konsol Amazon S3.

Mengedit metadata objek di konsol Amazon S3

Anda dapat menggunakan konsol Amazon S3 untuk mengedit metadata objek S3 yang ada dengan menggunakan tindakan Salin. Untuk mengedit metadata, Anda menyalin objek ke tujuan yang sama dan menentukan metadata baru yang ingin Anda terapkan, yang menggantikan metadata lama untuk objek tersebut. Beberapa metadata diatur oleh Amazon S3 saat Anda mengunggah objek. Misalnya, `Content-Length` dan `Last-Modified` merupakan bidang metadata objek yang ditentukan sistem yang tidak dapat dimodifikasi oleh pengguna.

Anda juga dapat mengatur metadata yang ditentukan pengguna saat mengunggah objek dan menggantinya saat kebutuhan Anda berubah. Misalnya, Anda mungkin memiliki serangkaian objek yang awalnya Anda simpan di dalam kelas penyimpanan `STANDARD`. Seiring waktu, Anda mungkin tidak lagi memerlukan data ini untuk menjadi sangat tersedia. Jadi, Anda dapat mengubah kelas penyimpanan `GLACIER` dengan mengganti nilai `x-amz-storage-class` kunci dari `STANDARD` ke`GLACIER`.

Note

Pertimbangkan hal berikut saat Anda mengganti metadata objek di Amazon S3:

- Anda harus menentukan metadata yang ada yang ingin Anda pertahankan, metadata yang ingin Anda tambahkan, dan metadata yang ingin Anda edit.
- Jika objek Anda kurang dari 5 GB, Anda dapat menggunakan tindakan Copy di konsol S3 untuk mengganti metadata objek. Jika objek Anda lebih besar dari 5 GB, Anda dapat mengganti metadata objek saat Anda menyalin objek dengan unggahan multipart dengan menggunakan or. AWS CLIAWS SDKs Untuk informasi selengkapnya, lihat Menyalin objek menggunakan unggahan multibagian.
- Untuk daftar izin tambahan yang diperlukan untuk mengganti metadata, lihat. the section called “Izin yang diperlukan untuk operasi API S3” Misalnya kebijakan yang memberikan izin ini, lihatthe section called “Contoh kebijakan berbasis identitas”.
- Tindakan ini membuat salinan objek dengan pengaturan yang diperbarui dan tanggal modifikasi terakhir. Jika Penentuan Versi S3 diaktifkan, versi baru objek akan dibuat, dan objek yang sudah ada menjadi versi yang lebih lama. Jika Pembuatan Versi S3 tidak diaktifkan, salinan baru objek menggantikan objek asli. Akun AWS Terkait dengan peran IAM yang mengubah properti juga menjadi pemilik objek baru atau (versi objek).
- Mengedit metadata menggantikan nilai untuk nama kunci yang ada.
- Objek yang dienkripsi dengan kunci enkripsi yang disediakan pelanggan (SSE-C) tidak dapat disalin dengan menggunakan konsol. Anda harus menggunakan AWS CLI, AWS SDK, atau Amazon S3 REST API.
- Saat menyalin objek menggunakan konsol Amazon S3, Anda mungkin menerima pesan kesalahan “Metadata yang disalin tidak dapat diverifikasi.” Konsol menggunakan header untuk mengambil dan mengatur metadata untuk objek Anda. Jika konfigurasi jaringan atau browser Anda mengubah permintaan jaringan Anda, perilaku ini dapat menyebabkan metadata yang tidak diinginkan (seperti `Cache-Control` header yang dimodifikasi) ditulis ke objek yang Anda salin. Amazon S3 tidak dapat memverifikasi metadata yang tidak diinginkan ini.

  Untuk mengatasi masalah ini, periksa konfigurasi jaringan dan browser Anda untuk memastikan tidak mengubah header, seperti`Cache-Control`. Untuk informasi selengkapnya, lihat Model Tanggung Jawab Bersama.

> **Warning**
>
> Saat mengganti metadata untuk folder, tunggu hingga tindakan Salin selesai sebelum menambahkan objek baru ke folder. Jika tidak, objek baru juga akan diedit.

Topik berikut menjelaskan cara mengganti metadata untuk objek dengan menggunakan tindakan Salin di konsol Amazon S3.

Mengganti metadata yang ditentukan sistem

Anda dapat mengganti beberapa metadata yang ditentukan sistem untuk objek S3. Untuk daftar metadata dan nilai yang ditentukan sistem yang dapat Anda modifikasi, lihat. Metadata objek yang ditentukan sistem

Untuk mengganti metadata objek yang ditentukan sistem

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum atau ember Direktori.
3. Dalam daftar ember, pilih nama bucket yang berisi objek yang ingin Anda ubah.
4. Pilih kotak centang untuk objek yang ingin Anda ubah.
5. Pada menu Tindakan, pilih Salin dari daftar opsi yang muncul.
6. Untuk menentukan jalur tujuan, pilih Browse S3, navigasikan ke tujuan yang sama dengan objek sumber, dan pilih kotak centang tujuan. Pilih Pilih tujuan di sudut kanan bawah.

   Atau, masukkan jalur tujuan.
7. Jika Anda tidak mengaktifkan versi bucket, Anda akan melihat peringatan yang merekomendasikan Anda mengaktifkan Bucket Versioning untuk membantu melindungi dari penimpaan atau penghapusan objek secara tidak sengaja. Jika Anda ingin menyimpan semua versi objek dalam bucket ini, pilih Aktifkan Penentuan Versi Bucket. Anda juga dapat melihat enkripsi default dan properti Object Lock di Detail tujuan.
8. Di bawah Pengaturan salinan tambahan, pilih Tentukan pengaturan untuk menentukan pengaturan untuk Metadata.
9. Gulir ke bagian Metadata, lalu pilih Ganti semua metadata.
10. Pilih Tambah Metadata.

11. Untuk metadata Jenis, pilih Ditentukan sistem.
12. Tentukan Kunci unik dan Nilai metadata.
13. Untuk mengedit metadata tambahan, pilih Tambahkan metadata. Anda juga dapat memilih Hapus untuk menghapus satu set type-key-values.
14. Pilih Salin. Amazon S3 menyimpan perubahan metadata Anda.

Mengganti metadata yang ditentukan pengguna

Anda dapat mengganti metadata objek yang ditentukan pengguna dengan menggabungkan awalan metadata`x-amz-meta-`, dan nama yang Anda pilih untuk membuat kunci kustom. Misalnya, jika Anda menambahkan nama kustom `alt-name`, kunci metadata adalah `x-amz-meta-alt-name`.

Metadata yang ditentukan pengguna dapat berukuran sebesar 2 KB. Untuk menghitung ukuran total metadata yang ditentukan pengguna, hitung jumlah byte dalam pengodean UTF-8 untuk setiap kunci dan nilai. Kunci dan nilainya harus sesuai dengan standar US-ASCII. Untuk informasi selengkapnya, lihat Metadata objek yang ditentukan pengguna.

Untuk mengganti metadata objek yang ditentukan pengguna

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi, pilih Bucket, lalu pilih tab Bucket tujuan umum atau Bucket direktori. Arahkan ke bucket Amazon S3 atau folder yang berisi objek yang ingin Anda ubah.
3. Pilih kotak centang untuk objek yang ingin Anda ubah.
4. Pada menu Tindakan, pilih Salin dari daftar opsi yang muncul.
5. Untuk menentukan jalur tujuan, pilih Browse S3, navigasikan ke tujuan yang sama dengan objek sumber, dan pilih kotak centang tujuan. Pilih Pilih tujuan.

   Atau, masukkan jalur tujuan.
6. Jika Anda tidak mengaktifkan versi bucket, Anda akan melihat peringatan yang merekomendasikan Anda mengaktifkan Bucket Versioning untuk membantu melindungi dari penimpaan atau penghapusan objek secara tidak sengaja. Jika Anda ingin menyimpan semua versi objek dalam bucket ini, pilih Aktifkan Penentuan Versi Bucket. Anda juga dapat melihat enkripsi default dan properti Object Lock di Detail tujuan.
7. Di bawah Pengaturan salinan tambahan, pilih Tentukan pengaturan untuk menentukan pengaturan untuk Metadata.

8. Gulir ke bagian Metadata, lalu pilih Ganti semua metadata.
9. Pilih Tambah Metadata.
10. Untuk metadata Jenis, pilih Ditetapkan pengguna.
11. Memasukkan Kunci kustom yang unik, yang mengikuti `x-amz-meta-`. Masukkan juga metadata Nilai.
12. Untuk menambahkan metadata tambahan, pilih Tambahkan metadata. Anda juga dapat memilih Hapus untuk menghapus satu set type-key-values.
13. Pilih Salin. Amazon S3 menyimpan perubahan metadata Anda.

Mempercepat penemuan data dengan Metadata S3

Amazon S3 Metadata mempercepat penemuan data dengan secara otomatis menangkap metadata untuk objek di bucket tujuan umum Anda dan menyimpannya dalam tabel hanya-baca dan terkelola penuh yang dapat Anda kueri. Apache Iceberg Tabel read-only ini disebut tabel metadata. Saat objek ditambahkan, diperbarui, dan dihapus dari bucket tujuan umum Anda, Metadata S3 secara otomatis menyegarkan tabel metadata yang sesuai untuk mencerminkan perubahan terbaru.

Secara default, Metadata S3 menyediakan tiga jenis metadata:

- Metadata yang ditentukan sistem, seperti waktu pembuatan objek dan kelas penyimpanan
- Metadata kustom, seperti tag dan metadata yang ditentukan pengguna yang disertakan selama pengunggahan objek
- Metadata peristiwa, seperti ketika objek diperbarui atau dihapus, dan Akun AWS yang membuat permintaan

Untuk detail tentang data apa yang disimpan dalam tabel metadata, lihat. the section called “Skema tabel metadata”

Dengan Metadata S3, Anda dapat dengan mudah menemukan, menyimpan, dan menanyakan metadata untuk objek S3 Anda, sehingga Anda dapat dengan cepat menyiapkan data untuk digunakan dalam analisis bisnis, pengambilan konten, kecerdasan buatan dan pelatihan model pembelajaran mesin (AI/ML), dan banyak lagi.

Tabel metadata disimpan dalam bucket tabel S3, yang menyediakan penyimpanan yang dioptimalkan untuk data tabular. Untuk menanyakan metadata Anda dengan mudah, Anda dapat mengintegrasikan keranjang tabel Anda dengan. AWS Glue Data Catalog Setelah bucket meja

terintegrasi AWS Glue Data Catalog, Anda dapat langsung menanyakan tabel metadata Anda dengan mesin kueri seperti Amazon Athena, Amazon EMR, Amazon Apache Spark Redshift, dan. Trino Anda juga dapat menanyakan tabel metadata Anda dengan aplikasi lain yang mendukung format. Apache Iceberg Untuk membuat dasbor dari tabel metadata Anda, gunakan Amazon. QuickSight

Untuk harga Metadata S3, lihat Harga Amazon S3.

Cara kerja tabel metadata

Tabel metadata dikelola oleh Amazon S3, dan tidak dapat dimodifikasi oleh prinsipal IAM di luar Amazon S3 itu sendiri. (Namun, Anda dapat menghapus tabel metadata Anda.) Akibatnya, tabel metadata bersifat hanya-baca, yang membantu memastikan bahwa tabel tersebut mencerminkan konten bucket Anda dengan benar.

Agar tabel Apache Iceberg metadata Anda berkinerja terbaik, Amazon S3 melakukan aktivitas pemeliharaan berkala pada tabel Anda, seperti pemadatan dan penghapusan file yang tidak direferensikan. Aktivitas pemeliharaan ini membantu meminimalkan biaya penyimpanan tabel metadata Anda dan mengoptimalkan kinerja kueri. Pemeliharaan tabel ini terjadi secara otomatis, tidak memerlukan keikutsertaan atau manajemen berkelanjutan oleh Anda. Namun, jika diperlukan, Anda dapat mengonfigurasi aktivitas pemeliharaan tabel ini. Untuk informasi lebih lanjut, lihat Perawatan bucket tabel.

> **Note**
>
> Metadata S3 dirancang untuk terus ditambahkan ke tabel metadata saat Anda membuat perubahan pada bucket tujuan umum Anda. Setiap pembaruan membuat snapshot —versi baru dari tabel metadata. Karena sifat read-only dari tabel metadata, Anda tidak dapat menghapus catatan dalam tabel metadata. Anda juga tidak dapat menggunakan kemampuan kedaluwarsa snapshot dari Tabel S3 untuk menghapus snapshot lama tabel metadata Anda. Untuk membantu meminimalkan biaya, Anda dapat menghapus konfigurasi tabel metadata dan tabel metadata secara berkala, lalu membuatnya kembali. Untuk informasi selengkapnya, lihat the section called “Menghapus konfigurasi tabel metadata” dan the section called “Menghapus tabel metadata”.

Untuk membuat dan menyimpan metadata objek dalam tabel metadata terkelola S3, Anda membuat konfigurasi tabel metadata untuk bucket tujuan umum Anda. Amazon S3 dirancang untuk terus

memperbarui tabel metadata untuk mencerminkan perubahan terbaru pada data Anda selama konfigurasi aktif di bucket.

Untuk membuat konfigurasi tabel metadata, Anda harus memastikan bahwa Anda memiliki izin AWS Identity and Access Management (IAM) yang diperlukan untuk membuat dan mengelola tabel metadata. Untuk informasi selengkapnya, lihat the section called “Izin untuk tabel metadata”. Anda juga harus membuat atau menentukan bucket tabel S3 untuk menyimpan tabel metadata Anda. Bucket meja ini harus sama Wilayah AWS dan sesuai dengan ember tujuan umum Anda. Untuk informasi selengkapnya tentang membuat bucket tabel, lihat Membuat bucket tabel.

> **Note**
>
> Metadata S3 tidak berlaku untuk objek apa pun yang sudah ada di bucket tujuan umum sebelum membuat konfigurasi tabel metadata. Dengan kata lain, Metadata S3 hanya menangkap metadata untuk peristiwa perubahan (seperti unggahan, pembaruan, dan penghapusan) yang terjadi setelah Anda membuat konfigurasi tabel metadata Anda.

Untuk memantau pembaruan konfigurasi tabel metadata Anda, Anda dapat menggunakannya. AWS CloudTrail Untuk informasi selengkapnya, lihat the section called “Tindakan tingkat ember Amazon S3 yang dilacak dengan logging CloudTrail ”.

Topik

- Batasan dan batasan tabel metadata
- Skema tabel Metadata S3
- Mengkonfigurasi tabel metadata
- Menanyakan tabel metadata

Batasan dan batasan tabel metadata

Sebelum membuat konfigurasi tabel metadata, perhatikan batasan dan batasan berikut:

- Metadata S3 saat ini hanya tersedia di Wilayah AS Timur (Virginia N.), AS Timur (Ohio), dan AS Barat (Oregon).
- S3 Metadata mendukung semua kelas penyimpanan, kecuali untuk yang berikut ini:
  - Kelas penyimpanan S3 Express One Zone

- Kelas penyimpanan S3 One Zone-Infrequent Access (S3 One Zone-IA; Z-IA) dalam bucket direktori di Local Zones

> **Note**
>
> Untuk kelas penyimpanan S3 Intelligent-Tiering, tingkat tertentu tidak ditampilkan dalam tabel metadata.

- Untuk membuat konfigurasi tabel metadata, Anda harus membuat atau menentukan bucket tabel S3 untuk menyimpan tabel metadata Anda. Ember meja ini harus sama Wilayah AWS dan Akun AWS sebagai ember tujuan umum Anda.
- Metadata S3 tidak didukung untuk bucket direktori atau bucket tabel. Anda dapat membuat konfigurasi tabel metadata hanya untuk bucket tujuan umum.
- Metadata S3 tidak berlaku untuk objek apa pun yang sudah ada di bucket tujuan umum sebelum membuat konfigurasi tabel metadata. Dengan kata lain, Metadata S3 hanya menangkap metadata untuk peristiwa perubahan (seperti unggahan, pembaruan, dan penghapusan) yang terjadi setelah Anda membuat konfigurasi tabel metadata Anda.
- Metadata S3 dirancang untuk terus ditambahkan ke tabel metadata saat Anda membuat perubahan pada bucket tujuan umum Anda. Setiap pembaruan membuat snapshot —versi baru dari tabel metadata. Karena sifat read-only dari tabel metadata, Anda tidak dapat menghapus catatan dalam tabel metadata. Anda juga tidak dapat menggunakan kemampuan kedaluwarsa snapshot dari Tabel S3 untuk menghapus snapshot lama tabel metadata Anda.

  Untuk membantu meminimalkan biaya, Anda dapat menghapus konfigurasi tabel metadata dan tabel metadata secara berkala, lalu membuatnya kembali. Untuk informasi selengkapnya, lihat the section called “Menghapus konfigurasi tabel metadata” dan the section called “Menghapus tabel metadata”.
- Saat membuat atau memperbarui kebijakan bucket atau tabel tabel, pastikan Anda tidak membatasi Amazon S3 untuk menulis ke keranjang meja atau tabel metadata Anda. Jika Amazon S3 tidak dapat menulis ke keranjang tabel atau tabel metadata, Anda harus membuat tabel metadata baru dengan menghapus konfigurasi tabel metadata dan tabel metadata, lalu membuat konfigurasi baru.
- Sebelum menghapus tabel metadata, Anda harus terlebih dahulu menghapus konfigurasi tabel metadata terkait pada bucket tujuan umum Anda.
- Anda dapat membuat konfigurasi tabel metadata hanya untuk seluruh bucket tujuan umum. Anda tidak dapat menerapkan konfigurasi tabel metadata di tingkat awalan.

- Anda tidak dapat menjeda dan melanjutkan pembaruan ke tabel metadata. Sebagai gantinya, Anda dapat menghentikan pembaruan tabel metadata dengan menghapus konfigurasi tabel metadata terkait. Untuk mulai menerima pembaruan lagi, Anda harus membuat konfigurasi tabel metadata baru, yang membuat tabel metadata baru.
- Tabel metadata tidak berisi semua metadata yang sama seperti yang tersedia melalui S3 Inventory atau melalui Amazon S3 REST API. Misalnya, informasi berikut tidak tersedia dalam tabel metadata:
  - Status kedaluwarsa atau transisi Siklus Hidup S3
  - Periode retensi Object Lock atau mode tata kelola
  - Informasi daftar kontrol akses objek (ACL)
  - Status replikasi
- Anda tidak dapat menyesuaikan partisi atau pengurutan untuk tabel metadata. Akibatnya, beberapa kueri mungkin memerlukan pemindaian tabel dan karena itu mungkin kurang efisien.

Skema tabel Metadata S3

Tabel metadata Amazon S3 berisi baris dan kolom. Setiap baris mewakili peristiwa mutasi yang telah membuat, memperbarui, atau menghapus objek di bucket tujuan umum Anda. Sebagian besar peristiwa ini adalah hasil dari berbagai tindakan pengguna, tetapi beberapa peristiwa ini adalah hasil dari tindakan yang diambil oleh Amazon S3 atas nama Anda, seperti kedaluwarsa Siklus Hidup S3 atau transisi kelas penyimpanan.

S3 Metadata adalah pipeline pemrosesan peristiwa yang dirancang untuk menjaga tabel metadata pada akhirnya konsisten dengan perubahan apa yang telah terjadi di bucket tujuan umum Anda. Ketahuilah bahwa pada saat Metadata S3 diberi tahu bahwa objek dibuat atau diperbarui, objek tersebut mungkin sudah ditimpa atau dihapus di bucket. Secara default, baris tabel dibuat untuk setiap operasi bucket S3. Namun, jika metadata objek apa pun dihapus atau ditimpa, atau objek tidak dapat lagi diambil, beberapa kolom mungkin menunjukkan `NULL` nilai untuk menunjukkan skema metadata yang hilang.

Berikut ini adalah contoh tabel metadata untuk bucket tujuan umum bernama `amzn-s3-demo-bucket:`

```
bucket               key                         sequence_number
                                                                  record_type
 record_timestamp          version_id   is_delete_marker   size   last_modified_date
   e_tag                          storage_class  is_multipart   encryption_status
```

```
 is_bucket_key_enabled   kms_key_arn
               checksum_algorithm   object_tags   user_metadata

    requester      source_ip_address   request_id
amzn-s3-demo-bucket Finance/statement1.pdf
 80e737d8b4d82f776afffffffffffffff006737d8b4d82f776a00000000000000000000000000000000000000000000
   CREATE        2024-11-15 23:26:44.899                 FALSE
 6223   11/15/2024 23:26     e131b86632dda753aac4018f72192b83    STANDARD
 FALSE          SSE-KMS             FALSE                   arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890df   SSECRC32             {}
            {count -> Asia, customs -> false, family -> true, location -> Mary, name ->
 football, user -> United States}                       111122223333   192.0.2.1
    CVK8FWYRW0M9JW65
amzn-s3-demo-bucket s3-dg.pdf
 80e737d8b4e39f1dbdfffffffffffffff006737d8b4e39f1dbd00000000000000000000000000000000000000000000
   CREATE        2024-11-15 23:26:44.942                 FALSE
 3554   11/15/2024 23:26     9bb49efc2d92c05558ddffbbde8636d5    STANDARD
 FALSE          DSSE-KMS            FALSE                   arn:aws:kms:us-
east-1:936810216292:key/0dcebce6-49fd-4cae-b2e2-5512ad281afd   SSESHA1              {}
           {}
                                                    111122223333   192.0.2.1
   CVKAQDRAZEG7KXAY
amzn-s3-demo-bucket Development/Projects.xls
 80e737d8b4ed9ac5c6fffffffffffffff006737d8b4ed9ac5c60000000000000000000000000000000000000000000
   CREATE        2024-11-15 23:26:44.966                 FALSE              7746
 11/15/2024 23:26     729a6863e47fb9955b31bfabce984908    STANDARD   FALSE
 SSE-S3              FALSE                   NULL
                                  SSECRC32             {}            {count -> Asia,
 customs -> Canada, family -> Billiards, filter -> true, location -> Europe, name ->
 Asia, user -> United States}   111122223333   192.0.2.1           CVK7Z6XQTQ90BSRV
```

Tabel metadata memiliki skema berikut:

Nama kolom	Diperlukan?	Jenis data	
`bucket`	Ya	String	Nama ember tujuan umum. Untuk informasi selengkap nya, lihat the section called “Peraturan penamaan”.

Nama kolom	Diperlukan?	Jenis data	
`key`	Ya	String	Nama kunci objek (atau kunci) yang secara unik mengidentifikasi objek dalam ember. Untuk informasi selengkap nya, lihat the section called “Penamaan objek”.
`sequence_number`	Ya	String	Nomor urut, yang merupakan ordinal yang disertakan dalam catatan untuk objek tertentu. Untuk memesan catatan dari ember dan kunci yang sama, Anda dapat mengurutk ansequence_ number . Untuk ember dan kunci tertentu, sequence_ number nilai leksikografis yang lebih besar menyiratk an bahwa catatan diperkenalkan ke ember baru-baru ini.

Nama kolom	Diperlukan?	Jenis data	
record_type	Ya	String	Jenis rekaman ini, salah satu CREATEUPDATE_METADATA , atauDELETE. CREATEcatatan menunjukkan bahwa objek baru (atau versi baru dari objek) ditulis ke ember. UPDATE_METADATA catatan menangkap perubahan metadata yang bisa berubah untuk objek yang ada, seperti kelas penyimpanan atau tag. DELETEcatatan menunjukkan bahwa objek ini (atau versi objek ini) telah dihapus. Saat pembuatan versi diaktifkan, DELETE catatan mewakili penanda hapus atau penghapus an permanen. Hapus penanda memiliki record_ty

Nama kolom	Diperlukan?	Jenis data	
			pe nilai DELETE dan is_delete _marker nilai. True Catatan penghapus an permanen memiliki nilai nol di semua kolom lain kecualibucket,,key,sequenc number , record_ty pe record_ti mestamp , danversion_i d . Untuk informasi selengkapnya, lihat the section called “Menghapus versi objek”.
record_ti mestamp	Ya	Timestamp NTZ (tidak ada zona waktu)	Stempel waktu yang terkait dengan catatan ini.

Nama kolom	Diperlukan?	Jenis data	
`version_id`	Tidak	String	ID versi objek. Saat Anda mengaktifkan Penentuan Versi di bucket, Amazon S3 akan menetapkan nomor versi untuk objek yang ditambahkan ke dalam bucket. Untuk informasi selengkapnya, lihat the section called "Mempertahankan beberapa versi objek". Objek yang disimpan di bucket sebelum Anda menyetel status pembuatan versi memiliki ID versi null.

Nama kolom	Diperlukan?	Jenis data	
`is_delete_marker`	Tidak	Boolean	Status penanda hapus objek. Jika objek adalah penanda hapus, nilai ini adalah`True`. Kalau tidak, itu`False`. Untuk informasi selengkap nya, lihat the section called “Bekerja dengan penanda hapus”. **Note** Baris yang ditambahk an untuk penanda hapus memiliki `record_ty pe` nilai`DELETE`, bukan`UPDATE_ME TADATA` . Jika penanda hapus dibuat sebagai hasil dari kedaluwarsa Siklus Hidup S3, nilainya adalah. `requester`

Nama kolom	Diperlukan?	Jenis data	
			`s3.amazonaws.com`
`size`	Tidak	Panjang	Ukuran objek dalam byte, tidak termasuk ukuran unggahan multibagian yang tidak lengkap atau metadata objek. Jika `is_delete_marker` ya`True`, ukurannya adalah`0`. Untuk informasi selengkapnya, lihat the section called "Metadata objek yang ditentukan sistem".
`last_modified_date`	Tidak	Timestamp NTZ (tidak ada zona waktu)	Tanggal pembuatan objek atau tanggal modifikasi terakhir, mana pun yang terbaru. Untuk unggahan multipart , tanggal pembuatan objek adalah tanggal saat unggahan multipart dimulai. Untuk informasi selengkapnya, lihat the section called "Metadata objek yang ditentukan sistem".

Nama kolom	Diperlukan?	Jenis data	
`e_tag`	Tidak	String	Entity tag (ETag), yang merupakan hash dari objek. ETag Merefleksikan perubahan hanya pada isi objek, bukan metadatanya. Itu ETag bisa menjadi MD5 intisari dari data objek. Apakah MD5 intisari ETag adalah tergantung pada bagaimana objek itu dibuat dan bagaimana itu dienkripsi. Untuk informasi selengkap nya, lihat Object di Referensi API Amazon S3.

Nama kolom	Diperlukan?	Jenis data	
`storage_class`	Tidak	String	Kelas penyimpanan yang digunakan untuk menyimpan objek. Salah satu`STANDARD`,`REDUCED_REDUNDANCY`,`STANDARD_IA`,`ONEZONE_IA`,`INTELLIGENT_TIERING`,`GLACIER`,`DEEP_ARCHIVE`, atau`GLACIER_IR`. Untuk informasi selengkapnya, lihat the section called “Memahami dan mengelola kelas penyimpanan”.
`is_multipart`	Tidak	Boolean	Jenis upload objek. Jika objek diunggah sebagai unggahan multipart, nilai ini adalah. `True` Kalau tidak, itu`False`. Untuk informasi selengkapnya, lihat the section called “Menggunakan unggahan multibagian”.

Nama kolom	Diperlukan?	Jenis data	
`encryption_status`	Tidak	String	Status enkripsi sisi server objek, tergantung pada jenis kunci enkripsi yang digunakan: enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3), enkripsi sisi server dengan kunci () (SSE-KMS), enkripsi sisi server dua lapis AWS Key Management Service dengan AWS KMS(DSSE-KMS), atau enkripsi sisi server dengan kunci yang disediakan pelanggan (SSE-C). AWS KMS keys Jika objek tidak dienkripsi, nilai ini adalah null. Nilai yang mungkin adalah`SSE-S3`,`SSE-KMS`,`DSSE-KMS`,`SSE-C`, atau nol. Untuk informasi selengkapnya, lihat the section called “Enkripsi data”.

Nama kolom	Diperlukan?	Jenis data	
`is_bucket_key_enabled`	Tidak	Boolean	Status pengaktifan Kunci Bucket S3 objek. Jika objek menggunakan S3 Bucket Key untuk SSE-KMS, nilai ini adalah. `True` Kalau tidak, itu`False`. Untuk informasi selengkap nya, lihat the section called “Mengonfigurasi Kunci Bucket S3 untuk sebuah objek”.

Nama kolom	Diperlukan?	Jenis data	
`kms_key_arn`	Tidak	String	Nama Sumber Daya Amazon (ARN) untuk kunci KMS yang dengannya objek dienkripsi, untuk baris di mana atau. `encryption_status` `SSE-KMS` `DSSE-KMS` Jika objek tidak dienkripsi dengan SSE-KMS atau DSSE-KMS, nilainya adalah null. Untuk informasi selengkapnya, lihat the section called "Kunci KMS disimpan di AWS KMS (SSE-KMS)" dan the section called "Enkripsi di sisi server dua lapis (DSSE-KMS)". **Note** Jika baris mewakili versi objek yang tidak lagi ada pada saat peristiwa hapus atau timpa diproses, `kms_key_a`

Nama kolom	Diperlukan?	Jenis data	
			`rn` berisi nilai null, bahkan jika nilai `encryption_status` kolom adalah `SSE-KMS` atau. `DSSE-KMS`
`checksum_algorithm`	Tidak	String	Algoritma yang digunakan untuk membuat checksum untuk objek, salah satu dari`CRC64NVME` ,,`CRC32`, `CRC32CSHA1`, atau`SHA256`. Jika tidak ada checksum hadir, nilai ini adalah null. Untuk informasi selengkapnya, lihat the section called "Menggunakan algoritma checksum yang didukung".

Nama kolom	Diperlukan?	Jenis data	
`object_tags`	Tidak	Peta <String, String>	Tag objek yang terkait dengan objek. Tag objek disimpan sebagai peta pasangan kunci-nilai. Jika objek tidak memiliki tag objek, peta kosong (`{}`) disimpan. Untuk informasi selengkap nya, lihat the section called “Mengkate gorikan objek dengan tag”

> **Note**
>
> Jika `record_type` nilainya `DELETE`, `object_tags` kolom berisi nilai nol. Jika `record_type` nilainya `CREATE` atau `UPDATE_METADATA`, baris yang mewakili versi objek yang tidak ada lagi pada saat

Nama kolom	Diperlukan?	Jenis data	
			peristiwa hapus atau timpa diproses akan berisi nilai null di `object_tags` kolom.

Nama kolom	Diperlukan?	Jenis data	
`user_metadata`	Tidak	Peta <String, String>	Metadata pengguna yang terkait dengan objek. Metadata pengguna disimpan sebagai peta pasangan nilai kunci. Jika objek tidak memiliki metadata pengguna, peta kosong ({}) disimpan. Untuk informasi selengkapnya, lihat the section called "Metadata objek yang ditentukan pengguna". **Note** Jika `record_type` nilainya `DELETE`, `user_metadata` kolom berisi nilai nol. Jika `record_type` nilainya `CREATE` atau `UPDATE_METADATA`, baris yang mewakili versi objek yang tidak ada lagi

Nama kolom	Diperlukan?	Jenis data	
			pada saat peristiwa hapus atau timpa diproses akan berisi nilai null di `user_metadata` kolom.
`requester`	Tidak	String	Akun AWS ID pemohon atau Layanan AWS kepala sekolah yang membuat permintaan.
`source_ip_address`	Tidak	String	Alamat IP sumber permintaan. Untuk catatan yang dihasilkan oleh permintaan pengguna, kolom ini berisi alamat IP sumber permintaan. Untuk tindakan yang diambil oleh Amazon S3 atau lainnya Layanan AWS atas nama pengguna, kolom ini berisi nilai nol.
`request_id`	Tidak	String	ID permintaan yang terkait dengan permintaan.

Mengkonfigurasi tabel metadata

Amazon S3 Metadata mempercepat penemuan data dengan secara otomatis menangkap metadata untuk objek di bucket tujuan umum Anda dan menyimpannya dalam tabel hanya-baca dan terkelola penuh yang dapat Anda kueri. Apache Iceberg Tabel read-only ini disebut tabel metadata. Saat objek ditambahkan, diperbarui, dan dihapus dari bucket tujuan umum Anda, Metadata S3 secara otomatis menyegarkan tabel metadata yang sesuai untuk mencerminkan perubahan terbaru.

Dengan Metadata S3, Anda dapat dengan mudah menemukan, menyimpan, dan menanyakan metadata untuk objek S3 Anda, sehingga Anda dapat dengan cepat menyiapkan data untuk digunakan dalam analisis bisnis, kecerdasan buatan dan pelatihan model pembelajaran mesin (AI/ML), dan banyak lagi.

Untuk membuat dan menyimpan metadata objek dalam tabel metadata terkelola S3, Anda membuat konfigurasi tabel metadata untuk bucket tujuan umum Anda. Amazon S3 dirancang untuk terus memperbarui tabel metadata untuk mencerminkan perubahan terbaru pada data Anda selama konfigurasi aktif di bucket. Selain itu, Amazon S3 terus mengoptimalkan tabel metadata Anda untuk membantu mengurangi biaya penyimpanan dan meningkatkan kinerja kueri analitik.

Untuk membuat konfigurasi tabel metadata, pastikan Anda memiliki izin AWS Identity and Access Management (IAM) yang diperlukan untuk membuat dan mengelola tabel metadata. Anda juga harus membuat atau menentukan bucket tabel S3 untuk menyimpan tabel metadata Anda. Ember meja ini harus sama Wilayah AWS dan Akun AWS sebagai ember tujuan umum Anda.

Untuk memantau pembaruan konfigurasi tabel metadata Anda, Anda dapat menggunakannya. AWS CloudTrail Untuk informasi selengkapnya, lihat the section called “Tindakan tingkat ember Amazon S3 yang dilacak dengan logging CloudTrail ”.

Topik

- Menyiapkan izin untuk mengonfigurasi tabel metadata
- Membuat konfigurasi tabel metadata
- Mengontrol akses ke tabel metadata
- Menghapus konfigurasi tabel metadata
- Menghapus tabel metadata

Menyiapkan izin untuk mengonfigurasi tabel metadata

Untuk membuat konfigurasi tabel metadata, Anda harus memiliki izin AWS Identity and Access Management (IAM) yang diperlukan untuk membuat dan mengelola konfigurasi tabel metadata Anda dan untuk membuat dan mengelola tabel metadata dan keranjang tabel tempat tabel metadata Anda disimpan.

Untuk membuat dan mengelola konfigurasi tabel metadata Anda, Anda harus memiliki izin berikut:

- `s3:CreateBucketMetadataTableConfiguration`— Izin ini memungkinkan Anda membuat konfigurasi tabel metadata untuk bucket tujuan umum Anda.
- `s3:GetBucketMetadataTableConfiguration`— Izin ini memungkinkan Anda untuk mengambil informasi tentang konfigurasi tabel metadata Anda.
- `s3:DeleteBucketMetadataTableConfiguration`— Izin ini memungkinkan Anda untuk menghapus konfigurasi tabel metadata Anda.

Untuk membuat dan bekerja dengan tabel dan bucket tabel, Anda harus memiliki `s3tables` izin tertentu. Minimal, untuk membuat konfigurasi tabel metadata, Anda harus memiliki izin berikut: `s3tables`

- `s3tables:CreateNamespace`— Izin ini memungkinkan Anda untuk membuat namespace di keranjang meja. Tabel metadata menggunakan namespace default`aws_s3_metadata`.
- `s3tables:GetTable`— Izin ini memungkinkan Anda untuk mengambil informasi tentang tabel metadata Anda.
- `s3tables:CreateTable`— Izin ini memungkinkan Anda untuk membuat tabel metadata Anda.
- `s3tables:PutTablePolicy`— Izin ini memungkinkan Anda untuk menambahkan atau memperbarui kebijakan tabel metadata Anda.

Untuk informasi mendetail tentang semua izin bucket tabel dan tabel, lihat Manajemen akses untuk Tabel S3.

> **Important**
>
> Jika Anda juga ingin mengintegrasikan bucket tabel Anda dengan layanan AWS analitik sehingga Anda dapat menanyakan tabel metadata Anda, Anda memerlukan izin tambahan.

Untuk informasi selengkapnya, lihat Mengintegrasikan Tabel Amazon S3 AWS dengan layanan analitik.

Izin untuk SSE-KMS

Untuk mengakses bucket tabel S3 atau tabel S3 yang menggunakan enkripsi sisi server dengan kunci AWS Key Management Service (AWS KMS) (SSE-KMS), Anda perlu menyertakan izin tambahan.

1. Peran pengguna atau IAM memerlukan izin berikut. Anda dapat memberikan izin ini dengan menggunakan konsol IAM -. https://console.aws.amazon.com/iam/
   a. `s3tables:PutTableEncryption`untuk mengkonfigurasi enkripsi tabel
   b. `s3tables:PutTableBucketEncryption`untuk mengkonfigurasi enkripsi bucket tabel
   c. `kms:DescribeKey`pada AWS KMS kunci yang digunakan
2. Pada kebijakan sumber daya untuk kunci KMS, Anda memerlukan izin berikut. Anda dapat memberikan izin ini dengan menggunakan konsol KMS - https://console.aws.amazon.com /kms.
   a. `kms:GenerateDataKey`izin untuk `metadata.s3.amazonaws.com` dan `maintenance.s3tables.amazonaws.com`
   b. `kms:Decrypt`izin untuk `metadata.s3.amazonaws.com` dan `maintenance.s3tables.amazonaws.com`
   c. `kms:DescribeKey`izin untuk prinsipal pemanggilan AWS

Untuk informasi selengkapnya tentang pemberian izin yang diperlukan ke layanan Metadata S3, lihat dokumentasi tentang Pemberian izin utama layanan Metadata S3 untuk menggunakan kunci KMS Anda.

Contoh kebijakan

Untuk membuat dan bekerja dengan tabel metadata dan bucket tabel, Anda dapat menggunakan contoh kebijakan berikut. Dalam kebijakan ini, bucket tujuan umum tempat Anda menerapkan konfigurasi tabel metadata disebut sebagai. *`amzn-s3-demo-source-bucket`* Bucket meja tempat Anda menyimpan tabel metadata Anda disebut sebagai. *`amzn-s3-demo-bucket`* Untuk menggunakan kebijakan ini, ganti nama bucket ini dan *`user input placeholders`* dengan informasi Anda sendiri:

```
{
```

```
    "Version":"2012-10-17",
    "Statement":[
       {
          "Sid":"PermissionsToWorkWithMetadataTables",
          "Effect":"Allow",
          "Action":[
             "s3:CreateBucketMetadataTableConfiguration",
             "s3:GetBucketMetadataTableConfiguration",
             "s3:DeleteBucketMetadataTableConfiguration",
             "s3tables:*"
          ],
          "Resource":[
             "arn:aws:s3:::bucket/amzn-s3-demo-source-bucket",
             "arn:aws:s3tables:region:account_id:bucket/amzn-s3-demo-bucket",
             "arn:aws:s3tables:region:account_id:bucket/amzn-s3-demo-bucket/table/*"
          ]
       }
    ]
}
```

Membuat konfigurasi tabel metadata

Untuk membuat dan menyimpan Metadata Amazon S3 dalam tabel metadata yang dikelola sepenuhnya, Anda membuat Apache Iceberg konfigurasi tabel metadata untuk bucket tujuan umum Anda. Amazon S3 dirancang untuk terus memperbarui tabel metadata untuk mencerminkan perubahan terbaru pada data Anda selama konfigurasi aktif di bucket. Selain itu, Amazon S3 terus mengoptimalkan tabel metadata Anda untuk membantu mengurangi biaya penyimpanan dan meningkatkan kinerja kueri analitik.

Tabel metadata memiliki format Amazon Resource Name (ARN) berikut:

`arn:aws:s3tables:region-code:account-id:bucket/table-bucket-name/table/metadata_table_name`

Tabel metadata Amazon S3 yang dikelola sepenuhnya disimpan di `aws_s3_metadata` namespace di bucket tabel Anda. Untuk informasi selengkapnya tentang ruang nama dalam bucket tabel, lihat Ruang nama tabel.

Anda dapat membuat konfigurasi tabel metadata menggunakan konsol Amazon S3,AWS CLI(), API AWS Command Line Interface REST Amazon S3, AWS SDKs atau Amazon S3.

Prasyarat

Untuk membuat konfigurasi tabel metadata, Anda harus terlebih dahulu melakukan hal berikut:

- Pastikan Anda memiliki izin AWS Identity and Access Management (IAM) yang diperlukan untuk membuat dan mengelola tabel metadata. Untuk informasi selengkapnya, lihat the section called “Izin untuk tabel metadata”.
- Buat bucket tabel S3 untuk menyimpan tabel metadata Anda. Ember meja ini harus sama Wilayah AWS dan Akun AWS sebagai ember tujuan umum Anda. Untuk informasi selengkapnya tentang membuat bucket tabel, lihat Membuat bucket tabel. Jika Anda menggunakan konsol Amazon S3 untuk membuat konfigurasi, Anda dapat melakukan langkah ini sebagai bagian dari proses itu.
- Integrasikan keranjang meja Anda AWS Glue Data Catalog sehingga Anda dapat langsung menanyakan tabel metadata Anda dengan mesin kueri seperti Amazon Athena, Amazon EMR, Amazon Redshift,,, dan aplikasi Apache Spark lain Apache Trino apa pun yang mendukung format tersebut. Apache Iceberg Untuk informasi selengkapnya, lihat the section called “Menanyakan tabel metadata dengan layanan analitik AWS ”.

Buat konfigurasi tabel metadata

Menggunakan konsol S3

Untuk membuat konfigurasi tabel metadata

Sebelum Anda membuat konfigurasi tabel metadata, pastikan bahwa Anda telah meninjau dan memenuhi prasyarat dan yang telah Anda ulas. the section called “Keterbatasan dan pembatasan”

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Pilih bucket tujuan umum yang Anda inginkan untuk membuat konfigurasi tabel metadata.

   > **Note**
   >
   > Pastikan ember tujuan umum ini adalah Wilayah AWS tempat ember meja tersedia. Bucket meja hanya tersedia di Wilayah AS Timur (Virginia N.), AS Timur (Ohio), dan AS Barat (Oregon).

4. Pada halaman detail ember, pilih tab Metadata.
5. Pada tab Metadata, pilih Buat konfigurasi metadata.

6. Pada halaman Buat konfigurasi metadata, di bawah bucket tabel Tujuan, tentukan keranjang tabel untuk menyimpan tabel metadata Anda. Ember meja harus sama Akun AWS dan Wilayah AWS sebagai ember tujuan umum Anda.

   Jika Anda belum memiliki ember meja, pilih Buat ember meja. Nama keranjang meja harus 3 hingga 63 karakter dan unik Akun AWS di dalam Wilayah AWS yang Anda pilih. Karakter yang valid adalah a-z, 0-9, dan tanda hubung (-). Untuk informasi selengkapnya tentang membuat bucket tabel, lihat Membuat bucket tabel.

   Saat Anda membuat ember meja, pastikan Anda mengintegrasikannya AWS Glue Data Catalog. Untuk informasi selengkapnya, lihat the section called “Menanyakan tabel metadata dengan layanan analitik AWS ”.
7. Untuk nama tabel Metadata, tentukan nama yang Anda inginkan untuk tabel Anda miliki. Nama tabel metadata harus antara 1 dan 255 karakter dan unik dalam `aws_s3_metadata` namespace di keranjang tabel Anda. Karakter yang valid adalah huruf kecil, angka, dan garis bawah (). _
8. Pilih Buat konfigurasi tabel metadata.

Jika konfigurasi tabel metadata Anda berhasil, ARN untuk tabel metadata Anda ditampilkan pada tab Metadata, bersama dengan bucket tabel dan nama tabel metadata yang ditentukan.

Untuk memantau pembaruan konfigurasi tabel metadata Anda, Anda dapat menggunakannya. AWS CloudTrail Untuk informasi selengkapnya, lihat the section called “Tindakan tingkat ember Amazon S3 yang dilacak dengan logging CloudTrail ”.

Menggunakan AWS CLI

Untuk menjalankan perintah berikut, Anda harus AWS CLI menginstal dan mengkonfigurasi. Jika Anda belum menginstal, lihat Menginstal atau memperbarui ke versi terbaru dari AWS CLI Panduan AWS Command Line Interface Pengguna. AWS CLI

Atau, Anda dapat menjalankan AWS CLI perintah dari konsol dengan menggunakan AWS CloudShell. AWS CloudShell adalah shell berbasis browser dan pra-otentikasi yang dapat Anda luncurkan langsung dari file. AWS Management Console Untuk informasi lebih lanjut, lihat Apa itu CloudShell? dan Memulai dengan AWS CloudShell di Panduan AWS CloudShell Pengguna.

Untuk membuat konfigurasi tabel metadata dengan menggunakan AWS CLI

Sebelum Anda membuat konfigurasi tabel metadata, pastikan bahwa Anda telah meninjau dan memenuhi prasyarat dan yang telah Anda ulas. the section called “Keterbatasan dan pembatasan”

Untuk menggunakan perintah contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri.

1. Jika Anda belum memiliki keranjang tabel, gunakan perintah berikut untuk membuat keranjang tabel untuk menyimpan tabel metadata Anda. Pastikan bucket tabel Wilayah AWS sama dengan bucket tujuan umum yang Anda inginkan untuk membuat konfigurasi tabel metadata.

```
aws s3tables create-table-bucket --name amzn-s3-demo-bucket --region us-east-2
```

2. Untuk memverifikasi bahwa bucket tabel Anda telah dibuat, gunakan perintah berikut:

```
aws s3tables list-table-buckets --region us-east-2
```

3. Buat file JSON yang berisi konfigurasi tabel metadata Anda, dan simpan (misalnya,). `metadata-config.json` Berikut ini adalah konfigurasi sampel.

   Nama keranjang meja harus 3 hingga 63 karakter dan unik dalam nama Anda Akun AWS Wilayah AWS yang telah Anda pilih. Karakter yang valid adalah a-z, 0-9, dan tanda hubung (-). Untuk informasi selengkapnya tentang membuat bucket tabel, lihat Membuat bucket tabel.

   Nama tabel metadata harus antara 1 dan 255 karakter dan unik dalam `aws_s3_metadata` namespace di keranjang tabel Anda. Karakter yang valid adalah huruf kecil, angka, dan garis bawah (). _

```
{
  "S3TablesDestination": {
    "TableBucketArn": "arn:aws:s3tables:us-east-2:111122223333:bucket/amzn-s3-demo-bucket",
    "TableName": "test_metadata_table"
  }
}
```

4. Gunakan perintah berikut untuk menerapkan konfigurasi tabel metadata ke bucket tujuan umum Anda (misalnya,*amzn-s3-demo-source-bucket*):

```
aws s3api create-bucket-metadata-table-configuration \
--bucket amzn-s3-demo-source-bucket \
--metadata-table-configuration  file://./metadata-config.json \
--region us-east-2
```

5. Untuk memverifikasi bahwa konfigurasi telah dibuat, gunakan perintah berikut:

```
aws s3api get-bucket-metadata-table-configuration \
--bucket amzn-s3-demo-source-bucket \
--region us-east-2
```

Untuk memantau pembaruan konfigurasi tabel metadata Anda, Anda dapat menggunakannya. AWS CloudTrail Untuk informasi selengkapnya, lihat the section called “Tindakan tingkat ember Amazon S3 yang dilacak dengan logging CloudTrail ”.

Penggunaan API REST

Anda dapat mengirim permintaan REST untuk membuat konfigurasi tabel metadata. Untuk informasi selengkapnya, lihat CreateBucketMetadataTableConfiguration di Referensi API Amazon S3.

Menggunakan AWS SDKs

Anda dapat menggunakan AWS SDKs untuk membuat konfigurasi tabel metadata di Amazon S3. Untuk selengkapnya, lihat daftar yang didukung SDKs di Referensi API Amazon S3.

Mengontrol akses ke tabel metadata

Untuk mengontrol akses ke tabel metadata Amazon S3, Anda dapat menggunakan kebijakan berbasis sumber daya AWS Identity and Access Management (IAM) yang dilampirkan ke keranjang tabel dan tabel metadata Anda. Dengan kata lain, Anda dapat mengontrol akses ke tabel metadata Anda di tingkat bucket tabel dan tingkat tabel.

Untuk informasi selengkapnya tentang mengontrol akses ke bucket dan tabel tabel, lihat Manajemen akses untuk Tabel S3.

> **Important**
>
> Pastikan Anda tidak membatasi Amazon S3 dari menulis ke keranjang meja atau tabel metadata Anda. Jika Amazon S3 tidak dapat menulis ke keranjang tabel atau tabel metadata, Anda harus membuat tabel metadata baru dengan menghapus konfigurasi tabel metadata lalu membuat konfigurasi baru.

Anda juga dapat mengontrol akses ke baris dan kolom dalam tabel metadata Anda. AWS Lake FormationUntuk informasi selengkapnya, lihat Mengelola izin Lake Formation dan Pemfilteran data serta keamanan tingkat sel di Lake Formation di Panduan Pengembang.AWS Lake Formation

Menghapus konfigurasi tabel metadata

Jika ingin berhenti memperbarui konfigurasi tabel metadata untuk bucket tujuan umum Amazon S3, Anda dapat menghapus konfigurasi tabel metadata yang dilampirkan ke bucket. Menghapus konfigurasi tabel metadata hanya menghapus konfigurasi. Bucket tabel dan tabel metadata Anda masih ada, bahkan jika Anda menghapus konfigurasi tabel metadata. Namun, tabel metadata tidak akan lagi diperbarui.

Untuk menghapus tabel metadata Anda, lihat. the section called “Hapus tabel metadata” Untuk menghapus bucket tabel, lihat Menghapus bucket tabel dan DeleteTableBucketdi Referensi API Amazon S3.

Anda dapat menghapus konfigurasi tabel metadata menggunakan konsol Amazon S3,AWS CLI(), API AWS Command Line Interface REST Amazon S3, AWS SDKs atau Amazon S3.

Hapus konfigurasi tabel metadata

Menggunakan konsol S3

Untuk menghapus konfigurasi tabel metadata

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Pilih bucket tujuan umum tempat Anda ingin menghapus konfigurasi tabel metadata.
4. Pada halaman detail ember, pilih tab Metadata.
5. Pada tab Metadata, pilih Hapus.
6. Di kotak dialog Hapus konfigurasi metadata, masukkan **`confirm`** untuk mengonfirmasi bahwa Anda ingin menghapus konfigurasi. Lalu pilih Hapus.

Menggunakan AWS CLI

Untuk menjalankan perintah berikut, Anda harus AWS CLI menginstal dan mengkonfigurasi. Jika Anda belum menginstal, lihat Menginstal atau memperbarui ke versi terbaru dari AWS CLI Panduan AWS Command Line Interface Pengguna. AWS CLI

Atau, Anda dapat menjalankan AWS CLI perintah dari konsol dengan menggunakan AWS CloudShell. AWS CloudShell adalah shell berbasis browser dan pra-otentikasi yang dapat Anda

luncurkan langsung dari file. AWS Management Console Untuk informasi lebih lanjut, lihat Apa itu CloudShell? dan Memulai dengan AWS CloudShell di Panduan AWS CloudShell Pengguna.

Untuk menghapus konfigurasi tabel metadata dengan menggunakan AWS CLI

Untuk menggunakan perintah contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri.

1. Gunakan perintah berikut untuk menghapus konfigurasi tabel metadata dari bucket tujuan umum Anda (misalnya,*amzn-s3-demo-source-bucket*):

```
aws s3api delete-bucket-metadata-table-configuration \
--bucket amzn-s3-demo-source-bucket \
--region us-east-2
```

2. Untuk memverifikasi bahwa konfigurasi telah dihapus, gunakan perintah berikut:

```
aws s3api get-bucket-metadata-table-configuration \
--bucket amzn-s3-demo-source-bucket \
--region us-east-2
```

Penggunaan API REST

Anda dapat mengirim permintaan REST untuk menghapus konfigurasi tabel metadata. Untuk informasi selengkapnya, lihat DeleteBucketMetadataTableConfiguration.

Menggunakan AWS SDKs

Anda dapat menggunakan AWS SDKs untuk menghapus konfigurasi tabel metadata di Amazon S3. Untuk selengkapnya, lihat daftar yang didukung SDKs.

Menghapus tabel metadata

Jika ingin menghapus tabel metadata yang Anda buat untuk bucket tujuan umum Amazon S3, Anda dapat menghapus tabel metadata dari keranjang tabel.

> **Note**
>
> Sebelum menghapus tabel metadata, sebaiknya hapus terlebih dahulu konfigurasi tabel metadata terkait pada bucket tujuan umum. Untuk informasi selengkapnya, lihat the section called “Menghapus konfigurasi tabel metadata”.

Untuk menghapus bucket tabel, lihat Menghapus bucket tabel dan DeleteTableBucketdi Referensi API Amazon S3.

Anda dapat menghapus tabel metadata menggunakan AWS Command Line Interface (AWS CLI), the AWS SDKs, atau Amazon S3 REST API.

Hapus tabel metadata

Menggunakan AWS CLI

Untuk menjalankan perintah berikut, Anda harus AWS CLI menginstal dan mengkonfigurasi. Jika Anda belum menginstal, lihat Menginstal atau memperbarui ke versi terbaru dari AWS CLI Panduan AWS Command Line Interface Pengguna. AWS CLI

Atau, Anda dapat menjalankan AWS CLI perintah dari konsol dengan menggunakan AWS CloudShell. AWS CloudShell adalah shell berbasis browser dan pra-otentikasi yang dapat Anda luncurkan langsung dari file. AWS Management Console Untuk informasi lebih lanjut, lihat Apa itu CloudShell? dan Memulai dengan AWS CloudShell di Panduan AWS CloudShell Pengguna.

Untuk menghapus konfigurasi tabel metadata dengan menggunakan AWS CLI

Untuk menggunakan perintah contoh berikut, ganti *`user input placeholders`* dengan informasi Anda sendiri.

1. Gunakan perintah berikut untuk menghapus tabel metadata dari keranjang tabel Anda (misalnya,*`amzn-s3-demo-bucket`*):

```
aws s3tables delete-table \
--table-bucket-arn arn:aws:s3tables:us-east-2:111122223333:bucket/amzn-s3-demo-bucket \
--namespace aws_s3_metadata \
--name test_metadata_table \
--region us-east-2
```

2. Untuk memverifikasi bahwa tabel telah dihapus, gunakan perintah berikut:

```
aws s3tables get-table \
--table-bucket-arn arn:aws:s3tables:us-east-2:111122223333:bucket/amzn-s3-demo-bucket \
--namespace aws_s3_metadata \
--name test_metadata_table \
--region us-east-2
```

Penggunaan API REST

Anda dapat mengirim permintaan REST untuk menghapus konfigurasi tabel metadata. Untuk informasi selengkapnya, lihat DeleteTable di Referensi API Amazon S3.

Menggunakan AWS SDKs

Anda dapat menggunakan AWS SDKs untuk menghapus konfigurasi tabel metadata di Amazon S3. Untuk selengkapnya, lihat daftar yang didukung SDKs di Referensi API Amazon S3.

Menanyakan tabel metadata

Amazon S3 Metadata memungkinkan Anda menganalisis tabel metadata terkelola S3 Anda dengan mesin kueri apa pun yang mendukung format Apache Iceberg. Saat mengintegrasikan bucket tabel S3 dengan layanan AWS analitik, Anda dapat menggunakan layanan seperti Amazon Athena, Amazon Redshift, dan lainnya untuk membantu Anda melakukan hal berikut dengan tabel metadata Anda:

- Temukan pola dan tren penggunaan penyimpanan
- Audit AWS Key Management Service (AWS KMS) penggunaan kunci enkripsi di seluruh objek Anda
- Cari objek berdasarkan metadata yang ditentukan pengguna dan tag objek
- Memahami perubahan metadata objek dari waktu ke waktu
- Pelajari kapan objek diperbarui atau dihapus, termasuk Akun AWS ID atau alamat IP yang membuat permintaan

Setelah bucket tabel terintegrasi dengan layanan AWS analitik, Anda dapat menanyakan tabel metadata Anda. Ini termasuk menggabungkan tabel metadata terkelola S3 dan tabel metadata

kustom, memungkinkan Anda melakukan kueri di beberapa kumpulan data selama tabel metadata disimpan dalam keranjang tabel yang sama.

Dari sana, Anda dapat membuat dasbor interaktif dengan data kueri Anda dengan menggunakan Amazon QuickSight.

Pertimbangan harga kueri

Harga tambahan berlaku untuk menjalankan kueri pada tabel metadata Anda. Untuk informasi selengkapnya, lihat informasi harga untuk mesin kueri yang Anda gunakan.

Untuk informasi tentang membuat kueri Anda lebih hemat biaya, lihatMengoptimalkan kinerja kueri tabel metadata.

Topik

- Menanyakan tabel metadata dengan layanan analitik AWS
- Kueri tabel metadata dengan mesin kueri sumber terbuka
- Mengoptimalkan kinerja kueri tabel metadata
- Contoh kueri tabel metadata
- Menggabungkan metadata kustom dengan tabel metadata S3
- Memvisualisasikan data tabel metadata dengan Amazon QuickSight

Menanyakan tabel metadata dengan layanan analitik AWS

Anda dapat menanyakan tabel metadata terkelola S3 Anda dengan layanan AWS analitik seperti Amazon Athena, Amazon Redshift, dan Amazon EMR.

Sebelum Anda dapat menjalankan kueri, Anda harus terlebih dahulu mengintegrasikan bucket tabel S3 di wilayah Anda Akun AWS dan wilayah dengan AWS layanan analitik.

Menanyakan tabel metadata dengan Amazon Athena

Setelah mengintegrasikan bucket tabel S3 dengan layanan AWS analitik, Anda dapat mulai menanyakan tabel metadata Anda di Athena. Dalam kueri Anda, tentukan katalog Anda sebagai `s3tablescatalog` dan database Anda sebagai `aws_s3_metadata` (yang merupakan namespace untuk tabel metadata Anda). Untuk informasi selengkapnya, lihat Menanyakan tabel Amazon S3 dengan Athena.

Menanyakan tabel metadata dengan Amazon Redshift

Setelah mengintegrasikan bucket tabel S3 dengan layanan AWS analitik, Anda membuat tautan sumber daya ke namespace tabel metadata (). `aws_s3_metadata` Setelah selesai, Anda dapat mulai menanyakan tabel metadata Anda di konsol Amazon Redshift. Untuk informasi selengkapnya, lihat Mengakses tabel Amazon S3 dengan Amazon Redshift.

Menanyakan tabel metadata dengan Amazon EMR

Untuk menanyakan tabel metadata Anda menggunakan Amazon EMR, Anda membuat klaster EMR Amazon yang dikonfigurasi untuk Apache Iceberg dan terhubung ke tabel metadata Anda menggunakan Apache Spark. Anda dapat mengaturnya dengan mengintegrasikan bucket tabel S3 Anda dengan layanan AWS analitik atau menggunakan katalog klien Amazon S3 Tables Catalog for Iceberg sumber terbuka.

Untuk informasi selengkapnya, lihat Mengakses tabel Amazon S3 dengan Amazon EMR.

Kueri tabel metadata dengan mesin kueri sumber terbuka

Anda dapat menanyakan tabel metadata terkelola S3 Anda dengan menggunakan mesin kueri sumber terbuka, seperti Apache Spark. Untuk menanyakan tabel metadata Anda, Anda memerlukan Katalog Tabel Amazon S3 untuk katalog klien Apache Iceberg (pustaka sumber terbuka yang dihosting oleh Labs). AWS

Untuk informasi selengkapnya, lihat Menanyakan tabel Amazon S3 dengan Apache Spark.

Mengoptimalkan kinerja kueri tabel metadata

Karena Metadata S3 didasarkan pada format tabel Apache Iceberg, Anda dapat mengoptimalkan kinerja dan biaya kueri tabel metadata Anda dengan menggunakan rentang waktu tertentu.

Misalnya, kueri SQL berikut memberikan tingkat sensitivitas objek baru dalam bucket tujuan umum S3:

```
SELECT key, object_tags['SensitivityLevel']
FROM aws_s3_metadata.my_metadata_table
WHERE record_type = 'CREATE'
GROUP BY object_tags['SensitivityLevel']
```

Kueri ini memindai seluruh tabel metadata, yang mungkin membutuhkan waktu lama untuk dijalankan. Untuk meningkatkan kinerja, Anda dapat menyertakan `record_timestamp` kolom untuk

fokus pada rentang waktu tertentu. Berikut adalah versi terbaru dari kueri sebelumnya yang melihat objek baru dari bulan lalu:

```
SELECT key, object_tags['SensitivityLevel']
FROM aws_s3_metadata.my_metadata_table
WHERE record_type = 'CREATE'
AND record_timestamp > (CURRENT_TIMESTAMP - interval '1' month)
GROUP BY object_tags['SensitivityLevel']
```

Contoh kueri tabel metadata

Contoh berikut menunjukkan bagaimana Anda bisa mendapatkan informasi jenis yang berbeda dari tabel Metadata S3 Anda dengan menggunakan kueri SQL standar.

Ingat saat menggunakan contoh-contoh ini:

- Contohnya ditulis untuk bekerja dengan Amazon Athena. Anda mungkin harus memodifikasi contoh untuk bekerja dengan mesin kueri yang berbeda.
- Pastikan Anda memahami cara mengoptimalkan kueri Anda.
- Ganti *amzn-s3-demo-bucket* dengan nama bucket tabel S3 yang menyimpan tabel metadata Anda.
- Ganti *my_metadata_table* dengan nama tabel metadata yang Anda kueri.
- Untuk daftar lengkap kolom yang didukung, lihatSkema tabel Metadata S3.

Menemukan objek dengan ekstensi file

Query berikut mengembalikan objek dengan ekstensi file tertentu (`.jpg`dalam hal ini).

```
SELECT key FROM "s3tablescatalog/amzn-s3-demo-
bucket"."aws_s3_metadata"."my_metadata_table"
WHERE key LIKE '%.jpg'
AND record_type = 'CREATE'
```

Daftar penghapusan objek

Kueri berikut mengembalikan peristiwa penghapusan objek, termasuk Akun AWS ID atau prinsip AWS layanan yang membuat permintaan.

```
SELECT DISTINCT bucket, key, sequence_number, record_type, record_timestamp, requester,
 source_ip_address, version_id
```

```
FROM "s3tablescatalog/amzn-s3-demo-bucket"."aws_s3_metadata"."my_metadata_table"
WHERE record_type = 'DELETE';
```

Daftar kunci AWS KMS enkripsi yang digunakan oleh objek Anda

Query berikut mengembalikan ARNs kunci AWS Key Management Service (AWS KMS) mengenkripsi objek Anda.

```
SELECT DISTINCT kms_key_arn
FROM "s3tablescatalog/amzn-s3-demo-bucket"."aws_s3_metadata"."my_metadata_table";
```

Daftar objek yang tidak menggunakan kunci KMS

Query berikut mengembalikan objek yang tidak dienkripsi dengan AWS KMS kunci.

```
SELECT DISTINCT kms_key_arn
FROM "s3tablescatalog/amzn-s3-demo-bucket"."aws_s3_metadata"."my_metadata_table"
WHERE encryption_status NOT IN ('SSE-KMS', 'DSSE-KMS')
AND record_type = 'CREATE';
```

Melihat metadata yang disediakan oleh Amazon Bedrock

Beberapa AWS layanan (seperti Amazon Bedrock), mengunggah objek ke Amazon S3. Anda dapat menanyakan metadata objek yang disediakan oleh layanan ini. Misalnya, kueri berikut menyertakan `user_metadata` kolom untuk menentukan apakah ada objek yang diunggah oleh Amazon Bedrock ke bucket tujuan umum.

```
SELECT DISTINCT bucket, key, sequence_number, record_type, record_timestamp,
 user_metadata
FROM "s3tablescatalog/amzn-s3-demo-bucket"."aws_s3_metadata"."my_metadata_table"
WHERE record_type = 'CREATE'
AND user_metadata['content-source'] = 'AmazonBedrock';
```

Jika Amazon Bedrock mengunggah objek ke bucket Anda, `user_metadata` kolom akan menampilkan metadata berikut yang terkait dengan objek dalam hasil kueri:

```
user_metadata
{content-additional-params -> requestid="CVK8FWYRW0M9JW65",
 signedContentSHA384="38b060a751ac96384cd9327eb1b1e36a21fdb71114be07434c0cc7bf63f6e1da274edebfe
 content-model-id -> bedrock-model-arn, content-source -> AmazonBedrock}
```

Memahami keadaan objek Anda saat ini

Kueri berikut dapat membantu Anda menentukan keadaan objek Anda saat ini. Kueri mengidentifikasi versi terbaru dari setiap objek, menyaring objek yang dihapus, dan menandai versi terbaru dari setiap objek berdasarkan nomor urut. Hasil diurutkan olehbucket,key, dan sequence_number kolom.

```
WITH records_of_interest as (
   -- Start with a query that can narrow down the records of interest.
    SELECT * from "s3tablescatalog/amzn-s3-demo-
bucket"."aws_s3_metadata"."my_metadata_table"
),

version_stacks as (
   SELECT *,
          -- Introduce a column called 'next_sequence_number', which is the next larger
          -- sequence_number for the same key version_id in sorted order.
          LEAD(sequence_number, 1) over (partition by (bucket, key,
 coalesce(version_id, '')) order by sequence_number ASC) as next_sequence_number
   from records_of_interest
),

-- Pick the 'tip' of each version stack triple: (bucket, key, version_id).
-- The tip of the version stack is the row of that triple with the largest sequencer.
-- Selecting only the tip filters out any row duplicates.
-- This isn't typical, but some events can be delivered more than once to the table
-- and include rows that might no longer exist in the bucket (since the
-- table contains rows for both extant and extinct objects).
-- In the next subquery, eliminate the rows that contain deleted objects.
current_versions as (
    SELECT * from version_stacks where next_sequence_number is NULL
),

-- Eliminate the rows that are extinct from the bucket by filtering with
-- record_type. An object version has been deleted from the bucket if its tip is
-- record_type==DELETE.
existing_current_versions as (
    SELECT * from current_versions where not (record_type = 'DELETE' and
 is_delete_marker = FALSE)
),

-- Optionally, to determine which of several object versions is the 'latest',
-- you can compare their sequence numbers. A version_id is the latest if its
-- tip's sequencer is the largest among all other tips in the same key.
```

```
with_is_latest as (
    SELECT *,
           -- Determine if the sequence_number of this row is the same as the largest
 sequencer for the key that still exists.
           sequence_number = (MAX(sequence_number) over (partition by (bucket, key)))
 as is_latest_version
    FROM existing_current_versions
)

SELECT * from with_is_latest
ORDER BY bucket, key, sequence_number;
```

Menggabungkan metadata kustom dengan tabel metadata S3

Anda dapat menganalisis data di seluruh tabel metadata terkelola S3 dan tabel metadata pelanggan (dikelola sendiri). Dengan menggunakan `JOIN` operator SQL standar, Anda dapat melakukan kueri data dari berbagai sumber ini.

Contoh kueri SQL berikut menemukan catatan yang cocok antara tabel metadata terkelola S3 (*`my_s3_metadata_table`*) dan tabel metadata yang dikelola sendiri (). *`my_self_managed_metadata_table`* Kueri juga memfilter informasi berdasarkan `CREATE` peristiwa, yang menunjukkan bahwa objek baru (atau versi baru objek) ditulis ke ember. (Untuk informasi lebih lanjut, lihatSkema tabel Metadata S3.)

```
SELECT *
FROM aws_s3_metadata.my_s3_metadata_table a
JOIN my_namespace.my_self_managed_metadata_table b
ON a.bucket = b.bucket AND a.key = b.key AND a.version_id = b.version_id
WHERE a.record_type = 'CREATE';
```

Memvisualisasikan data tabel metadata dengan Amazon QuickSight

Dengan Amazon QuickSight, Anda dapat membuat dasbor interaktif untuk menganalisis dan memvisualisasikan hasil kueri SQL tentang tabel metadata terkelola S3 Anda. QuickSight dasbor dapat membantu Anda memantau statistik, melacak perubahan, dan mendapatkan wawasan operasional tentang tabel metadata Anda.

Dasbor tentang tabel metadata Anda mungkin menunjukkan kepada Anda:

- Berapa banyak objek di kelas penyimpanan yang berbeda?
- Berapa persentase data penyimpanan Anda adalah objek kecil dibandingkan dengan objek besar?

- Jenis objek apa yang ada di ember saya?
- Berapa persentase unggahan objek dibandingkan dengan penghapusan?

Setelah mengintegrasikan bucket tabel S3 dengan layanan AWS analitik, Anda dapat membuat kumpulan data dari tabel metadata dan bekerja dengannya di Amazon QuickSight menggunakan SPICE atau kueri SQL langsung dari mesin kueri Anda. QuickSight mendukung Amazon Athena dan Amazon Redshift sebagai sumber data.

Untuk informasi selengkapnya, lihat Memvisualisasikan data tabel dengan Amazon QuickSight.

Mengunggah Objek

Saat Anda mengunggah file ke Amazon S3, file tersebut disimpan sebagai objek S3. Objek terdiri dari data file dan metadata yang menjelaskan objek tersebut. Anda dapat memiliki jumlah objek tak terbatas dalam satu bucket. Sebelum Anda dapat mengunggah file ke bucket Amazon S3, Anda memerlukan izin menulis untuk bucket tersebut. Untuk informasi selengkapnya tentang izin akses, lihat Identity and Access Management untuk Amazon S3.

Anda dapat mengunggah jenis file apa pun—gambar, cadangan, data, film, dan sebagainya—ke dalam bucket S3. Ukuran maksimum file yang dapat Anda unggah menggunakan konsol Amazon S3 adalah 160 GB. Untuk mengunggah file yang lebih besar dari 160 GB, gunakan AWS Command Line Interface (AWS CLI) AWS SDKs, atau Amazon S3 REST API.

Jika Anda mengunggah sebuah objek dengan nama kunci yang sudah ada dalam bucket yang diaktifkan dengan Penentuan Versi, Amazon S3 membuat versi lain dari objek tersebut dan bukan mengganti objek yang ada. Untuk informasi selengkapnya tentang mengaktifkan pembuatan versi, lihat. Mengaktifkan Penentuan Versi pada bucket

Tergantung dari ukuran data yang Anda unggah, Amazon S3 menawarkan opsi berikut:

- Unggah objek dalam satu operasi dengan menggunakan AWS SDKs, REST API, atau AWS CLI — Dengan satu PUT operasi, Anda dapat mengunggah satu objek berukuran hingga 5 GB.
- Unggah objek tunggal menggunakan konsol Amazon S3 –Dengan konsol Amazon S3, Anda dapat mengunggah objek tunggal yang berukuran hingga 160 GB.
- Unggah objek dalam beberapa bagian dengan menggunakan AWS SDKs, REST API, atau AWS CLI — Menggunakan operasi API unggahan multibagian, Anda dapat mengunggah satu objek besar, berukuran hingga 5 TB.

Pengoperasian API unggahan multibagian dirancang untuk meningkatkan pengalaman pengunggahan untuk objek yang lebih besar. Anda dapat mengunggah objek dalam beberapa bagian. Bagian-bagian objek ini dapat diunggah secara mandiri, dalam urutan apa pun, dan secara paralel. Anda dapat menggunakan unggahan multibagian untuk objek dengan ukuran mulai dari 5 MB hingga 5 TB. Untuk informasi selengkapnya, lihat Mengunggah dan menyalin objek menggunakan unggahan multipart di Amazon S3.

Saat Anda mengunggah sebuah objek, objek secara otomatis dienkripsi menggunakan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3) secara default. Saat Anda mengunduhnya, objek tersebut didekripsi. Untuk informasi selengkapnya, lihat Mengatur perilaku enkripsi sisi server default untuk bucket Amazon S3 dan Melindungi data dengan enkripsi.

Saat mengunggah objek, jika ingin menggunakan jenis enkripsi default yang berbeda, Anda juga dapat menentukan enkripsi sisi server dengan AWS Key Management Service (AWS KMS) kunci (SSE-KMS) dalam PUT permintaan S3 atau mengatur konfigurasi enkripsi default di bucket tujuan untuk menggunakan SSE-KMS untuk mengenkripsi data Anda. Untuk informasi lebih lanjut tentang SSE-KMS, lihat Menentukan enkripsi di sisi server dengan AWS KMS (SSE-KMS). Jika Anda ingin menggunakan kunci KMS yang dimiliki oleh akun lain, Anda harus memiliki izin untuk menggunakan kunci tersebut. Untuk informasi selengkapnya tentang izin lintas akun untuk kunci KMS, lihat Membuat kunci KMS yang dapat digunakan oleh akun lain di Panduan Pengembang AWS Key Management Service .

Jika Anda menemukan error Access Denied (403 Forbidden) di Amazon S3, Memecahkan masalah kesalahan akses ditolak (403 Forbidden) di Amazon S3 lihat untuk mempelajari lebih lanjut tentang penyebab umumnya.

Mengunggah objek

Menggunakan konsol S3

Prosedur ini menjelaskan cara untuk mengunggah objek dan folder ke bucket Amazon S3 menggunakan konsol tersebut.

Ketika Anda mengunggah objek, nama kunci objek adalah nama file dan prefiks opsional. Di konsol Amazon S3, Anda dapat membuat folder untuk mengatur objek Anda. Di Amazon S3, folder diwakili sebagai prefiks yang muncul dalam nama kunci objek. Jika Anda mengunggah objek individu ke folder di konsol Amazon S3, nama folder disertakan dalam nama kunci objek.

Misalnya, jika Anda mengunggah sebuah objek bernama `sample1.jpg` ke folder bernama `backup`, nama kuncinya adalah `backup/sample1.jpg`. Namun, objek ditampilkan di konsol sebagai `sample1.jpg` dalam folder `backup`. Untuk informasi selengkapnya tentang nama kunci, lihat Bekerja dengan metadata objek.

> **Note**
>
> Jika Anda mengganti sebuah objek atau mengubah properti apa pun di konsol Amazon S3, misalnya Kelas Penyimpanan, Enkripsi, atau Metadata, sebuah objek baru dibuat untuk menggantikan yang lama. Jika Penentuan Versi S3 diaktifkan, versi baru objek akan dibuat, dan objek yang sudah ada menjadi versi yang lebih lama. Peran yang mengubah properti juga menjadi pemilik objek baru (atau versi objek).

Saat Anda mengunggah folder, Amazon S3 mengunggah semua file dan subfolder dari folder yang ditentukan ke bucket Anda. Kode tersebut kemudian menetapkan nama kunci objek yang merupakan kombinasi nama file unggahan dan nama folder. Misalnya, jika Anda mengunggah folder bernama `/images` yang berisi dua file, `sample1.jpg` dan `sample2.jpg`, Amazon S3 mengunggah file tersebut dan kemudian menetapkan nama kunci yang sesuai, `images/sample1.jpg` dan `images/sample2.jpg`. Nama-nama kunci termasuk nama folder sebagai prefiks. Konsol Amazon S3 hanya menampilkan bagian dari nama kunci yang mengikuti `/` yang terakhir. Misalnya, dalam folder `images`, objek `images/sample1.jpg` dan `images/sample2.jpg` ditampilkan sebagai `sample1.jpg` dan `sample2.jpg`.

Untuk mengunggah file dan folder ke bucket S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Di dalam daftar Bucket, pilih nama bucket tempat Anda ingin mengunggah folder dan file.
4. Pilih Unggah.
5. Di jendela Unggah, lakukan salah satu hal berikut:
   - Seret dan jatuhkan file dan folder ke jendela Unggah.
   - Pilih Tambahkan file atau Tambahkan folder, pilih file atau folder untuk diunggah, dan pilih Buka.

6. Untuk mengaktifkan Penentuan Versi, di bawah Tujuan memilih Aktifkan Penentuan Versi Bucket.
7. Untuk mengunggah file dan folder yang terdaftar tanpa mengonfigurasi opsi pengunggahan tambahan, di bagian bawah halaman, pilih Unggah.

   Amazon S3 mengunggah objek dan folder Anda. Setelah unggahan selesai, Anda melihat pesan sukses di halaman Unggahan: status.

Untuk mengonfigurasi properti objek tambahan

1. Untuk mengubah izin daftar kontrol akses, pilih Izin.
2. Di bagian bawah Daftar kontrol akses (ACL), edit izinnya.

   Untuk informasi selengkapnya tentang izin akses objek, lihat Menggunakan konsol S3 untuk mengatur izin ACL untuk objek. Anda dapat memberikan akses baca ke objek Anda kepada publik (semua orang di dunia) untuk semua file yang Anda unggah. Namun, kami merekomendasikan untuk tidak mengubah pengaturan default untuk mengakses baca publik. Pemberian akses baca publik berlaku untuk sebagian kecil kasus penggunaan, seperti saat bucket digunakan untuk situs web. Anda dapat selalu mengubah izin objek setelah Anda mengunggah objek.
3. Untuk mengonfigurasi properti tambahan lainnya, pilih Properti.
4. Di bagian bawah Kelas penyimpanan pilih kelas penyimpanan untuk file yang Anda unggah.

   Untuk informasi selengkapnya tentang kelas penyimpanan, lihat Memahami dan mengelola kelas penyimpanan Amazon S3.
5. Untuk memperbarui pengaturan enkripsi untuk objek Anda, di bagian bawah Pengaturan enkripsi di sisi server, lakukan hal berikut.

   a. Pilih Tentukan kunci enkripsi.

   b. Di bawah Pengaturan enkripsi, pilih Gunakan pengaturan bucket untuk enkripsi default, atau Ganti pengaturan bucket untuk enkripsi default.

   c. Jika Anda memilih Ganti pengaturan bucket untuk enkripsi default, Anda harus mengonfigurasi pengaturan enkripsi berikut.

      - Untuk mengenkripsi file yang diunggah dengan menggunakan kunci yang dikelola oleh Amazon S3, pilih Kunci yang dikelola Amazon S3 (SSE-S3).

Untuk informasi selengkapnya, lihat Menggunakan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3).

- Untuk mengenkripsi file yang diunggah dengan menggunakan kunci yang disimpan di AWS Key Management Service (AWS KMS), pilih AWS Key Management Service kunci (SSE-KMS). Lalu pilih salah satu opsi berikut ini untuk kunci AWS KMS :
  - Untuk memilih dari daftar kunci KMS yang tersedia, pilih Pilih dari AWS KMS keys Anda, dan pilih Kunci KMS Anda dari daftar kunci yang tersedia.

    Kunci Kunci yang dikelola AWS (`aws/s3`) dan kunci terkelola pelanggan Anda muncul dalam daftar ini. Untuk informasi selengkapnya tentang CMK, lihat Kunci pelanggan dan AWS kunci di AWS Key Management Service Panduan Pengembang.
  - Untuk memasukkan ARN kunci KMS, pilih Masukkan AWS KMS key ARN, lalu masukkan ARN kunci KMS Anda di bidang yang muncul.
  - Untuk membuat kunci terkelola pelanggan baru di AWS KMS konsol, pilih Buat kunci KMS.

    Untuk informasi selengkapnya tentang membuat AWS KMS key, lihat Membuat kunci di Panduan AWS Key Management Service Pengembang.

> **Important**
>
> Anda hanya dapat menggunakan tombol KMS yang tersedia Wilayah AWS sama dengan bucket. Konsol Amazon S3 hanya mencantumkan kunci 100 KMS pertama di Wilayah yang sama dengan bucket. Untuk menggunakan kunci KMS yang tidak terdaftar, Anda harus memasukkan ARN kunci KMS Anda. Jika Anda ingin menggunakan kunci KMS yang dimiliki oleh akun lain, Anda harus terlebih dahulu memiliki izin untuk menggunakan kunci tersebut, dan kemudian Anda harus memasukkan ARN kunci KMS.
> Amazon S3 hanya mendukung kunci KMS enkripsi simetris, dan tidak mendukung kunci KMS asimetris. Untuk informasi selengkapnya, lihat Mengidentifikasi tombol KMS simetris dan asimetris dalam Panduan Pengembang AWS Key Management Service .

6. Untuk menggunakan checksum tambahan, pilih Aktif. Kemudian untuk Fungsi checksum, pilih fungsi yang ingin Anda gunakan. Amazon S3 menghitung dan menyimpan nilai checksum setelah menerima objek secara keseluruhan. Anda dapat menggunakan kotak Nilai

prakalkulasi untuk memberikan nilai yang telah dihitung sebelumnya. Jika sudah, Amazon S3 membandingkan nilai yang Anda berikan dengan nilai yang dihitung. Jika kedua nilai tidak cocok, Amazon S3 menghasilkan kesalahan.

Checksum tambahan memungkinkan Anda menentukan algoritma checksum yang ingin Anda gunakan untuk memverifikasi data Anda. Untuk informasi selengkapnya tentang checksum tambahan, lihat Memeriksa integritas objek di Amazon S3.

7. Untuk menambahkan tanda ke semua objek yang Anda unggah, pilih Tambahkan tag. Masukkan nama tag di bidang Kunci. Masukkan nilai untuk tanda.

   Penandaan objek memberi Anda cara untuk mengategorikan penyimpanan. Setiap tag adalah pasangan nilai kunci. Kunci dan nilai tag peka huruf besar dan kecil. Anda dapat membuat hingga 10 tanda per objek. Kunci tanda dapat terdiri dari hingga 128 karakter Unicode, dan nilai tanda dapat terdiri dari hingga 256 karakter Unicode. Untuk informasi selengkapnya tentang tag objek, lihat Mengategorikan penyimpanan Anda menggunakan tag.

8. Untuk menambahkan metadata, pilih Tambahkan metadata.

   a. Di bawah Jenis, pilih Ditentukan sistem atau Ditentukan pengguna.

      Untuk metadata yang ditentukan sistem, Anda dapat memilih header HTTP umum, seperti Jenis Konten dan Konten-Disposisi. Untuk daftar metadata yang ditentukan sistem, serta informasi tentang apakah Anda dapat menambahkan nilai, lihat Metadata objek yang ditentukan sistem. Metadata apa pun yang dimulai dengan prefiks `x-amz-meta-` diperlakukan sebagai metadata yang ditentukan pengguna. Metadata yang ditentukan pengguna disimpan bersama objek, dan dikembalikan saat Anda mengunduh objek tersebut. Baik kunci maupun nilainya harus sesuai dengan standar US-ASCII. Metadata yang ditentukan pengguna dapat berukuran sebesar 2 KB. Untuk informasi selengkapnya tentang metadata yang ditentukan sistem dan yang ditentukan pengguna, lihat Bekerja dengan metadata objek.

   b. Untuk Kunci, pilih kuncinya.

   c. Ketikkan nilai untuk kunci tersebut.

9. Untuk mengunggah objek Anda, pilih Unggah.

   Amazon S3 mengunggah objek Anda. Setelah unggahannya selesai, Anda dapat melihat pesan sukses di halaman Unggah: status.

10. Pilih Keluar.

Menggunakan AWS CLI

Anda dapat mengirim permintaan PUT untuk mengunggah objek hingga 5 GB dalam satu operasi. Untuk informasi dan contoh selengkapnya, lihat contoh `PutObject` di AWS CLI Referensi Perintah.

Penggunaan API REST

Anda dapat mengirim permintaan REST untuk mengunggah sebuah objek. Anda dapat mengirim permintaan PUT untuk mengunggah dalam satu operasi. Untuk informasi lebih lanjut, lihat PUT Objek.

Menggunakan AWS SDKs

Anda dapat menggunakan AWS SDKs untuk mengunggah objek di Amazon S3. Pustaka pembungkus SDKs menyediakan bagi Anda untuk mengunggah data dengan mudah. Untuk selengkapnya, lihat Daftar yang didukung SDKs.

Berikut adalah beberapa contoh dengan beberapa pilihan SDKs:

.NET

Contoh kode C# berikut ini membuat dua objek dengan dua permintaan `PutObjectRequest`:

- Permintaan `PutObjectRequest` yang pertama menyimpan sebuah string teks sebagai data objek sampel. Permintaan ini juga menentukan nama kunci bucket dan objek.
- Permintaan `PutObjectRequest` yang kedua mengunggah sebuah file dengan menentukan nama file tersebut. Permintaan ini juga menentukan header `ContentType` dan metadata objek opsional (sebuah judul).

Untuk informasi tentang menyiapkan dan menjalankan contoh kode, lihat Memulai SDK for .NET di AWSAWS SDK for .NET Developer Guide.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class UploadObjectTest
    {
```

```
        private const string bucketName = "*** bucket name ***";
        // For simplicity the example creates two objects from the same file.
        // You specify key names for these objects.
        private const string keyName1 = "*** key name for first object created ***";
        private const string keyName2 = "*** key name for second object created
***";
        private const string filePath = @"*** file path ***";
        private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.EUWest1;

        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            WritingAnObjectAsync().Wait();
        }

        static async Task WritingAnObjectAsync()
        {
            try
            {
                // 1. Put object-specify only key name for the new object.
                var putRequest1 = new PutObjectRequest
                {
                    BucketName = bucketName,
                    Key = keyName1,
                    ContentBody = "sample text"
                };

                PutObjectResponse response1 = await
client.PutObjectAsync(putRequest1);

                // 2. Put the object-set ContentType and add metadata.
                var putRequest2 = new PutObjectRequest
                {
                    BucketName = bucketName,
                    Key = keyName2,
                    FilePath = filePath,
                    ContentType = "text/plain"
                };

                putRequest2.Metadata.Add("x-amz-meta-title", "someTitle");
```

```
                PutObjectResponse response2 = await
 client.PutObjectAsync(putRequest2);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine(
                        "Error encountered ***. Message:'{0}' when writing an
 object"
                        , e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine(
                    "Unknown encountered on server. Message:'{0}' when writing an
 object"
                    , e.Message);
            }
        }
    }
}
```

Java

Contoh berikut ini menampilkan pembuatan dua objek. Objek pertama memiliki string teks sebagai data, dan objek kedua adalah sebuah file. Contoh ini membuat objek pertama dengan menentukan nama bucket, kunci objek, dan data teks dalam panggilan langsung ke `AmazonS3Client.putObject()`. Contoh ini membuat objek kedua dengan menggunakan `PutObjectRequest` yang menentukan nama bucket, kunci objek, dan jalur file. `PutObjectRequest` juga menentukan header `ContentType` dan metadata judul.

Untuk petunjuk cara membuat dan menguji sampel yang berfungsi, lihat Memulai di Panduan AWS SDK untuk Java Pengembang.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ObjectMetadata;
import com.amazonaws.services.s3.model.PutObjectRequest;

import java.io.File;
```

```
import java.io.IOException;

public class UploadObject {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String stringObjKeyName = "*** String object key name ***";
        String fileObjKeyName = "*** File object key name ***";
        String fileName = "*** Path to file to upload ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withRegion(clientRegion)
                    .build();

            // Upload a text string as a new object.
            s3Client.putObject(bucketName, stringObjKeyName, "Uploaded String
 Object");

            // Upload a file as a new object with ContentType and title specified.
            PutObjectRequest request = new PutObjectRequest(bucketName,
 fileObjKeyName, new File(fileName));
            ObjectMetadata metadata = new ObjectMetadata();
            metadata.setContentType("plain/text");
            metadata.addUserMetadata("title", "someTitle");
            request.setMetadata(metadata);
            s3Client.putObject(request);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

JavaScript

Contoh berikut ini menampilkan pengunggahan sebuah file yang ada ke bucket Amazon S3 di Wilayah tertentu.

```
import { readFile } from "node:fs/promises";

import {
  PutObjectCommand,
  S3Client,
  S3ServiceException,
} from "@aws-sdk/client-s3";

/**
 * Upload a file to an S3 bucket.
 * @param {{ bucketName: string, key: string, filePath: string }}
 */
export const main = async ({ bucketName, key, filePath }) => {
  const client = new S3Client({});
  const command = new PutObjectCommand({
    Bucket: bucketName,
    Key: key,
    Body: await readFile(filePath),
  });

  try {
    const response = await client.send(command);
    console.log(response);
  } catch (caught) {
    if (
      caught instanceof S3ServiceException &&
      caught.name === "EntityTooLarge"
    ) {
      console.error(
        `Error from S3 while uploading object to ${bucketName}. \
The object was too large. To upload objects larger than 5GB, use the S3 console
 (160GB max) \
or the multipart upload API (5TB max).`,
      );
    } else if (caught instanceof S3ServiceException) {
      console.error(
        `Error from S3 while uploading object to ${bucketName}.  ${caught.name}:
 ${caught.message}`,
      );
```

```
    } else {
      throw caught;
    }
  }
};
```

PHP

Contoh ini memandu Anda menggunakan kelas dari AWS SDK untuk PHP untuk mengunggah objek berukuran hingga 5 GB. Untuk file yang lebih besar, Anda harus menggunakan operasi pengunggahan API multibagian. Untuk informasi selengkapnya, lihat Mengunggah dan menyalin objek menggunakan unggahan multipart di Amazon S3.

Untuk informasi selengkapnya tentang AWS SDK for Ruby API, AWS buka SDK for Ruby - Versi 2.

Example — Membuat sebuah objek dalam sebuah bucket Amazon S3 dengan mengunggah data

Contoh PHP berikut ini membuat objek dalam bucket tertentu dengan mengunggah data menggunakan metode `putObject()`.

```
 require 'vendor/autoload.php';

use Aws\S3\Exception\S3Exception;
use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

try {
    // Upload data.
    $result = $s3->putObject([
        'Bucket' => $bucket,
        'Key'    => $keyname,
        'Body'   => 'Hello, world!',
        'ACL'    => 'public-read'
    ]);
```

```
    // Print the URL to the object.
    echo $result['ObjectURL'] . PHP_EOL;
} catch (S3Exception $e) {
    echo $e->getMessage() . PHP_EOL;
}
```

Ruby

The AWS SDK untuk Ruby - Versi 3 memiliki dua cara untuk mengunggah objek ke Amazon S3. Yang pertama menggunakan pengunggah file terkelola, yang mempermudah pengunggahan file berukuran berapa pun dari disk. Untuk menggunakan metode pengunggah file terkelola:

1. Ciptakan contoh dari kelas `Aws::S3::Resource`.
2. Referensikan objek target berdasarkan nama dan kunci bucket. Objek muncul di dalam bucket dan memiliki kunci unik yang mengidentifikasi setiap objek.
3. Panggil`#upload_file` pada objek.

Example

```
require 'aws-sdk-s3'

# Wraps Amazon S3 object actions.
class ObjectUploadFileWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  # Uploads a file to an Amazon S3 object by using a managed uploader.
  #
  # @param file_path [String] The path to the file to upload.
  # @return [Boolean] True when the file is uploaded; otherwise false.
  def upload_file(file_path)
    @object.upload_file(file_path)
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't upload file #{file_path} to #{@object.key}. Here's why:
 #{e.message}"
```

```
    false
  end
end

# Example usage:
def run_demo
  bucket_name = "amzn-s3-demo-bucket"
  object_key = "my-uploaded-file"
  file_path = "object_upload_file.rb"

  wrapper = ObjectUploadFileWrapper.new(Aws::S3::Object.new(bucket_name,
 object_key))
  return unless wrapper.upload_file(file_path)

  puts "File #{file_path} successfully uploaded to #{bucket_name}:#{object_key}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Cara kedua bahwa AWS SDK untuk Ruby - Versi 3 dapat meng-upload objek menggunakan `#put` metode`Aws::S3::Object`. Ini berguna jika objek adalah string atau I/O objek yang bukan file pada disk. Untuk menggunakan metode ini:

1. Buat instans dari kelas `Aws::S3::Resource`.
2. Referensikan objek target berdasarkan nama dan kunci bucket.
3. Panggil`#put`, meneruskan string atau I/O objek.

Example

```
require 'aws-sdk-s3'

# Wraps Amazon S3 object actions.
class ObjectPutWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  def put_object(source_file_path)
```

```
    File.open(source_file_path, 'rb') do |file|
      @object.put(body: file)
    end
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't put #{source_file_path} to #{object.key}. Here's why:
 #{e.message}"
    false
  end
end

# Example usage:
def run_demo
  bucket_name = "amzn-s3-demo-bucket"
  object_key = "my-object-key"
  file_path = "my-local-file.txt"

  wrapper = ObjectPutWrapper.new(Aws::S3::Object.new(bucket_name, object_key))
  success = wrapper.put_object(file_path)
  return unless success

  puts "Put file #{file_path} into #{object_key} in #{bucket_name}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Mencegah mengunggah objek dengan nama kunci yang identik

Anda dapat memeriksa keberadaan objek di bucket Anda sebelum membuatnya menggunakan penulisan bersyarat pada operasi unggahan. Ini dapat mencegah penimpaan data yang ada. Penulisan bersyarat akan memvalidasi tidak ada objek yang ada dengan nama kunci yang sama yang sudah ada di bucket Anda saat mengunggah.

Anda dapat menggunakan penulisan bersyarat untuk PutObjectatau CompleteMultipartUploadpermintaan.

Untuk informasi selengkapnya tentang permintaan bersyarat, lihat,Tambahkan prasyarat ke operasi S3 dengan permintaan bersyarat.

Mengunggah dan menyalin objek menggunakan unggahan multipart di Amazon S3

Unggahan multipart memungkinkan Anda mengunggah satu objek ke Amazon S3 sebagai satu set bagian. Setiap bagian merupakan bagian data objek yang saling berkaitan. Anda dapat mengunggah bagian objek ini secara independen, dan dalam urutan apa pun. Untuk unggahan, AWS klien Anda yang diperbarui secara otomatis menghitung checksum objek dan mengirimkannya ke Amazon S3 bersama dengan ukuran objek sebagai bagian dari permintaan. Jika ada transmisi bagian mana pun yang gagal, Anda dapat mentransmisikan ulang bagian tersebut tanpa memengaruhi bagian lainnya. Setelah semua bagian objek Anda diunggah, Amazon S3 merakitnya untuk membuat objek. Ini adalah praktik terbaik untuk menggunakan unggahan multipart untuk objek yang 100 MB atau lebih besar daripada mengunggahnya dalam satu operasi.

Penggunaan unggahan multibagian memberikan keuntungan sebagai berikut:

- Peningkatan throughput–Anda dapat mengunggah bagian-bagian secara paralel untuk meningkatkan throughput.
- Pemulihan yang cepat dari masalah jaringan apa pun–Ukuran bagian yang lebih kecil meminimalkan dampak pengunggahan ulang karena kesalahan jaringan.
- Jeda dan pelanjutan pengunggahan objek–Anda dapat mengunggah bagian-bagian objek kapan saja. Setelah Anda memulai unggahan multibagian, tidak ada kedaluwarsa; Anda harus secara eksplisit menyelesaikan atau menghentikan unggahan multibagian.
- Mulai unggahan sebelum Anda mengetahui ukuran objek akhir — Anda dapat mengunggah objek saat Anda membuatnya.

Kami menyarankan agar Anda menggunakan unggahan multibagian dengan cara berikut:

- Jika Anda mengunggah objek besar melalui jaringan bandwidth tinggi yang stabil, gunakan unggahan multipart untuk memaksimalkan penggunaan bandwidth yang tersedia dengan mengunggah bagian objek secara paralel untuk kinerja multi-threaded.
- Jika Anda mengunggah melalui jaringan jerawatan, gunakan unggahan multibagian untuk meningkatkan ketahanan terhadap kesalahan jaringan dengan menghindari pengunggahan dimulai ulang. Saat menggunakan unggahan multibagian, Anda hanya perlu mencoba mengunggah kembali bagian-bagian yang terputus selama pengunggahan. Anda tidak perlu mengunggah ulang objek Anda dari awal.

> **Note**
>
> Untuk informasi selengkapnya tentang menggunakan kelas penyimpanan Amazon S3 Express One Zone dengan bucket direktori, lihat S3 Express One Zone dan Bekerja dengan bucket direktori. Untuk informasi selengkapnya tentang penggunaan unggahan multibagian dengan S3 Express One Zone dan bucket direktori, lihat Menggunakan unggahan multibagian dengan bucket direktori.

Proses pengunggahan multibagian

Unggahan multipart adalah proses tiga langkah: Anda memulai unggahan, mengunggah bagian objek, dan—setelah mengunggah semua bagian—menyelesaikan unggahan multipart. Setelah menerima permintaan upload multipart lengkap, Amazon S3 membuat objek dari bagian yang diunggah, dan Anda dapat mengakses objek seperti halnya objek lain di bucket Anda.

Anda dapat membuat daftar semua unggahan multibagian yang sedang berlangsung, atau mendapatkan daftar bagian yang telah Anda unggah untuk unggahan multibagian tertentu. Setiap operasi ini dijelaskan dalam bagian ini.

Menginisiasi unggahan multibagian

Saat Anda mengirim permintaan untuk memulai unggahan multipart, pastikan untuk menentukan jenis checksum. Amazon S3 kemudian akan mengembalikan respons dengan ID unggahan, yang merupakan pengenal unik untuk unggahan multipart Anda. ID unggahan ini diperlukan saat Anda mengunggah suku cadang, mencantumkan bagian, menyelesaikan unggahan, atau menghentikan unggahan. Jika Anda ingin memberikan metadata yang menjelaskan objek yang sedang diunggah, Anda harus menyediakannya dalam permintaan untuk memulai unggahan multipart. Pengguna anonim tidak dapat memulai unggahan multipart.

Pengunggahan bagian-bagian

Saat mengunggah bagian, Anda harus menentukan nomor bagian selain ID unggahan. Anda dapat memilih nomor bagian antara 1 hingga 10.000. Nomor bagian secara unik mengidentifikasi sebuah bagian dan posisinya dalam objek yang Anda unggah. Nomor bagian yang Anda pilih tidak harus berurutan (misalnya, nomornya dapat berupa 1, 5, dan 14). Ketahuilah bahwa jika Anda mengunggah bagian baru menggunakan nomor bagian yang sama dengan bagian yang diunggah sebelumnya, bagian yang diunggah sebelumnya akan ditimpa.

Saat Anda mengunggah bagian, Amazon S3 mengembalikan jenis algoritma checksum dengan nilai checksum untuk setiap bagian sebagai header dalam respons. Untuk setiap unggahan bagian, Anda harus mencatat nomor bagian dan ETag nilainya. Anda harus memasukkan nilai-nilai ini dalam permintaan selanjutnya untuk menyelesaikan unggahan multibagian. Setiap bagian akan memiliki sendiri ETag pada saat diunggah. Namun, setelah unggahan multipart selesai dan semua bagian dikonsolidasikan, semua bagian menjadi milik satu ETag sebagai checksum checksum.

> **Important**
>
> Setelah Anda memulai pengunggahan multibagian dan mengunggah satu atau beberapa bagian, Anda harus menyelesaikan atau menghentikan unggahan multibagian untuk menghentikan biaya penyimpanan suku cadang yang diunggah. Hanya setelah Anda menyelesaikan atau menghentikan unggahan multibagian, Amazon S3 akan membebaskan penyimpanan suku cadang dan berhenti menagih Anda untuk penyimpanan suku cadang. Setelah menghentikan unggahan multibagian, Anda tidak dapat mengunggah bagian apa pun menggunakan ID unggahan itu lagi. Jika sebagian unggahan sedang berlangsung, mereka masih dapat berhasil atau gagal bahkan setelah Anda menghentikan unggahan. Untuk memastikan Anda membebaskan semua penyimpanan yang dikonsumsi oleh semua bagian, Anda harus menghentikan unggahan multibagian hanya setelah semua unggahan bagian selesai.

Penyelesaian unggahan multibagian

Saat Anda menyelesaikan unggahan multibagian, Amazon S3 membuat objek dengan menggabungkan bagian-bagian dalam urutan menaik berdasarkan nomor bagian. Jika metadata objek disediakan dalam permintaan mulai unggahan multibagian, Amazon S3 akan mengaitkan metadatanya dengan objek tersebut. Setelah permintaan selesai sepenuhnya, bagiantersebut tidak akan ada lagi.

Permintaan pengunggahan multibagian lengkap Anda harus menyertakan ID unggahan dan daftar nomor bagian serta ETag nilainya yang sesuai. Respons Amazon S3 mencakup ETag yang secara unik mengidentifikasi data objek gabungan. ETag Ini belum tentu MD5 hash dari data objek.

Saat Anda memberikan checksum objek lengkap selama pengunggahan multibagian, AWS SDK meneruskan checksum ke Amazon S3, dan S3 memvalidasi sisi server integritas objek, membandingkannya dengan nilai yang diterima. Kemudian, S3 menyimpan objek jika nilainya cocok. Jika kedua nilai tidak cocok, Amazon S3 gagal permintaan dengan `BadDigest` kesalahan.

Checksum objek Anda juga disimpan dalam metadata objek yang nantinya akan Anda gunakan untuk memvalidasi integritas data objek.

Contoh panggilan unggahan multibagian

Untuk contoh ini, asumsikan bahwa Anda membuat unggahan multipart untuk file 100 GB. Dalam hal ini, Anda akan memiliki panggilan API berikut untuk seluruh proses. Akan ada total 1.002 panggilan API.

- Panggilan `CreateMultipartUpload` untuk memulai proses.
- 1.000 `UploadPart` panggilan individu, masing-masing mengunggah bagian dari 100 MB, dengan ukuran total 100 GB.
- `CompleteMultipartUpload`Panggilan untuk menyelesaikan proses.

Pendaftaran unggahan multibagian

Anda dapat mendaftar bagian-bagian dari unggahan multibagian tertentu atau semua unggahan multibagian yang sedang berlangsung. Operasi bagian daftar mengembalikan informasi bagian yang Anda unggah untuk unggahan multibagian tertentu. Untuk setiap permintaan daftar bagian, Amazon S3 akan menampilkan informasi bagian untuk unggahan multibagian tertentu, hingga maksimum 1.000 bagian. Jika ada lebih dari 1.000 bagian dalam unggahan multipart, Anda harus mengirim serangkaian permintaan bagian daftar untuk mengambil semua bagian. Perhatikan bahwa daftar bagian yang ditampilkan tidak mencakup bagian yang belum selesai diunggah. Dengan menggunakan operasi daftar unggahan multibagian, Anda dapat memperoleh daftar unggahan multibagian yang sedang berlangsung.

Unggahan multibagian yang sedang berlangsung adalah unggahan yang telah Anda mulai, tetapi belum selesai atau dihentikan. Setiap permintaan akan ditampilkan sebanyak maksimum 1.000 unggahan multibagian. Jika ada lebih dari 1.000 unggahan multibagian yang sedang berlangsung, Anda harus mengirim permintaan tambahan untuk mengambil unggahan multibagian yang tersisa. Gunakan pendaftaran yang ditampilkan untuk verifikasi.

> **Important**
>
> Jangan menggunakan hasil pendaftaran ini saat mengirim permintaan selesaikan unggahan multibagian. Sebagai gantinya, pertahankan daftar nomor bagian Anda sendiri yang Anda tentukan saat mengunggah bagian dan ETag nilai terkait yang dikembalikan Amazon S3.

Checksum dengan operasi unggahan multibagian

Saat mengunggah sebuah objek ke Amazon S3, Anda dapat menentukan algoritma checksum untuk Amazon S3 untuk digunakan. Secara default, konsol AWS SDK dan S3 menggunakan algoritme untuk semua unggahan objek, yang dapat Anda timpa. Jika Anda menggunakan SDK yang lebih lama dan objek yang diunggah tidak memiliki checksum yang ditentukan, Amazon S3 secara otomatis menggunakan algoritma checksum CRC-64/NVME (). `CRC64NVME` (Ini juga merupakan opsi yang disarankan untuk verifikasi integritas data yang efisien.) Saat menggunakan CRC-64/NVME, Amazon S3 menghitung checksum objek lengkap setelah unggahan multipart atau satu bagian selesai. Algoritma checksum CRC-64/NVME digunakan untuk menghitung checksum langsung dari seluruh objek, atau checksum checksum, untuk setiap bagian individu.

Setelah Anda mengunggah objek ke S3 menggunakan unggahan multipart, Amazon S3 menghitung nilai checksum untuk setiap bagian, atau untuk objek lengkap—dan menyimpan nilainya. Anda dapat menggunakan S3 API atau AWS SDK untuk mengambil nilai checksum dengan cara berikut:

- Untuk masing-masing bagian, Anda dapat menggunakan `GetObject`atau `HeadObject`. Jika Anda ingin mengambil nilai checksum untuk masing-masing bagian dari unggahan multibagian saat masih dalam proses, Anda dapat menggunakannya. `ListParts`
- Untuk seluruh objek, Anda dapat menggunakan `PutObject`. Jika Anda ingin melakukan upload multipart dengan checksum objek lengkap, gunakan `CreateMultipartUpload`dan `CompleteMultipartUpload`dengan menentukan jenis checksum objek lengkap. Untuk memvalidasi nilai checksum dari seluruh objek atau untuk mengonfirmasi jenis checksum mana yang digunakan dalam unggahan multipart, gunakan. `ListParts`

> **Important**
>
> Jika Anda menggunakan unggahan multibagian dengan Checksum, nomor bagian untuk setiap unggahan bagian (dalam unggahan multibagian) harus menggunakan nomor bagian berturut-turut dan dimulai dengan 1. Saat menggunakan Checksum, jika Anda mencoba menyelesaikan permintaan unggahan multibagian dengan nomor bagian yang tidak berurutan, Amazon S3 menghasilkan kesalahan. `HTTP 500 Internal Server`

Untuk informasi selengkapnya tentang cara kerja checksum dengan objek unggahan multibagian, lihat. Memeriksa integritas objek di Amazon S3

Untuk end-to-end prosedur yang menunjukkan cara mengunggah objek menggunakan unggahan multipart dengan checksum tambahan, lihat. Tutorial: Unggah objek melalui unggahan multipart dan verifikasi integritas datanya

Operasi pengunggahan multibagian serentak

Dalam lingkungan pengembangan terdistribusi, aplikasi Anda dapat memulai beberapa pembaruan pada objek yang sama secara bersamaan. Aplikasi Anda dapat memulai beberapa unggahan multibagian menggunakan kunci objek yang sama. Untuk setiap unggahan ini, aplikasi Anda kemudian dapat mengunggah bagian dan mengirim sebuah permintaan menyelesaikan unggahan ke Amazon S3 untuk membuat objek. Jika Penentuan Versi S3 diaktifkan pada bucket, menyelesaikan pengunggahan multibagian akan selalu membuat versi baru. Saat Anda memulai beberapa unggahan multibagian yang menggunakan kunci objek yang sama dalam bucket berkemampuan versi, versi objek saat ini ditentukan oleh pengunggahan yang paling baru dimulai (). `createdDate`

Misalnya, Anda memulai `CreateMultipartUpload` permintaan objek pada pukul 10:00 pagi. Kemudian, Anda mengirimkan `CreateMultipartUpload` permintaan kedua untuk objek yang sama pada pukul 11:00. Karena permintaan kedua dikirimkan paling baru, objek yang diunggah oleh permintaan 11:00 AM menjadi versi saat ini, bahkan jika unggahan pertama selesai setelah yang kedua. Untuk bucket yang tidak mengaktifkan versi, ada kemungkinan permintaan lain yang diterima antara waktu saat unggahan multibagian dimulai dan saat selesai, permintaan lainnya mungkin diutamakan.

Contoh lain ketika permintaan upload multipart bersamaan dapat diutamakan adalah jika operasi lain menghapus kunci setelah Anda memulai unggahan multipart dengan kunci tersebut. Sebelum Anda menyelesaikan operasi, respons unggahan multibagian lengkap mungkin menunjukkan pembuatan objek yang berhasil tanpa Anda pernah melihat objek.

Mencegah mengunggah objek dengan nama kunci yang identik selama pengunggahan multipart

Anda dapat memeriksa keberadaan objek di bucket Anda sebelum membuatnya menggunakan penulisan bersyarat pada operasi unggahan. Ini dapat mencegah penimpaan data yang ada. Penulisan bersyarat akan memvalidasi bahwa tidak ada objek yang ada dengan nama kunci yang sama yang sudah ada di bucket Anda saat mengunggah.

Anda dapat menggunakan penulisan bersyarat untuk PutObjectatau CompleteMultipartUploadpermintaan.

Untuk informasi selengkapnya tentang permintaan bersyarat, lihat,[Tambahkan prasyarat ke operasi S3 dengan permintaan bersyarat](#).

Unggahan multibagian dan harga

Setelah Anda memulai unggahan multibagian, Amazon S3 akan menyimpan semua bagian hingga Anda menyelesaikan atau menghentikan unggahan. Sepanjang masa pakainya, Anda akan ditagih untuk semua penyimpanan, bandwidth, dan permintaan untuk unggahan multibagian ini dan bagian terkaitnya.

Bagian-bagian ini ditagih sesuai dengan kelas penyimpanan yang ditentukan saat bagian diunggah. Namun, Anda tidak akan ditagih untuk suku cadang ini jika diunggah ke S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive. Suku cadang multipart yang sedang berlangsung untuk permintaan PUT ke kelas penyimpanan S3 Glacier Flexible Retrieval ditagih sebagai penyimpanan staging S3 Glacier Flexible Retrieval pada tingkat penyimpanan Standar S3 hingga unggahan selesai. Selain itu, keduanya `CreateMultipartUpload` dan `UploadPart` ditagih dengan tarif Standar S3. Hanya `CompleteMultipartUpload` permintaan yang ditagih dengan tarif Pengambilan Fleksibel S3 Glacier. Demikian pula, suku cadang multipart yang sedang berlangsung untuk kelas penyimpanan PUT ke S3 Glacier Deep Archive ditagih sebagai penyimpanan pementasan Pengambilan Fleksibel Gletser S3 pada tingkat penyimpanan Standar S3 hingga unggahan selesai, dengan hanya permintaan yang dikenakan tarif S3 Glacier Deep Archive. `CompleteMultipartUpload`

Jika Anda menghentikan pengunggahan multibagian, Amazon S3 menghapus artefak unggahan dan semua bagian yang Anda unggah. Anda tidak akan ditagih untuk artefak itu. Tidak ada biaya penghapusan awal untuk menghapus unggahan multibagian yang belum selesai terlepas dari kelas penyimpanan yang ditentukan. Untuk informasi selengkapnya tentang harga, lihat Harga Amazon S3.

> **Note**
>
> Untuk meminimalkan biaya penyimpanan, kami menyarankan untuk Anda mengonfigurasi aturan siklus hidup untuk menghapus unggahan multibagian yang tidak lengkap setelah beberapa hari tertentu dengan menggunakan tindakan `AbortIncompleteMultipartUpload`. Untuk informasi selengkapnya tentang membuat aturan siklus hidup untuk menghapus unggahan multibagian yang tidak lengkap, lihat [Mengonfigurasi konfigurasi siklus hidup bucket untuk menghapus unggahan multibagian yang tidak lengkap](#).

Dukungan API untuk unggahan multibagian

Bagian berikut dalam Referensi API Amazon Simple Storage Service menjelaskan tentang API REST untuk unggahan multibagian.

Untuk panduan unggahan multibagian yang menggunakan fungsi AWS Lambda, lihat Mengunggah objek besar ke Amazon S3 menggunakan akselerasi unggahan dan transfer multibagian.

- Buat Unggahan Multibagian
- Unggah Bagian
- Unggah Bagian (Salinan)
- Selesaikan Unggahan Multibagian
- Batalkan Unggahan Multibagian
- Daftarkan Bagian
- Daftarkan Unggahan Multibagian

AWS Command Line Interface dukungan untuk upload multipart

Topik-topik berikut dalam AWS Command Line Interface menjelaskan operasi untuk upload multipart.

- Mulai Unggahan Multibagian
- Unggah Bagian
- Unggah Bagian (Salinan)
- Selesaikan Unggahan Multibagian
- Batalkan Unggahan Multibagian
- Daftarkan Bagian
- Daftarkan Unggahan Multibagian

AWS Dukungan SDK untuk unggahan multipart

Anda dapat menggunakan AWS SDKs untuk mengunggah objek di beberapa bagian. Untuk daftar tindakan API yang AWS SDKs didukung, lihat:

- Buat Unggahan Multibagian

- Unggah Bagian
- Unggah Bagian (Salinan)
- Selesaikan Unggahan Multibagian
- Batalkan Unggahan Multibagian
- Daftarkan Bagian
- Daftarkan Unggahan Multibagian

API dan izin unggahan multibagian

Anda harus memiliki izin yang diperlukan untuk menggunakan operasi pengunggahan multibagian. Anda dapat menggunakan daftar kontrol akses (ACLs), kebijakan bucket, atau kebijakan pengguna untuk memberikan izin individu untuk melakukan operasi ini. Tabel berikut mencantumkan izin yang diperlukan untuk berbagai operasi unggahan multibagian saat menggunakan ACLs, kebijakan bucket, atau kebijakan pengguna.

Tindakan	Izin yang diperlukan
Buat Unggahan Multibagian	Anda harus diizinkan untuk melakukan `s3:PutObject` tindakan pada objek untuk membuat permintaan upload multipart. Pemilik bucket dapat mengizinkan pengguna utama lain untuk melakukan tindakan `s3:PutObject` .
Mulai Unggahan Multibagian	Anda harus diizinkan untuk melakukan `s3:PutObject` tindakan pada objek untuk memulai unggahan multipart. Pemilik bucket dapat mengizinkan penanggung jawab lain untuk melakukan tindakan `s3:PutObject` .
Inisiator	Elemen kontainer yang mengidentifikasi siapa yang memulai unggahan multibagi an. Jika inisiator adalah Akun AWS, elemen ini memberikan informasi yang sama dengan elemen Pemilik. Jika inisiator adalah seorang pengguna IAM, elemen ini akan menyediakan ARN pengguna dan nama tampilan.
Unggah Bagian	Anda harus diizinkan untuk melakukan tindakan `s3:PutObject` pada sebuah objek untuk mengunggah sebuah bagian.

Tindakan	Izin yang diperlukan
	Pemilik bucket harus mengizinkan inisiator untuk melakukan tindakan `s3:PutObject` pada sebuah objek agar inisiator dapat mengunggah bagian untuk objek tersebut.
Unggah Bagian (Salinan)	Anda harus diizinkan untuk melakukan tindakan `s3:PutObject` pada sebuah objek guna mengunggah sebuah bagian. Karena Anda mengunggah sebuah bagian dari objek yang sudah ada, Anda harus diizinkan `s3:GetObject` pada objek sumber. Agar inisiator dapat mengunggah bagian untuk sebuah objek, pemilik bucket harus mengizinkan inisiator untuk melakukan tindakan `s3:PutObject` pada objek.
Selesaikan Unggahan Multibagian	Anda harus diizinkan untuk melakukan tindakan `s3:PutObject` pada sebuah objek untuk menyelesaikan sebuah unggahan multibagian. Pemilik bucket harus mengizinkan inisiator untuk melakukan tindakan `s3:PutObject` pada sebuah objek agar inisiator dapat menyelesaikan sebuah unggahan multibagian untuk objek tersebut.
Hentikan Unggahan Multibagian	Anda harus diizinkan untuk melakukan tindakan `s3:AbortMultipartUpload` untuk menghentikan sebuah unggahan multibagian. Secara default, pemilik bucket dan inisiator unggahan multipart diizinkan untuk melakukan tindakan ini sebagai bagian dari kebijakan bucket IAM dan S3. Jika inisiator adalah pengguna IAM, pengguna tersebut juga Akun AWS diizinkan untuk menghentikan unggahan multibagian tersebut. Dengan kebijakan titik akhir VPC, inisiator unggahan multibagian tidak secara otomatis mendapatkan izin untuk melakukan tindakan. `s3:AbortMultipartUpload` Selain pengaturan default ini, pemilik bucket dapat mengizinkan pengguna utama lain untuk melakukan tindakan `s3:AbortMultipartUpload` terhadap suatu objek. Pemilik bucket dapat menolak kemampuan pengguna utama mana pun untuk melakukan tindakan `s3:AbortMultipartUpload` .

Tindakan	Izin yang diperlukan
Daftarkan Bagian	Anda harus diizinkan untuk melakukan tindakan `s3:ListMultipartUploadParts` untuk mendaftar bagian dalam unggahan multibagian. Secara default, pemilik bucket memiliki izin untuk mendaftarkan bagian untuk unggahan multibagian apa pun pada bucket. Inisiator unggahan multibagian memiliki izin untuk mendaftarkan bagian-bagian dari unggahan multibagian tertentu. Jika inisiator unggahan multipart adalah pengguna IAM, Akun AWS pengendali pengguna IAM tersebut juga memiliki izin untuk membuat daftar bagian dari unggahan tersebut. Selain pengaturan default ini, pemilik bucket dapat mengizinkan pengguna utama lain untuk melakukan tindakan `s3:ListMultipartUploadParts` terhadap suatu objek. Pemilik bucket dapat juga menolak kemampuan pengguna utama mana pun untuk melakukan tindakan `s3:ListMultipartUploadParts` .
Daftarkan Unggahan Multibagian	Anda harus diizinkan untuk melakukan tindakan `s3:ListBucketMultipartUploads` pada sebuah bucket untuk mendaftar unggahan multibagian yang sedang berlangsung pada bucket tersebut. Selain pengaturan default ini, pemilik bucket dapat mengizinkan pengguna utama lain untuk melakukan tindakan `s3:ListBucketMultipartUploads` terhadap suatu objek.

Tindakan	Izin yang diperlukan
AWS KMS Enkripsi dan Dekripsi izin terkait	Untuk melakukan upload multipart dengan enkripsi menggunakan kunci AWS Key Management Service (AWS KMS) KMS, pemohon harus memiliki izin berikut: • `kms:Decrypt` Dan `kms:GenerateDataKey` tindakan pada kuncinya. • `kms:GenerateDataKey` Tindakan untuk CreateMultipartUploadAPI. • `kms:Decrypt` Tindakan pada UploadPartdan UploadPartCopy APIs. Izin ini diperlukan karena Amazon S3 harus mendekripsi dan membaca data dari bagian file terenkripsi sebelum menyelesaikan unggahan multibagi an. `kms:Decrypt` Izin, dan enkripsi sisi server dengan kunci enkripsi yang disediakan pelanggan, juga diperlukan bagi Anda untuk mendapatkan nilai checksum objek. Jika Anda tidak memiliki izin yang diperlukan ini saat menggunakan CompleteMultipartUploadAPI, objek dibuat tanpa nilai checksum. Jika pengguna atau peran IAM Anda Akun AWS sama dengan kunci KMS, maka validasi bahwa Anda memiliki izin pada kebijakan kunci dan IAM. Jika pengguna atau peran IAM Anda memiliki akun yang berbeda dengan kunci KMS Anda, maka Anda harus memiliki izin pada kebijakan kunci maupun pengguna atau pengguna IAM Anda.
SSE-C (enkripsi sisi server dengan kunci enkripsi yang disediakan pelanggan)	Saat Anda menggunakan CompleteMultipartUploadAPI, Anda harus menyediak an SSE-C (enkripsi sisi server dengan kunci enkripsi yang disediakan pelanggan), atau objek Anda akan dibuat tanpa checksum, dan tidak ada nilai checksum yang dikembalikan.

Untuk informasi tentang hubungan antara izin ACL dan izin dalam kebijakan akses, lihat Pemetaan izin ACL dan izin kebijakan akses. Untuk informasi selengkapnya tentang pengguna IAM dan praktik terbaik, lihat Identitas IAM (pengguna, grup pengguna, dan peran) di Panduan Pengguna IAM.

Checksum dengan operasi unggahan multibagian

Ada tiga Amazon S3 APIs yang digunakan untuk melakukan unggahan multipart yang sebenarnya: `CreateMultipartUpload`,, `UploadPart`dan. `CompleteMultipartUpload` Tabel berikut menunjukkan header dan nilai checksum mana yang harus disediakan untuk masing-masing: APIs

Algoritma checksum	Jenis Checksum	**CreateMultipartUpload**	**UploadPart**	**CompleteMultipartUpload**

CRC-64/NVME () `CRC64NVME`	Objek penuh	Header yang diperlukan:
		`x-amz-checksum-algorithm`
			Header opsional:
			`x-amz-checksum-crc64nvme`
				Header opsional:
				`x-amz-checksum-algorithm`
				`x-amz-crc64`

CRC-32 () `CRC32`
CRC 32-C () `CRC32C`
	Objek penuh	Header yang diperlukan:
		`x-amz-checksum-algorithm`
		`x-amz-checksum-type`
			Header opsional:
			`x-amz-checksum-crc64nvme`
				Header opsional:
				`x-amz-checksum-algorithm`
				`x-amz-crc32`
				`x-amz-crc32c`

CRC-32 () `CRC32`
CRC-32C () `CRC32C`
SHA-1 () `SHA1`
SHA-256 () `SHA256`
	Komposit	Header yang diperlukan:
		`x-amz-checksum-algorithm`
			Header yang diperlukan:
			`x-amz-checksum-crc32`
			`x-amz-checksum-crc32c`
				Header yang diperlukan:
				Semua checksum tingkat bagian harus disertakan dalam permintaan. `CompleteM`

Algoritma checksum	Jenis Checksum	**CreateMultipartUpload**	**UploadPart**	**CompleteMultipartUpload**

			`x-amz-checksum-sha1`
			
			`x-amz-checksum-sha256`
				`ultiPartUpload`
				
				Header opsional:
				
				`x-amz-crc32`
				
				`x-amz-crc32c`
				
				`x-amz-sha1`
				
				`x-amz-sha256`

Topik

- Mengonfigurasi siklus hidup bucket untuk menghapus unggahan multibagian yang tidak lengkap
- Pengunggahan objek menggunakan unggahan multibagian
- Pengunggahan sebuah direktori menggunakan kelas TransferUtility .NET tingkat tinggi
- Mendaftarkan unggahan multibagian
- Melacak unggahan multipart dengan AWS SDKs
- Membatalkan unggahan multibagian
- Menyalin objek menggunakan unggahan multibagian
- Tutorial: Unggah objek melalui unggahan multipart dan verifikasi integritas datanya
- Batas unggahan multibagian Amazon S3

Mengonfigurasi siklus hidup bucket untuk menghapus unggahan multibagian yang tidak lengkap

Sebagai praktik terbaik, kami menyarankan Anda untuk mengonfigurasi aturan siklus hidup dengan menggunakan tindakan `AbortIncompleteMultipartUpload` untuk meminimalkan biaya

penyimpanan Anda. Untuk informasi lebih lanjut tentang membatalkan unggahan multibagian, lihat Membatalkan unggahan multibagian.

Amazon S3 mendukung aturan siklus hidup bucket yang dapat Anda gunakan untuk mengarahkan Amazon S3 menghentikan unggahan multibagian yang tidak diselesaikan dalam jumlah hari tertentu setelah prosesnya dimulai. Jika unggahan multibagian tidak selesai dalam jangka waktu yang ditentukan, unggahan tersebut memenuhi syarat untuk operasi pembatalan. Amazon S3 kemudian menghentikan unggahan multibagian dan menghapus bagian yang terkait dengan unggahan multibagian. Aturan ini berlaku untuk unggahan multibagian yang ada dan yang Anda buat nanti.

Berikut ini adalah contoh konfigurasi siklus hidup yang menentukan aturan dengan tindakan `AbortIncompleteMultipartUpload`.

```
<LifecycleConfiguration>
    <Rule>
        <ID>sample-rule</ID>
        <Prefix></Prefix>
        <Status>Enabled</Status>
        <AbortIncompleteMultipartUpload>
          <DaysAfterInitiation>7</DaysAfterInitiation>
        </AbortIncompleteMultipartUpload>
    </Rule>
</LifecycleConfiguration>
```

Dalam contoh ini, aturan tidak menentukan nilai untuk elemen `Prefix` (prefiks nama kunci objek). Oleh karena itu, aturan ini berlaku untuk semua objek dalam bucket tempat Anda memulai unggahan multibagian. Setiap unggahan multibagian yang dimulai dan tidak diselesaikan dalam waktu tujuh hari memenuhi syarat untuk operasi pembatalan. Tindakan membatalkan tidak memengaruhi unggahan multibagian yang sudah selesai.

Untuk informasi selengkapnya tentang konfigurasi siklus hidup bucket, lihat Mengelola siklus hidup objek.

> **Note**
>
> Jika unggahan multibagian selesai dalam jumlah hari yang ditentukan dalam aturan, tindakan siklus hidup `AbortIncompleteMultipartUpload` tidak berlaku lagi (yaitu, Amazon S3 tidak akan mengambil tindakan apa pun). Selain itu, tindakan ini tidak berlaku untuk objek. Tidak ada objek yang dihapus oleh tindakan siklus hidup ini. Selain itu, Anda tidak akan

dikenakan biaya penghapusan awal untuk Siklus Hidup S3 saat Anda menghapus bagian unggahan multibagian yang tidak lengkap.

Menggunakan konsol S3

Untuk mengelola unggahan multibagian yang tidak lengkap secara otomatis, Anda dapat menggunakan konsol S3 untuk membuat aturan siklus hidup untuk mengakhiri byte unggahan multibagian yang tidak lengkap dari bucket Anda setelah jumlah hari yang ditentukan. Prosedur berikut menunjukkan cara menambahkan aturan siklus hidup untuk menghapus unggahan multibagian yang tidak lengkap setelah 7 hari. Untuk informasi selengkapnya tentang cara menambahkan aturan siklus hidup, lihat Menyetel konfigurasi Siklus Hidup S3 pada bucket.

Untuk menambahkan aturan siklus hidup untuk membatalkan unggahan multibagian yang tidak lengkap yang berumur lebih dari 7 hari

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di daftar Bucket, pilih nama bucket yang ingin Anda buatkan aturan siklus hidupnya.
3. Pilih tab Manajemen, dan pilih Buat aturan siklus hidup.
4. Di Nama aturan siklus hidup, masukkan nama untuk aturan Anda.

   Nama dalam bucket harus unik.
5. Pilih cakupan aturan siklus hidup:

   - Untuk membuat aturan siklus hidup untuk semua objek dengan prefiks tertentu, pilih Batasi cakupan aturan ini menggunakan satu atau lebih filter, dan masukkan bidang Prefiks.
   - Untuk membuat aturan siklus hidup ini pada semua objek dalam bucket, pilih Aturan ini berlaku untuk semua objek di dalam bucket, lalu pilih Saya menyatakan bahwa aturan ini berlaku untuk semua objek dalam bucket.
6. Di bagian bawah Tindakan aturan siklus hidup, pilih Hapus penanda hapus objek kedaluwarsa atau unggahan multibagian yang tidak lengkap.
7. Di bagian bawah Hapus penanda hapus objek kedaluwarsa atau unggahan multibagian yang tidak lengkap, pilih Hapus unggahan multibagian yang tidak lengkap.
8. Di bidang Jumlah hari, masukkan jumlah hari setelahnya untuk menghapus unggahan multibagian yang tidak lengkap (untuk contoh ini, 7 hari).
9. Pilih Buat aturan.

Menggunakan AWS CLI

Perintah berikut `put-bucket-lifecycle-configuration`AWS Command Line Interface (AWS CLI) menambahkan konfigurasi siklus hidup untuk bucket yang ditentukan. Untuk menggunakan perintah ini, ganti *`user input placeholders`* dengan informasi Anda.

```
aws s3api put-bucket-lifecycle-configuration  \
        --bucket amzn-s3-demo-bucket  \
        --lifecycle-configuration filename-containing-lifecycle-configuration
```

Contoh berikut menunjukkan cara menambahkan aturan siklus hidup untuk membatalkan unggahan multibagian yang tidak lengkap dengan menggunakan AWS CLI. Ini mencakup contoh konfigurasi siklus hidup JSON untuk membatalkan unggahan multibagian yang tidak lengkap yang berusia lebih dari 7 hari.

Untuk menggunakan perintah CLI dalam contoh ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

Untuk menambahkan aturan siklus hidup untuk membatalkan unggahan multibagian yang tidak lengkap

1. Mengatur AWS CLI. Untuk petunjuknya, lihat Mengembangkan dengan Amazon S3 menggunakan AWS CLI di Referensi API Amazon S3.
2. Simpan konfigurasi siklus hidup contoh berikut dalam sebuah file (misalnya, *`lifecycle.json`*). Konfigurasi contoh ini menentukan prefiks kosong, dan karenanya berlaku untuk semua objek di bucket. Untuk membatasi konfigurasi ke subset objek, Anda dapat menentukan prefiks.

```
{
    "Rules": [
        {
            "ID": "Test Rule",
            "Status": "Enabled",
            "Filter": {
                "Prefix": ""
            },
            "AbortIncompleteMultipartUpload": {
                "DaysAfterInitiation": 7
            }
        }
    ]
```

```
}
```

3. Jalankan perintah CLI berikut ini untuk menetapkan konfigurasi siklus hidup pada bucket Anda.

```
aws s3api put-bucket-lifecycle-configuration   \
--bucket amzn-s3-demo-bucket  \
--lifecycle-configuration file://lifecycle.json
```

4. Untuk memverifikasi bahwa konfigurasi siklus hidup telah ditetapkan pada bucket Anda, ambil konfigurasi siklus hidup dengan menggunakan perintah `get-bucket-lifecycle` berikut.

```
aws s3api get-bucket-lifecycle  \
--bucket amzn-s3-demo-bucket
```

5. Untuk menghapus konfigurasi siklus hidup, gunakan perintah `delete-bucket-lifecycle` berikut ini.

```
aws s3api delete-bucket-lifecycle \
--bucket amzn-s3-demo-bucket
```

Pengunggahan objek menggunakan unggahan multibagian

Anda dapat menggunakan unggahan multibagian untuk mengunggah satu objek ke Amazon S3 secara terprogram. Setiap objek diunggah sebagai satu set bagian. Setiap bagian merupakan bagian data objek yang saling berkaitan. Anda dapat mengunggah bagian-bagian objek tersebut secara independen dan dengan urutan apa pun. Jika ada transmisi bagian mana pun yang gagal, Anda dapat mentransmisikan ulang bagian tersebut tanpa memengaruhi bagian lainnya. Setelah semua bagian objek Anda diunggah, Amazon S3 merakit bagian-bagian ini dan menciptakan objek. Pengguna anonim tidak dapat memulai unggahan multipart.

Untuk end-to-end prosedur mengunggah objek dengan unggahan multipart dengan checksum tambahan, lihat. Tutorial: Unggah objek melalui unggahan multipart dan verifikasi integritas datanya

Bagian berikut menunjukkan cara menggunakan upload multipart dengan AWS Command Line Interface, dan AWS SDKs.

Menggunakan konsol S3

Anda dapat mengunggah jenis file apa pun—gambar, cadangan, data, film, dan sebagainya—ke dalam bucket S3. Ukuran maksimum file yang dapat Anda unggah menggunakan konsol Amazon S3

adalah 160 GB. Untuk mengunggah file yang lebih besar dari 160 GB, gunakan AWS Command Line Interface (AWS CLI) AWS SDKs, atau Amazon S3 REST API.

Untuk petunjuk tentang mengunggah objek melalui AWS Management Console, lihatMengunggah Objek.

Menggunakan AWS CLI

Berikut ini menjelaskan operasi Amazon S3 untuk unggahan multibagian menggunakan. AWS CLI

- Mulai Unggahan Multibagian
- Unggah Bagian
- Unggah Bagian (Salinan)
- Selesaikan Unggahan Multibagian
- Batalkan Unggahan Multibagian
- Daftarkan Bagian
- Daftarkan Unggahan Multibagian

Penggunaan API REST

Bagian berikut dalam Referensi API Amazon Simple Storage Service menjelaskan tentang API REST untuk unggahan multibagian.

- Mulai Unggahan Multibagian
- Unggah Bagian
- Selesaikan Unggahan Multibagian
- Hentikan Unggahan Multibagian
- Daftarkan Bagian
- Daftarkan Unggahan Multibagian

Menggunakan AWS SDKs (API tingkat tinggi)

Beberapa AWS SDKs mengekspos API tingkat tinggi yang menyederhanakan pengunggahan multibagian dengan menggabungkan berbagai operasi API yang diperlukan untuk menyelesaikan unggahan multibagian ke dalam satu operasi. Untuk informasi selengkapnya, lihat Mengunggah dan menyalin objek menggunakan unggahan multipart di Amazon S3.

Jika Anda perlu menjeda dan melanjutkan unggahan multipart, memvariasikan ukuran bagian selama pengunggahan, atau tidak mengetahui ukuran data sebelumnya, gunakan metode API tingkat rendah. Metode API tingkat rendah untuk unggahan multipart menawarkan fungsionalitas tambahan, untuk informasi selengkapnya, lihat. Menggunakan AWS SDKs (API tingkat rendah)

.NET

Untuk mengunggah file ke bucket S3, gunakan kelas `TransferUtility`. Saat mengunggah data dari sebuah file, Anda harus menyediakan nama kunci objeknya. Jika tidak, API akan menggunakan nama file untuk nama kunci. Saat mengunggah data dari stream, Anda harus memberikan nama kunci objek.

Untuk mengatur opsi unggahan tingkat lanjut—seperti ukuran bagian, jumlah thread saat mengunggah beberapa bagian secara bersamaan, metadata, kelas penyimpanan, atau ACL—gunakan kelas `TransferUtilityUploadRequest`.

> **Note**
>
> Saat Anda menggunakan stream untuk sumber data, kelas `TransferUtility` tidak mengunggah secara bersamaan.

Contoh C# berikut mengunggah file ke bucket Amazon S3 dalam beberapa bagian. Contoh ini menunjukkan cara menggunakan berbagai kelebihan muatan `TransferUtility.Upload` untuk mengunggah sebuah file. Setiap panggilan berurutan untuk unggahan menggantikan unggahan sebelumnya. Untuk informasi tentang menyiapkan dan menjalankan contoh kode, lihat Memulai SDK for .NET di AWSAWS SDK for .NET Developer Guide.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Transfer;
using System;
using System.IO;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class UploadFileMPUHighLevelAPITest
    {
        private const string bucketName = "*** provide bucket name ***";
```

```
        private const string keyName = "*** provide a name for the uploaded object
***";
        private const string filePath = "*** provide the full path name of the file
to upload ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            UploadFileAsync().Wait();
        }

        private static async Task UploadFileAsync()
        {
            try
            {
                var fileTransferUtility =
                    new TransferUtility(s3Client);

                // Option 1. Upload a file. The file name is used as the object key
name.
                await fileTransferUtility.UploadAsync(filePath, bucketName);
                Console.WriteLine("Upload 1 completed");

                // Option 2. Specify object key name explicitly.
                await fileTransferUtility.UploadAsync(filePath, bucketName,
keyName);
                Console.WriteLine("Upload 2 completed");

                // Option 3. Upload data from a type of System.IO.Stream.
                using (var fileToUpload =
                    new FileStream(filePath, FileMode.Open, FileAccess.Read))
                {
                    await fileTransferUtility.UploadAsync(fileToUpload,
                                                bucketName, keyName);
                }
                Console.WriteLine("Upload 3 completed");

                // Option 4. Specify advanced settings.
                var fileTransferUtilityRequest = new TransferUtilityUploadRequest
                {
```

```
                    BucketName = bucketName,
                    FilePath = filePath,
                    StorageClass = S3StorageClass.StandardInfrequentAccess,
                    PartSize = 6291456, // 6 MB.
                    Key = keyName,
                    CannedACL = S3CannedACL.PublicRead
                };
                fileTransferUtilityRequest.Metadata.Add("param1", "Value1");
                fileTransferUtilityRequest.Metadata.Add("param2", "Value2");

                await fileTransferUtility.UploadAsync(fileTransferUtilityRequest);
                Console.WriteLine("Upload 4 completed");
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }

        }
    }
}
```

JavaScript

Example

Unggah file besar.

```
import { S3Client } from "@aws-sdk/client-s3";
import { Upload } from "@aws-sdk/lib-storage";

import {
  ProgressBar,
  logger,
} from "@aws-doc-sdk-examples/lib/utils/util-log.js";

const twentyFiveMB = 25 * 1024 * 1024;
```

```
export const createString = (size = twentyFiveMB) => {
  return "x".repeat(size);
};

/**
 * Create a 25MB file and upload it in parts to the specified
 * Amazon S3 bucket.
 * @param {{ bucketName: string, key: string }}
 */
export const main = async ({ bucketName, key }) => {
  const str = createString();
  const buffer = Buffer.from(str, "utf8");
  const progressBar = new ProgressBar({
    description: `Uploading "${key}" to "${bucketName}"`,
    barLength: 30,
  });

  try {
    const upload = new Upload({
      client: new S3Client({}),
      params: {
        Bucket: bucketName,
        Key: key,
        Body: buffer,
      },
    });

    upload.on("httpUploadProgress", ({ loaded, total }) => {
      progressBar.update({ current: loaded, total });
    });

    await upload.done();
  } catch (caught) {
    if (caught instanceof Error && caught.name === "AbortError") {
      logger.error(`Multipart upload was aborted. ${caught.message}`);
    } else {
      throw caught;
    }
  }
};
```

Example

Unduh file besar.

```
import { fileURLToPath } from "node:url";
import { GetObjectCommand, NoSuchKey, S3Client } from "@aws-sdk/client-s3";
import { createWriteStream, rmSync } from "node:fs";

const s3Client = new S3Client({});
const oneMB = 1024 * 1024;

export const getObjectRange = ({ bucket, key, start, end }) => {
  const command = new GetObjectCommand({
    Bucket: bucket,
    Key: key,
    Range: `bytes=${start}-${end}`,
  });

  return s3Client.send(command);
};

/**
 * @param {string | undefined} contentRange
 */
export const getRangeAndLength = (contentRange) => {
  const [range, length] = contentRange.split("/");
  const [start, end] = range.split("-");
  return {
    start: Number.parseInt(start),
    end: Number.parseInt(end),
    length: Number.parseInt(length),
  };
};

export const isComplete = ({ end, length }) => end === length - 1;

const downloadInChunks = async ({ bucket, key }) => {
  const writeStream = createWriteStream(
    fileURLToPath(new URL(`./${key}`, import.meta.url)),
  ).on("error", (err) => console.error(err));

  let rangeAndLength = { start: -1, end: -1, length: -1 };

  while (!isComplete(rangeAndLength)) {
```

```
    const { end } = rangeAndLength;
    const nextRange = { start: end + 1, end: end + oneMB };

    const { ContentRange, Body } = await getObjectRange({
      bucket,
      key,
      ...nextRange,
    });
    console.log(`Downloaded bytes ${nextRange.start} to ${nextRange.end}`);

    writeStream.write(await Body.transformToByteArray());
    rangeAndLength = getRangeAndLength(ContentRange);
  }
};

/**
 * Download a large object from and Amazon S3 bucket.
 *
 * When downloading a large file, you might want to break it down into
 * smaller pieces. Amazon S3 accepts a Range header to specify the start
 * and end of the byte range to be downloaded.
 *
 * @param {{ bucketName: string, key: string }}
 */
export const main = async ({ bucketName, key }) => {
  try {
    await downloadInChunks({
      bucket: bucketName,
      key: key,
    });
  } catch (caught) {
    if (caught instanceof NoSuchKey) {
      console.error(`Failed to download object. No such key "${key}".`);
      rmSync(key);
    }
  }
};
```

Go

Untuk informasi selengkapnya tentang contoh kode Go untuk unggahan multibagian, lihat Mengunggah atau mengunduh file besar ke dan dari Amazon S3 menggunakan AWS SDK.

Example

Unggah objek besar dengan menggunakan pengelola unggahan untuk memecah data menjadi beberapa bagian dan mengunggahnya secara bersamaan.

```
import (
 "bytes"
 "context"
 "errors"
 "fmt"
 "io"
 "log"
 "os"
 "time"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/feature/s3/manager"
 "github.com/aws/aws-sdk-go-v2/service/s3"
 "github.com/aws/aws-sdk-go-v2/service/s3/types"
 "github.com/aws/smithy-go"
)

// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3) actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform bucket
// and object actions.
type BucketBasics struct {
 S3Client *s3.Client
}
```

```
// UploadLargeObject uses an upload manager to upload data to an object in a bucket.
// The upload manager breaks large data into parts and uploads the parts
 concurrently.
func (basics BucketBasics) UploadLargeObject(ctx context.Context, bucketName string,
 objectKey string, largeObject []byte) error {
 largeBuffer := bytes.NewReader(largeObject)
 var partMiBs int64 = 10
 uploader := manager.NewUploader(basics.S3Client, func(u *manager.Uploader) {
  u.PartSize = partMiBs * 1024 * 1024
 })
```

```
 _, err := uploader.Upload(ctx, &s3.PutObjectInput{
  Bucket: aws.String(bucketName),
  Key:    aws.String(objectKey),
  Body:   largeBuffer,
 })
 if err != nil {
  var apiErr smithy.APIError
  if errors.As(err, &apiErr) && apiErr.ErrorCode() == "EntityTooLarge" {
   log.Printf("Error while uploading object to %s. The object is too large.\n"+
    "The maximum size for a multipart upload is 5TB.", bucketName)
  } else {
   log.Printf("Couldn't upload large object to %v:%v. Here's why: %v\n",
    bucketName, objectKey, err)
  }
 } else {
  err = s3.NewObjectExistsWaiter(basics.S3Client).Wait(
   ctx, &s3.HeadObjectInput{Bucket: aws.String(bucketName), Key:
 aws.String(objectKey)}, time.Minute)
  if err != nil {
   log.Printf("Failed attempt to wait for object %s to exist.\n", objectKey)
  }
 }

 return err
}
```

Example

Unduh objek besar dengan menggunakan pengelola unduhan untuk mendapatkan data dalam beberapa bagian dan mengunduhnya secara bersamaan.

```
// DownloadLargeObject uses a download manager to download an object from a bucket.
// The download manager gets the data in parts and writes them to a buffer until all
 of
// the data has been downloaded.
func (basics BucketBasics) DownloadLargeObject(ctx context.Context, bucketName
 string, objectKey string) ([]byte, error) {
 var partMiBs int64 = 10
 downloader := manager.NewDownloader(basics.S3Client, func(d *manager.Downloader) {
  d.PartSize = partMiBs * 1024 * 1024
 })
```

```
 buffer := manager.NewWriteAtBuffer([]byte{})
 _, err := downloader.Download(ctx, buffer, &s3.GetObjectInput{
  Bucket: aws.String(bucketName),
  Key:    aws.String(objectKey),
 })
 if err != nil {
  log.Printf("Couldn't download large object from %v:%v. Here's why: %v\n",
   bucketName, objectKey, err)
 }
 return buffer.Bytes(), err
}
```

PHP

Topik ini menjelaskan cara menggunakan `Aws\S3\Model\MultipartUpload\UploadBuilder` kelas tingkat tinggi dari AWS SDK untuk PHP untuk upload file multipart. Untuk informasi selengkapnya tentang AWS SDK for Ruby API, AWS buka SDK for Ruby - Versi 2.

Contoh PHP berikut mengunggah file ke bucket Amazon S3. Contoh ini menunjukkan cara menetukan parameter untuk objek `MultipartUploader`.

```
 require 'vendor/autoload.php';

use Aws\Exception\MultipartUploadException;
use Aws\S3\MultipartUploader;
use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// Prepare the upload parameters.
$uploader = new MultipartUploader($s3, '/path/to/large/file.zip', [
    'bucket' => $bucket,
    'key'    => $keyname
]);
```

```
// Perform the upload.
try {
    $result = $uploader->upload();
    echo "Upload complete: {$result['ObjectURL']}" . PHP_EOL;
} catch (MultipartUploadException $e) {
    echo $e->getMessage() . PHP_EOL;
}
```

Python

Contoh berikut memuat objek menggunakan API Python unggahan multibagian tingkat tinggi (kelas `TransferManager`).

```
import sys
import threading

import boto3
from boto3.s3.transfer import TransferConfig


MB = 1024 * 1024
s3 = boto3.resource("s3")


class TransferCallback:
    """
    Handle callbacks from the transfer manager.

    The transfer manager periodically calls the __call__ method throughout
    the upload and download process so that it can take action, such as
    displaying progress to the user and collecting data about the transfer.
    """

    def __init__(self, target_size):
        self._target_size = target_size
        self._total_transferred = 0
        self._lock = threading.Lock()
        self.thread_info = {}

    def __call__(self, bytes_transferred):
        """
        The callback method that is called by the transfer manager.
```

```
        Display progress during file transfer and collect per-thread transfer
        data. This method can be called by multiple threads, so shared instance
        data is protected by a thread lock.
        """
        thread = threading.current_thread()
        with self._lock:
            self._total_transferred += bytes_transferred
            if thread.ident not in self.thread_info.keys():
                self.thread_info[thread.ident] = bytes_transferred
            else:
                self.thread_info[thread.ident] += bytes_transferred

            target = self._target_size * MB
            sys.stdout.write(
                f"\r{self._total_transferred} of {target} transferred "
                f"({(self._total_transferred / target) * 100:.2f}%)."
            )
            sys.stdout.flush()


def upload_with_default_configuration(
    local_file_path, bucket_name, object_key, file_size_mb
):
    """
    Upload a file from a local folder to an Amazon S3 bucket, using the default
    configuration.
    """
    transfer_callback = TransferCallback(file_size_mb)
    s3.Bucket(bucket_name).upload_file(
        local_file_path, object_key, Callback=transfer_callback
    )
    return transfer_callback.thread_info


def upload_with_chunksize_and_meta(
    local_file_path, bucket_name, object_key, file_size_mb, metadata=None
):
    """
    Upload a file from a local folder to an Amazon S3 bucket, setting a
    multipart chunk size and adding metadata to the Amazon S3 object.

    The multipart chunk size controls the size of the chunks of data that are
    sent in the request. A smaller chunk size typically results in the transfer
```

```
    manager using more threads for the upload.

    The metadata is a set of key-value pairs that are stored with the object
    in Amazon S3.
    """
    transfer_callback = TransferCallback(file_size_mb)

    config = TransferConfig(multipart_chunksize=1 * MB)
    extra_args = {"Metadata": metadata} if metadata else None
    s3.Bucket(bucket_name).upload_file(
        local_file_path,
        object_key,
        Config=config,
        ExtraArgs=extra_args,
        Callback=transfer_callback,
    )
    return transfer_callback.thread_info


def upload_with_high_threshold(local_file_path, bucket_name, object_key,
 file_size_mb):
    """
    Upload a file from a local folder to an Amazon S3 bucket, setting a
    multipart threshold larger than the size of the file.

    Setting a multipart threshold larger than the size of the file results
    in the transfer manager sending the file as a standard upload instead of
    a multipart upload.
    """
    transfer_callback = TransferCallback(file_size_mb)
    config = TransferConfig(multipart_threshold=file_size_mb * 2 * MB)
    s3.Bucket(bucket_name).upload_file(
        local_file_path, object_key, Config=config, Callback=transfer_callback
    )
    return transfer_callback.thread_info


def upload_with_sse(
    local_file_path, bucket_name, object_key, file_size_mb, sse_key=None
):
    """
    Upload a file from a local folder to an Amazon S3 bucket, adding server-side
    encryption with customer-provided encryption keys to the object.
```

```
    When this kind of encryption is specified, Amazon S3 encrypts the object
    at rest and allows downloads only when the expected encryption key is
    provided in the download request.
    """
    transfer_callback = TransferCallback(file_size_mb)
    if sse_key:
        extra_args = {"SSECustomerAlgorithm": "AES256", "SSECustomerKey": sse_key}
    else:
        extra_args = None
    s3.Bucket(bucket_name).upload_file(
        local_file_path, object_key, ExtraArgs=extra_args,
 Callback=transfer_callback
    )
    return transfer_callback.thread_info


def download_with_default_configuration(
    bucket_name, object_key, download_file_path, file_size_mb
):
    """
    Download a file from an Amazon S3 bucket to a local folder, using the
    default configuration.
    """
    transfer_callback = TransferCallback(file_size_mb)
    s3.Bucket(bucket_name).Object(object_key).download_file(
        download_file_path, Callback=transfer_callback
    )
    return transfer_callback.thread_info


def download_with_single_thread(
    bucket_name, object_key, download_file_path, file_size_mb
):
    """
    Download a file from an Amazon S3 bucket to a local folder, using a
    single thread.
    """
    transfer_callback = TransferCallback(file_size_mb)
    config = TransferConfig(use_threads=False)
    s3.Bucket(bucket_name).Object(object_key).download_file(
        download_file_path, Config=config, Callback=transfer_callback
    )
    return transfer_callback.thread_info
```

```
def download_with_high_threshold(
    bucket_name, object_key, download_file_path, file_size_mb
):
    """
    Download a file from an Amazon S3 bucket to a local folder, setting a
    multipart threshold larger than the size of the file.

    Setting a multipart threshold larger than the size of the file results
    in the transfer manager sending the file as a standard download instead
    of a multipart download.
    """
    transfer_callback = TransferCallback(file_size_mb)
    config = TransferConfig(multipart_threshold=file_size_mb * 2 * MB)
    s3.Bucket(bucket_name).Object(object_key).download_file(
        download_file_path, Config=config, Callback=transfer_callback
    )
    return transfer_callback.thread_info


def download_with_sse(
    bucket_name, object_key, download_file_path, file_size_mb, sse_key
):
    """
    Download a file from an Amazon S3 bucket to a local folder, adding a
    customer-provided encryption key to the request.

    When this kind of encryption is specified, Amazon S3 encrypts the object
    at rest and allows downloads only when the expected encryption key is
    provided in the download request.
    """
    transfer_callback = TransferCallback(file_size_mb)

    if sse_key:
        extra_args = {"SSECustomerAlgorithm": "AES256", "SSECustomerKey": sse_key}
    else:
        extra_args = None
    s3.Bucket(bucket_name).Object(object_key).download_file(
        download_file_path, ExtraArgs=extra_args, Callback=transfer_callback
    )
    return transfer_callback.thread_info
```

Menggunakan AWS SDKs (API tingkat rendah)

AWS SDK mengekspos API tingkat rendah yang sangat mirip dengan Amazon S3 REST API untuk unggahan multipart (lihat. Mengunggah dan menyalin objek menggunakan unggahan multipart di Amazon S3 Gunakan API tingkat rendah saat Anda perlu menjeda dan melanjutkan unggahan multibagian, memvariasikan ukuran bagian selama pengunggahan, atau tidak mengetahui ukuran data unggahan sebelumnya. Jika Anda tidak memiliki persyaratan ini, gunakan API tingkat tinggi (lihatMenggunakan AWS SDKs (API tingkat tinggi)).

Java

Contoh berikut menunjukkan cara menggunakan kelas Java tingkat rendah untuk mengunggah file. Ini melakukan langkah-langkah berikut:

- Memulai unggahan multibagian menggunakan metode `AmazonS3Client.initiateMultipartUpload()`, dan melalui objek `InitiateMultipartUploadRequest`.
- Menyimpan ID unggahan yang dikembalikan oleh metode `AmazonS3Client.initiateMultipartUpload()`. Anda dapat menyediakan ID unggahan untuk setiap operasi unggahan multibagian selanjutnya.
- Mengunggah bagian objek. Untuk setiap bagian, Anda memanggil metode `AmazonS3Client.uploadPart()`. Anda memberikan bagian informasi unggahan menggunakan objek `UploadPartRequest`.
- Untuk setiap bagian, simpan ETag dari respons `AmazonS3Client.uploadPart()` metode dalam daftar. Anda dapat menggunakan nilai ETag untuk menyelesaikan unggahan multipart.
- Memanggil metode `AmazonS3Client.completeMultipartUpload()` untuk menyelesaikan unggahan multibagian.

Example

Untuk petunjuk cara membuat dan menguji sampel yang berfungsi, lihat Memulai di Panduan AWS SDK untuk Java Pengembang.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
```

```
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.io.File;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

public class LowLevelMultipartUpload {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String keyName = "*** Key name ***";
        String filePath = "*** Path to file to upload ***";

        File file = new File(filePath);
        long contentLength = file.length();
        long partSize = 5 * 1024 * 1024; // Set part size to 5 MB.

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withRegion(clientRegion)
                    .withCredentials(new ProfileCredentialsProvider())
                    .build();

            // Create a list of ETag objects. You retrieve ETags for each object
 part
            // uploaded,
            // then, after each individual part has been uploaded, pass the list of
 ETags to
            // the request to complete the upload.
            List<PartETag> partETags = new ArrayList<PartETag>();

            // Initiate the multipart upload.
            InitiateMultipartUploadRequest initRequest = new
 InitiateMultipartUploadRequest(bucketName, keyName);
            InitiateMultipartUploadResult initResponse =
 s3Client.initiateMultipartUpload(initRequest);

            // Upload the file parts.
            long filePosition = 0;
            for (int i = 1; filePosition < contentLength; i++) {
```

```
                // Because the last part could be less than 5 MB, adjust the part
 size as
                // needed.
                partSize = Math.min(partSize, (contentLength - filePosition));

                // Create the request to upload a part.
                UploadPartRequest uploadRequest = new UploadPartRequest()
                        .withBucketName(bucketName)
                        .withKey(keyName)
                        .withUploadId(initResponse.getUploadId())
                        .withPartNumber(i)
                        .withFileOffset(filePosition)
                        .withFile(file)
                        .withPartSize(partSize);

                // Upload the part and add the response's ETag to our list.
                UploadPartResult uploadResult = s3Client.uploadPart(uploadRequest);
                partETags.add(uploadResult.getPartETag());

                filePosition += partSize;
            }

            // Complete the multipart upload.
            CompleteMultipartUploadRequest compRequest = new
 CompleteMultipartUploadRequest(bucketName, keyName,
                    initResponse.getUploadId(), partETags);
            s3Client.completeMultipartUpload(compRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

Contoh C# berikut menunjukkan cara menggunakan API unggahan SDK untuk .NET multibagian tingkat rendah untuk mengunggah file ke bucket S3. Untuk informasi tentang unggahan

multibagian Amazon S3, lihat Mengunggah dan menyalin objek menggunakan unggahan multipart di Amazon S3.

> **Note**
>
> Saat Anda menggunakan SDK untuk .NET API untuk mengunggah objek besar, batas waktu mungkin terjadi saat data sedang ditulis ke aliran permintaan. Anda dapat mengatur waktu habis yang eksplisit menggunakan `UploadPartRequest`.

Contoh C# berikut mengunggah file ke bucket S3 menggunakan API unggahan multibagian tingkat rendah. Untuk informasi tentang menyiapkan dan menjalankan contoh kode, lihat Memulai SDK for .NET di AWSAWS SDK for .NET Developer Guide.

```
using Amazon;
using Amazon.Runtime;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.IO;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class UploadFileMPULowLevelAPITest
    {
        private const string bucketName = "*** provide bucket name ***";
        private const string keyName = "*** provide a name for the uploaded object
 ***";
        private const string filePath = "*** provide the full path name of the file
 to upload ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            Console.WriteLine("Uploading an object");
            UploadObjectAsync().Wait();
```

```
        }

        private static async Task UploadObjectAsync()
        {
            // Create list to store upload part responses.
            List<UploadPartResponse> uploadResponses = new
List<UploadPartResponse>();

            // Setup information required to initiate the multipart upload.
            InitiateMultipartUploadRequest initiateRequest = new
InitiateMultipartUploadRequest
            {
                BucketName = bucketName,
                Key = keyName
            };

            // Initiate the upload.
            InitiateMultipartUploadResponse initResponse =
                await s3Client.InitiateMultipartUploadAsync(initiateRequest);

            // Upload parts.
            long contentLength = new FileInfo(filePath).Length;
            long partSize = 5 * (long)Math.Pow(2, 20); // 5 MB

            try
            {
                Console.WriteLine("Uploading parts");

                long filePosition = 0;
                for (int i = 1; filePosition < contentLength; i++)
                {
                    UploadPartRequest uploadRequest = new UploadPartRequest
                        {
                            BucketName = bucketName,
                            Key = keyName,
                            UploadId = initResponse.UploadId,
                            PartNumber = i,
                            PartSize = partSize,
                            FilePosition = filePosition,
                            FilePath = filePath
                        };

                    // Track upload progress.
                    uploadRequest.StreamTransferProgress +=
```

```
                        new
EventHandler<StreamTransferProgressArgs>(UploadPartProgressEventCallback);

                    // Upload a part and add the response to our list.
                    uploadResponses.Add(await
s3Client.UploadPartAsync(uploadRequest));

                    filePosition += partSize;
                }

                // Setup to complete the upload.
                CompleteMultipartUploadRequest completeRequest = new
CompleteMultipartUploadRequest
                    {
                        BucketName = bucketName,
                        Key = keyName,
                        UploadId = initResponse.UploadId
                     };
                completeRequest.AddPartETags(uploadResponses);

                // Complete the upload.
                CompleteMultipartUploadResponse completeUploadResponse =
                    await s3Client.CompleteMultipartUploadAsync(completeRequest);
            }
            catch (Exception exception)
            {
                Console.WriteLine("An AmazonS3Exception was thrown: { 0}",
exception.Message);

                // Abort the upload.
                AbortMultipartUploadRequest abortMPURequest = new
AbortMultipartUploadRequest
                {
                    BucketName = bucketName,
                    Key = keyName,
                    UploadId = initResponse.UploadId
                };
               await s3Client.AbortMultipartUploadAsync(abortMPURequest);
            }
        }
        public static void UploadPartProgressEventCallback(object sender,
StreamTransferProgressArgs e)
        {
            // Process event.
```

```
            Console.WriteLine("{0}/{1}", e.TransferredBytes, e.TotalBytes);
        }
    }
}
```

PHP

Topik ini menunjukkan cara menggunakan `uploadPart` metode tingkat rendah dari versi 3 AWS SDK untuk PHP untuk mengunggah file di beberapa bagian. Untuk informasi selengkapnya tentang AWS SDK for Ruby API, AWS buka SDK for Ruby - Versi 2.

Contoh PHP berikut mengunggah file ke bucket Amazon S3 menggunakan unggahan multipart PHP API tingkat rendah.

```
 require 'vendor/autoload.php';

use Aws\S3\Exception\S3Exception;
use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';
$filename = '*** Path to and Name of the File to Upload ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

$result = $s3->createMultipartUpload([
    'Bucket'       => $bucket,
    'Key'          => $keyname,
    'StorageClass' => 'REDUCED_REDUNDANCY',
    'Metadata'     => [
        'param1' => 'value 1',
        'param2' => 'value 2',
        'param3' => 'value 3'
    ]
]);
$uploadId = $result['UploadId'];

// Upload the file in parts.
try {
    $file = fopen($filename, 'r');
```

```
    $partNumber = 1;
    while (!feof($file)) {
        $result = $s3->uploadPart([
            'Bucket'     => $bucket,
            'Key'        => $keyname,
            'UploadId'   => $uploadId,
            'PartNumber' => $partNumber,
            'Body'       => fread($file, 5 * 1024 * 1024),
        ]);
        $parts['Parts'][$partNumber] = [
            'PartNumber' => $partNumber,
            'ETag' => $result['ETag'],
        ];
        $partNumber++;

        echo "Uploading part $partNumber of $filename." . PHP_EOL;
    }
    fclose($file);
} catch (S3Exception $e) {
    $result = $s3->abortMultipartUpload([
        'Bucket'   => $bucket,
        'Key'      => $keyname,
        'UploadId' => $uploadId
    ]);

    echo "Upload of $filename failed." . PHP_EOL;
}

// Complete the multipart upload.
$result = $s3->completeMultipartUpload([
    'Bucket'   => $bucket,
    'Key'      => $keyname,
    'UploadId' => $uploadId,
    'MultipartUpload'    => $parts,
]);
$url = $result['Location'];

echo "Uploaded $filename to $url." . PHP_EOL;
```

Menggunakan AWS SDK untuk Ruby

AWS SDK untuk Ruby Versi 3 mendukung unggahan multipart Amazon S3 dengan dua cara. Untuk opsi pertama, Anda dapat menggunakan unggahan file terkelola. Untuk informasi lebih lanjut, lihat Mengunggah File ke Amazon S3 dalam AWS Blog Pengembang. Unggahan file terkelola adalah metode yang direkomendasikan untuk mengunggah file ke bucket. Mereka memberikan manfaat berikut:

- Mengelola unggahan multibagian untuk objek yang lebih besar dari 15 MB.
- Membuka file dengan benar dalam mode biner untuk menghindari masalah pengodean.
- Menggunakan beberapa thread untuk mengunggah bagian-bagian objek besar secara paralel.

Alternatifnya, Anda dapat menggunakan operasi klien unggahan multibagian berikut secara langsung:

- create_multipart_upload–Memulai sebuah unggahan multibagian dan menampilkan sebuah ID unggahan.
- upload_part–Mengunggah satu bagian dalam sebuah unggahan multibagian.
- upload_part_copy–Mengunggah satu bagian dengan menyalin data dari objek yang ada sebagai sumber data.
- complete_multipart_upload–Menyelesaikan sebuah unggahan multibagian dengan merangkai bagian-bagian yang diunggah sebelumnya.
- abort_multipart_upload–Menghentikan unggahan multibagian.

Pengunggahan sebuah direktori menggunakan kelas TransferUtility .NET tingkat tinggi

Anda dapat menggunakan kelas `TransferUtility` untuk mengunggah sebuah direktori secara keseluruhan. Secara default, API hanya mengunggah file di dalam root direktori yang ditentukan. Namun, Anda dapat menentukan pengunggahan file secara rekursif di semua sub direktori.

Untuk memilih file di direktori tertentu berdasarkan kriteria pemfilteran, tentukan ekspresi pemfilterannya. Misalnya, untuk mengunggah hanya PDF file dari direktori, tentukan ekspresi `"*.pdf"` filter.

Saat mengunggah file dari sebuah direktori, Anda tidak perlu menentukan nama kunci untuk objek yang dihasilkan. Amazon S3 membangun nama kunci menggunakan jalur file asli. Misalnya,

asumsikan bahwa Anda memiliki sebuah direktori yang disebut sebagai `c:\myfolder` dengan struktur berikut:

Example

```
C:\myfolder
      \a.txt
      \b.pdf
      \media\
             An.mp3
```

Saat Anda mengunggah direktori ini, Amazon S3 menggunakan nama kunci berikut:

Example

```
a.txt
b.pdf
media/An.mp3
```

Example

Contoh C# berikut mengunggah sebuah direktori ke bucket Amazon S3. Contoh ini menunjukkan cara menggunakan berbagai kelebihan muatan `TransferUtility.UploadDirectory` untuk mengunggah direktori. Setiap panggilan berurutan untuk unggahan menggantikan unggahan sebelumnya. Untuk informasi tentang menyiapkan dan menjalankan contoh kode, lihat Memulai SDK for .NET di AWSAWS SDK for .NET Developer Guide.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Transfer;
using System;
using System.IO;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class UploadDirMPUHighLevelAPITest
    {
        private const string existingBucketName = "*** bucket name ***";
        private const string directoryPath = @"*** directory path ***";
        // The example uploads only .txt files.
        private const string wildCard = "*.txt";
```

```
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
        static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            UploadDirAsync().Wait();
        }

        private static async Task UploadDirAsync()
        {
            try
            {
                var directoryTransferUtility =
                    new TransferUtility(s3Client);

                // 1. Upload a directory.
                await directoryTransferUtility.UploadDirectoryAsync(directoryPath,
                    existingBucketName);
                Console.WriteLine("Upload statement 1 completed");

                // 2. Upload only the .txt files from a directory
                //    and search recursively.
                await directoryTransferUtility.UploadDirectoryAsync(
                                                directoryPath,
                                                existingBucketName,
                                                wildCard,
                                                SearchOption.AllDirectories);
                Console.WriteLine("Upload statement 2 completed");

                // 3. The same as Step 2 and some optional configuration.
                //    Search recursively for .txt files to upload.
                var request = new TransferUtilityUploadDirectoryRequest
                {
                    BucketName = existingBucketName,
                    Directory = directoryPath,
                    SearchOption = SearchOption.AllDirectories,
                    SearchPattern = wildCard
                };

                await directoryTransferUtility.UploadDirectoryAsync(request);
                Console.WriteLine("Upload statement 3 completed");
            }
            catch (AmazonS3Exception e)
```

```
            {
                Console.WriteLine(
                        "Error encountered ***. Message:'{0}' when writing an object",
 e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine(
                    "Unknown encountered on server. Message:'{0}' when writing an
 object", e.Message);
            }
        }
    }
}
```

Mendaftarkan unggahan multibagian

Anda dapat menggunakan AWS CLI, REST API, atau AWS SDKs, untuk mengambil daftar unggahan multipart yang sedang berlangsung di Amazon S3. Anda dapat menggunakan unggahan multibagian untuk mengunggah satu objek ke Amazon S3 secara terprogram. Unggahan multipart memindahkan objek ke Amazon S3 dengan memindahkan sebagian data objek sekaligus. Untuk informasi lebih umum tentang unggahan multipart, lihat. Mengunggah dan menyalin objek menggunakan unggahan multipart di Amazon S3

Untuk end-to-end prosedur mengunggah objek dengan unggahan multipart dengan checksum tambahan, lihat. Tutorial: Unggah objek melalui unggahan multipart dan verifikasi integritas datanya

Bagian berikut menunjukkan cara membuat daftar unggahan multipart yang sedang berlangsung dengan AWS Command Line Interface, Amazon S3 REST API, dan. AWS SDKs

Membuat daftar unggahan multipart menggunakan AWS CLI

Bagian berikut dalam AWS Command Line Interface menjelaskan operasi untuk daftar unggahan multipart.

- list-parts-mendaftarkan bagian yang terunggah untuk unggahan multibagian tertentu.
- list-multipart-uploads-daftar unggahan multipart yang sedang berlangsung.

Mendaftarkan API REST untuk unggahan multibagian

Bagian dalam Referensi API Amazon Simple Storage Service berikut ini menjelaskan tentang API REST untuk unggahan multibagian:

- ListParts-daftar bagian yang diunggah untuk unggahan multibagian tertentu.
- ListMultipartUploads-daftar unggahan multipart yang sedang berlangsung.

Mencantumkan unggahan multibagian menggunakan AWS SDK (API tingkat rendah)

Java

Tugas berikut memandu Anda dalam menggunakan kelas Java tingkat rendah untuk mencantumkan semua unggahan multibagian yang sedang berlangsung di dalam sebuah bucket.

Proses pendaftaran unggahan multibagian API tingkat rendah

1	Membuat sebuah instans dari kelas `ListMultipartUploadsRequest` dan memberikan nama bucket.
2	Jalankan metode `AmazonS3Client.listMultipartUploads`. Metode tersebut akan menampilkan sebuah instans dari kelas `MultipartUploadListing` yang menyediakan informasi tentang unggahan multibagian yang sedang berlangsung.

Contoh kode Java berikut menunjukkan tugas sebelumnya.

Example

```
ListMultipartUploadsRequest allMultpartUploadsRequest =
     new ListMultipartUploadsRequest(existingBucketName);
MultipartUploadListing multipartUploadListing =
     s3Client.listMultipartUploads(allMultpartUploadsRequest);
```

.NET

Untuk mendaftarkan semua unggahan multibagian yang sedang berlangsung pada bucket tertentu, gunakan kelas SDK untuk .NET API unggahan multibagian tingkat rendah

`ListMultipartUploadsRequest`. Metode `AmazonS3Client.ListMultipartUploads` akan mengembalikan instans dari kelas `ListMultipartUploadsResponse` yang memberikan informasi tentang unggahan multibagian yang sedang berlangsung.

Unggahan multibagian yang sedang berlangsung adalah unggahan multibagian yang telah dimulai menggunakan permintaan unggahan multibagian, namun belum diselesaikan atau dihentikan. Untuk informasi selengkapnya tentang unggahan multipart Amazon S3, lihat Mengunggah dan menyalin objek menggunakan unggahan multipart di Amazon S3.

Contoh C# berikut menunjukkan cara menggunakan daftar semua SDK untuk .NET unggahan multipart yang sedang berlangsung di bucket. Untuk informasi tentang menyiapkan dan menjalankan contoh kode, lihat Memulai SDK for .NET di AWSAWS SDK for .NET Developer Guide.

```
ListMultipartUploadsRequest request = new ListMultipartUploadsRequest
{
  BucketName = bucketName // Bucket receiving the uploads.
};

ListMultipartUploadsResponse response = await
 AmazonS3Client.ListMultipartUploadsAsync(request);
```

PHP

Topik ini menunjukkan cara menggunakan class API tingkat rendah dari versi 3 AWS SDK untuk PHP untuk mencantumkan semua unggahan multibagian yang sedang berlangsung di bucket. Untuk informasi selengkapnya tentang AWS SDK for Ruby API, AWS buka SDK for Ruby - Versi 2.

Contoh PHP berikut mendemonstrasikan pendaftaran semua unggahan multibagian yang sedang berlangsung pada sebuah bucket.

```
 require 'vendor/autoload.php';

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';

$s3 = new S3Client([
    'version' => 'latest',
```

```
    'region'  => 'us-east-1'
]);

// Retrieve a list of the current multipart uploads.
$result = $s3->listMultipartUploads([
    'Bucket' => $bucket
]);

// Write the list of uploads to the page.
print_r($result->toArray());
```

Melacak unggahan multipart dengan AWS SDKs

Anda dapat melacak kemajuan unggahan objek ke Amazon S3 dengan antarmuka mendengarkan. API unggahan multibagian tingkat tinggi menyediakan antarmuka mendengarkan seperti itu, yang disebut. `ProgressListener` Peristiwa progres secara berkala memberi tahu pendengar bahwa byte telah ditransfer. Untuk informasi lebih umum tentang unggahan multipart, lihat. Mengunggah dan menyalin objek menggunakan unggahan multipart di Amazon S3

Untuk end-to-end prosedur mengunggah objek dengan unggahan multipart dengan checksum tambahan, lihat. Tutorial: Unggah objek melalui unggahan multipart dan verifikasi integritas datanya

Bagian berikut menunjukkan cara melacak unggahan multipart dengan file. AWS SDKs

Java

Example

```
TransferManager tm = new TransferManager(new ProfileCredentialsProvider());

PutObjectRequest request = new PutObjectRequest(
    existingBucketName, keyName, new File(filePath));

// Subscribe to the event and provide event handler.
request.setProgressListener(new ProgressListener() {
   public void progressChanged(ProgressEvent event) {
    System.out.println("Transferred bytes: " +
      event.getBytesTransfered());
            }
});
```

Example

Kode Java berikut mengunggah file dan menggunakan `ProgressListener` untuk melacak kemajuan unggahan. Untuk petunjuk tentang cara membuat dan menguji sampel kerja, lihat Memulai di Panduan AWS SDK untuk Java Pengembang.

```
import java.io.File;

import com.amazonaws.AmazonClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.event.ProgressEvent;
import com.amazonaws.event.ProgressListener;
import com.amazonaws.services.s3.model.PutObjectRequest;
import com.amazonaws.services.s3.transfer.TransferManager;
import com.amazonaws.services.s3.transfer.Upload;

public class TrackMPUProgressUsingHighLevelAPI {

    public static void main(String[] args) throws Exception {
        String existingBucketName = "*** Provide bucket name ***";
        String keyName            = "*** Provide object key ***";
        String filePath           = "*** file to upload ***";

        TransferManager tm = new TransferManager(new ProfileCredentialsProvider());


        // For more advanced uploads, you can create a request object
        // and supply additional request parameters (ex: progress listeners,
        // canned ACLs, etc.)
        PutObjectRequest request = new PutObjectRequest(
          existingBucketName, keyName, new File(filePath));

        // You can ask the upload for its progress, or you can
        // add a ProgressListener to your request to receive notifications
        // when bytes are transferred.
        request.setGeneralProgressListener(new ProgressListener() {
   @Override
   public void progressChanged(ProgressEvent progressEvent) {
    System.out.println("Transferred bytes: " +
      progressEvent.getBytesTransferred());
   }
  });
```

```
        // TransferManager processes all transfers asynchronously,
        // so this call will return immediately.
        Upload upload = tm.upload(request);

        try {
         // You can block and wait for the upload to finish
         upload.waitForCompletion();
        } catch (AmazonClientException amazonClientException) {
         System.out.println("Unable to upload file, upload aborted.");
         amazonClientException.printStackTrace();
        }
    }
}
```

.NET

Contoh C# berikut ini menampilkan mengunggah sebuah file ke sebuah bucket S3 menggunakan kelas `TransferUtility`, dan melacak progres unggahan. Untuk informasi tentang menyiapkan dan menjalankan contoh kode, lihat Memulai SDK for .NET di AWSAWS SDK for .NET Developer Guide.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Transfer;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class TrackMPUUsingHighLevelAPITest
    {
        private const string bucketName = "*** provide the bucket name ***";
        private const string keyName = "*** provide the name for the uploaded object
 ***";
        private const string filePath = " *** provide the full path name of the file
 to upload **";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;


        public static void Main()
```

```
        {
            s3Client = new AmazonS3Client(bucketRegion);
            TrackMPUAsync().Wait();
        }

        private static async Task TrackMPUAsync()
        {
            try
            {
                var fileTransferUtility = new TransferUtility(s3Client);

                // Use TransferUtilityUploadRequest to configure options.
                // In this example we subscribe to an event.
                var uploadRequest =
                    new TransferUtilityUploadRequest
                    {
                        BucketName = bucketName,
                        FilePath = filePath,
                        Key = keyName
                    };

                uploadRequest.UploadProgressEvent +=
                    new EventHandler<UploadProgressArgs>
                        (uploadRequest_UploadPartProgressEvent);

                await fileTransferUtility.UploadAsync(uploadRequest);
                Console.WriteLine("Upload completed");
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
        }

        static void uploadRequest_UploadPartProgressEvent(object sender,
UploadProgressArgs e)
        {
            // Process event.
```

```
            Console.WriteLine("{0}/{1}", e.TransferredBytes, e.TotalBytes);
        }
    }
}
```

Membatalkan unggahan multibagian

Setelah memulai unggahan multibagian, Anda memulai pengunggahan bagian. Amazon S3 menyimpan bagian-bagian ini, dan hanya membuat objek setelah Anda mengunggah semua bagian dan mengirim permintaan untuk menyelesaikan unggahan multipart. Setelah menerima keseluruhan permintaan unggahan multibagian, Amazon S3 akan menyusun bagian-bagiannya, dan membuat sebuah objek. Jika Anda tidak berhasil mengirim permintaan unggahan multibagian lengkap, S3 tidak merakit komponen dan tidak membuat objek apa pun. Jika Anda ingin tidak menyelesaikan unggahan multipart setelah mengunggah bagian, Anda harus membatalkan unggahan multipart.

Anda akan ditagih untuk semua penyimpanan yang terkait dengan bagian yang diunggah. Disarankan untuk selalu menyelesaikan unggahan multibagian atau menghentikan unggahan multibagian untuk menghapus bagian yang diunggah. Untuk informasi lebih lanjut tentang harga, lihatUnggahan multibagian dan harga.

Anda juga dapat menghentikan unggahan multibagian yang tidak lengkap menggunakan konfigurasi siklus hidup bucket. Untuk informasi selengkapnya, lihat Mengonfigurasi siklus hidup bucket untuk menghapus unggahan multibagian yang tidak lengkap.

Bagian berikut menunjukkan cara menghentikan unggahan multipart yang sedang berlangsung di Amazon S3 menggunakan AWS Command Line Interface, REST API, atau. AWS SDKs

Menggunakan AWS CLI

Untuk informasi selengkapnya tentang penggunaan AWS CLI untuk menghentikan unggahan multibagian, lihat abort-multipart-uploaddi Referensi AWS CLI Perintah.

Penggunaan API REST

Untuk informasi selengkapnya tentang penggunaan REST API untuk menghentikan unggahan multibagian, lihat AbortMultipartUploaddi Referensi API Amazon Simple Storage Service.

Menggunakan AWS SDKs (API tingkat tinggi)

Java

Kelas `TransferManager` menyediakan metode `abortMultipartUploads` untuk menghentikan unggahan multibagian yang sedang berlangsung. Pengunggahan dianggap sedang berlangsung setelah Anda memulainya, dan hingga Anda menyelesaikan atau menghentikannya. Anda memberikan nilai `Date`, dan API ini menghentikan semua unggahan multibagian pada bucket yang dimulai sebelum `Date` yang ditentukan dan masih dalam proses.

Tugas berikut memandu Anda dalam menggunakan kelas Java tingkat tinggi untuk menghentikan pengunggahan multibagian.

Proses penghentian unggahan multibagian API tingkat tinggi

1	Buat instans dari kelas `TransferManager` .
2	Jalankan metode `TransferManager.abortMultipartUploads` dengan meneruskan nama bucket dan nilai `Date`.

Contoh kode Java berikut menghentikan semua unggahan multibagian yang sedang berlangsung yang dimulai pada bucket tertentu lebih dari satu minggu yang lalu. Untuk petunjuk tentang cara membuat dan menguji sampel kerja, lihat Memulai di Panduan AWS SDK untuk Java Pengembang.

```
import java.util.Date;

import com.amazonaws.AmazonClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.transfer.TransferManager;

public class AbortMPUUsingHighLevelAPI {

    public static void main(String[] args) throws Exception {
        String existingBucketName = "*** Provide existing bucket name ***";

        TransferManager tm = new TransferManager(new ProfileCredentialsProvider());


        int sevenDays = 1000 * 60 * 60 * 24 * 7;
```

```
  Date oneWeekAgo = new Date(System.currentTimeMillis() - sevenDays);

        try {
         tm.abortMultipartUploads(existingBucketName, oneWeekAgo);
        } catch (AmazonClientException amazonClientException) {
         System.out.println("Unable to upload file, upload was aborted.");
         amazonClientException.printStackTrace();
        }
    }
}
```

> **Note**
>
> Anda juga dapat menghentikan unggahan multibagian tertentu. Untuk informasi selengkapnya, lihat Menggunakan AWS SDKs (API tingkat rendah).

.NET

Contoh C# berikut menghentikan semua unggahan multibagian yang sedang berlangsung yang dimulai pada bucket tertentu lebih dari seminggu yang lalu. Untuk informasi tentang menyiapkan dan menjalankan contoh kode, lihat Memulai SDK for .NET di AWSAWS SDK for .NET Developer Guide.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Transfer;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class AbortMPUUsingHighLevelAPITest
    {
        private const string bucketName = "*** provide bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main()
        {
```

```
            s3Client = new AmazonS3Client(bucketRegion);
            AbortMPUAsync().Wait();
        }

        private static async Task AbortMPUAsync()
        {
            try
            {
                var transferUtility = new TransferUtility(s3Client);

                // Abort all in-progress uploads initiated before the specified
 date.
                await transferUtility.AbortMultipartUploadsAsync(
                    bucketName, DateTime.Now.AddDays(-7));
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
        }
    }
}
```

> **Note**
>
> Anda juga dapat menghentikan unggahan multibagian tertentu. Untuk informasi selengkapnya, lihat Menggunakan AWS SDKs (API tingkat rendah).

Menggunakan AWS SDKs (API tingkat rendah)

Anda dapat menghentikan unggahan multibagian yang sedang berlangsung dengan memanggil metode `AmazonS3.abortMultipartUpload`. Metode ini menghapus semua bagian yang diunggah pada Amazon S3, dan mengosongkan sumber daya. Anda harus menyediakan ID unggahan, nama bucket, dan nama kunci. Contoh kode Java berikut menunjukkan cara menghentikan sebuah unggahan multibagian yang sedang berlangsung.

Untuk menghentikan unggahan multibagian, Anda harus memberikan ID unggahan, dan bucket serta nama kunci yang digunakan dalam unggahan tersebut. Setelah Anda menghentikan unggahan multibagian, Anda tidak dapat menggunakan ID unggahan untuk mengunggah bagian tambahan. Untuk informasi selengkapnya tentang unggahan multibagian Amazon S3, lihat Mengunggah dan menyalin objek menggunakan unggahan multipart di Amazon S3.

Java

Contoh kode Java berikut menghentikan unggahan multibagian yang sedang berlangsung.

Example

```
InitiateMultipartUploadRequest initRequest =
    new InitiateMultipartUploadRequest(existingBucketName, keyName);
InitiateMultipartUploadResult initResponse =
               s3Client.initiateMultipartUpload(initRequest);

AmazonS3 s3Client = new AmazonS3Client(new ProfileCredentialsProvider());
s3Client.abortMultipartUpload(new AbortMultipartUploadRequest(
            existingBucketName, keyName, initResponse.getUploadId()));
```

> **Note**
>
> Alih-alih melakukan unggahan multibagian tertentu, Anda dapat menghentikan semua unggahan multibagian yang dimulai sebelum waktu tertentu yang masih berlangsung. Operasi pembersihan ini berguna untuk menghentikan unggahan multibagian lama yang Anda mulai tetapi tidak diselesaikan atau dihentikan. Untuk informasi selengkapnya, lihat Menggunakan AWS SDKs (API tingkat tinggi).

.NET

Contoh C# berikut menunjukkan cara untuk menghentikan unggahan multibagian. Untuk keseluruhan sampel C# yang mencakup kode berikut, lihat Menggunakan AWS SDKs (API tingkat rendah).

```
AbortMultipartUploadRequest abortMPURequest = new AbortMultipartUploadRequest
{
    BucketName = existingBucketName,
    Key = keyName,
```

```
    UploadId = initResponse.UploadId
};
await AmazonS3Client.AbortMultipartUploadAsync(abortMPURequest);
```

Anda juga dapat membatalkan semua unggahan multibagian yang sedang berlangsung, yang dimulai sebelum waktu tertentu. Operasi pembersihan ini berguna untuk membatalkan unggahan multibagian yang tidak selesai atau dibatalkan. Untuk informasi selengkapnya, lihat Menggunakan AWS SDKs (API tingkat tinggi).

PHP

Contoh ini menunjukkan cara menggunakan kelas dari versi 3 AWS SDK untuk PHP untuk membatalkan unggahan multibagian yang sedang berlangsung. Untuk informasi selengkapnya tentang AWS SDK for Ruby API, AWS buka SDK for Ruby - Versi 2. Contoh metode `abortMultipartUpload()`.

Untuk informasi selengkapnya tentang AWS SDK for Ruby API, AWS buka SDK for Ruby - Versi 2.

```
 require 'vendor/autoload.php';

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';
$uploadId = '*** Upload ID of upload to Abort ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// Abort the multipart upload.
$s3->abortMultipartUpload([
    'Bucket'   => $bucket,
    'Key'      => $keyname,
    'UploadId' => $uploadId,
]);
```

Menyalin objek menggunakan unggahan multibagian

Unggahan multipart memungkinkan Anda menyalin objek sebagai satu set bagian. Contoh dalam bagian ini menunjukkan cara menyalin objek yang lebih besar dari 5 GB menggunakan API unggahan multibagian. Untuk informasi tentang unggahan multibagian, lihat. Mengunggah dan menyalin objek menggunakan unggahan multipart di Amazon S3

Anda dapat menyalin objek kurang dari 5 GB dalam satu operasi tanpa menggunakan API unggahan multibagian. Anda dapat menyalin objek kurang dari 5 GB menggunakan AWS Management Console, AWS CLI, REST API, atau AWS SDKs. Untuk informasi selengkapnya, lihat Menyalin, memindahkan, dan mengganti nama objek.

Untuk end-to-end prosedur mengunggah objek dengan unggahan multipart dengan checksum tambahan, lihat. Tutorial: Unggah objek melalui unggahan multipart dan verifikasi integritas datanya

Bagian berikut menunjukkan cara menyalin objek dengan unggahan multipart dengan REST API atau AWS SDKs.

Penggunaan API REST

Bagian berikut dalam Referensi API Amazon Simple Storage Service menjelaskan tentang API REST untuk unggahan multibagian. Untuk menyalin objek yang sudah ada, Anda dapat menggunakan API Unggah Bagian (Salinan), dan menentukan objek sumber dengan menambahkan header permintaan `x-amz-copy-source` dalam permintaan Anda.

- Mulai Unggahan Multibagian
- Unggah Bagian
- Unggah Bagian (Salinan)
- Selesaikan Unggahan Multibagian
- Batalkan Unggahan Multibagian
- Daftarkan Bagian
- Daftarkan Unggahan Multibagian

Anda dapat menggunakan ini APIs untuk membuat permintaan REST Anda sendiri, atau Anda dapat menggunakan salah satu dari yang SDKs kami sediakan. Untuk informasi selengkapnya tentang menggunakan Unggahan Multipart dengan AWS CLI, lihatMenggunakan AWS CLI. Untuk informasi lebih lanjut tentang SDKs, lihatAWS Dukungan SDK untuk unggahan multipart.

Menggunakan AWS SDKs

Untuk menyalin sebuah objek menggunakan API tingkat rendah, lakukan hal berikut ini:

- Mulai unggahan multibagian dengan memanggil metode `AmazonS3Client.initiateMultipartUpload()`.
- Simpan ID unggahan dari objek respons yang dikembalikan oleh metode `AmazonS3Client.initiateMultipartUpload()`. Anda memberikan ID unggahan ini untuk setiap operasi unggahan bagian.
- Salin semua bagiannya. Untuk setiap bagian yang harus Anda salin, buat sebuah instans baru dari kelas `CopyPartRequest`. Berikan informasi bagian, termasuk nama bucket sumber dan tujuan, kunci objek sumber dan tujuan, ID unggahan, lokasi byte pertama dan terakhir bagian tersebut, dan nomor bagian.
- Simpan respons dari panggilan metode `AmazonS3Client.copyPart()`. Setiap respons mencakup nilai `ETag` dan nomor bagian untuk bagian yang diunggah. Anda memerlukan informasi ini untuk menyelesaikan unggahan multibagian.
- Perintahkan metode `AmazonS3Client.completeMultipartUpload()` untuk menyelesaikan operasi penyalinan.

Java

Example

Contoh berikut ini menunjukkan cara menggunakan Java API tingkat rendah Amazon S3 untuk melakukan penyalinan multibagian. Untuk petunjuk cara membuat dan menguji sampel yang berfungsi, lihat Memulai di Panduan AWS SDK untuk Java Pengembang.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
```

```
public class LowLevelMultipartCopy {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String sourceBucketName = "*** Source bucket name ***";
        String sourceObjectKey = "*** Source object key ***";
        String destBucketName = "*** Target bucket name ***";
        String destObjectKey = "*** Target object key ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            // Initiate the multipart upload.
            InitiateMultipartUploadRequest initRequest = new
 InitiateMultipartUploadRequest(destBucketName,
                    destObjectKey);
            InitiateMultipartUploadResult initResult =
 s3Client.initiateMultipartUpload(initRequest);

            // Get the object size to track the end of the copy operation.
            GetObjectMetadataRequest metadataRequest = new
 GetObjectMetadataRequest(sourceBucketName, sourceObjectKey);
            ObjectMetadata metadataResult =
 s3Client.getObjectMetadata(metadataRequest);
            long objectSize = metadataResult.getContentLength();

            // Copy the object using 5 MB parts.
            long partSize = 5 * 1024 * 1024;
            long bytePosition = 0;
            int partNum = 1;
            List<CopyPartResult> copyResponses = new ArrayList<CopyPartResult>();
            while (bytePosition < objectSize) {
                // The last part might be smaller than partSize, so check to make
 sure
                // that lastByte isn't beyond the end of the object.
                long lastByte = Math.min(bytePosition + partSize - 1, objectSize -
 1);

                // Copy this part.
                CopyPartRequest copyRequest = new CopyPartRequest()
```

```
                        .withSourceBucketName(sourceBucketName)
                        .withSourceKey(sourceObjectKey)
                        .withDestinationBucketName(destBucketName)
                        .withDestinationKey(destObjectKey)
                        .withUploadId(initResult.getUploadId())
                        .withFirstByte(bytePosition)
                        .withLastByte(lastByte)
                        .withPartNumber(partNum++);
                copyResponses.add(s3Client.copyPart(copyRequest));
                bytePosition += partSize;
            }

            // Complete the upload request to concatenate all uploaded parts and
 make the
            // copied object available.
            CompleteMultipartUploadRequest completeRequest = new
 CompleteMultipartUploadRequest(
                    destBucketName,
                    destObjectKey,
                    initResult.getUploadId(),
                    getETags(copyResponses));
            s3Client.completeMultipartUpload(completeRequest);
            System.out.println("Multipart copy complete.");
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }

    // This is a helper function to construct a list of ETags.
    private static List<PartETag> getETags(List<CopyPartResult> responses) {
        List<PartETag> etags = new ArrayList<PartETag>();
        for (CopyPartResult response : responses) {
            etags.add(new PartETag(response.getPartNumber(), response.getETag()));
        }
        return etags;
    }
}
```

.NET

Contoh C # berikut menunjukkan cara menggunakan SDK untuk .NET untuk menyalin objek Amazon S3 yang lebih besar dari 5 GB dari satu lokasi sumber ke lokasi lain, seperti dari satu bucket ke bucket lainnya. Untuk menyalin objek yang lebih kecil dari 5 GB, gunakan prosedur penyalinan operasi tunggal yang dijelaskan dalam Menggunakan AWS SDKs. Untuk informasi selengkapnya tentang unggahan multibagian Amazon S3, lihat Mengunggah dan menyalin objek menggunakan unggahan multipart di Amazon S3.

Contoh ini menunjukkan cara menyalin objek Amazon S3 yang lebih besar dari 5 GB dari satu bucket S3 ke bucket lainnya menggunakan API unggahan SDK untuk .NET multipart.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class CopyObjectUsingMPUapiTest
    {
        private const string sourceBucket = "*** provide the name of the bucket with
 source object ***";
        private const string targetBucket = "*** provide the name of the bucket to
 copy the object to ***";
        private const string sourceObjectKey = "*** provide the name of object to
 copy ***";
        private const string targetObjectKey = "*** provide the name of the object
 copy ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            Console.WriteLine("Copying an object");
            MPUCopyObjectAsync().Wait();
```

```
        }
        private static async Task MPUCopyObjectAsync()
        {
            // Create a list to store the upload part responses.
            List<UploadPartResponse> uploadResponses = new
List<UploadPartResponse>();
            List<CopyPartResponse> copyResponses = new List<CopyPartResponse>();

            // Setup information required to initiate the multipart upload.
            InitiateMultipartUploadRequest initiateRequest =
                new InitiateMultipartUploadRequest
                {
                    BucketName = targetBucket,
                    Key = targetObjectKey
                };

            // Initiate the upload.
            InitiateMultipartUploadResponse initResponse =
                await s3Client.InitiateMultipartUploadAsync(initiateRequest);

            // Save the upload ID.
            String uploadId = initResponse.UploadId;

            try
            {
                // Get the size of the object.
                GetObjectMetadataRequest metadataRequest = new
GetObjectMetadataRequest
                {
                    BucketName = sourceBucket,
                    Key = sourceObjectKey
                };

                GetObjectMetadataResponse metadataResponse =
                    await s3Client.GetObjectMetadataAsync(metadataRequest);
                long objectSize = metadataResponse.ContentLength; // Length in
bytes.

                // Copy the parts.
                long partSize = 5 * (long)Math.Pow(2, 20); // Part size is 5 MB.

                long bytePosition = 0;
                for (int i = 1; bytePosition < objectSize; i++)
                {
```

```
                    CopyPartRequest copyRequest = new CopyPartRequest
                    {
                        DestinationBucket = targetBucket,
                        DestinationKey = targetObjectKey,
                        SourceBucket = sourceBucket,
                        SourceKey = sourceObjectKey,
                        UploadId = uploadId,
                        FirstByte = bytePosition,
                        LastByte = bytePosition + partSize - 1 >= objectSize ?
objectSize - 1 : bytePosition + partSize - 1,
                        PartNumber = i
                    };

                    copyResponses.Add(await s3Client.CopyPartAsync(copyRequest));

                    bytePosition += partSize;
                }

                // Set up to complete the copy.
                CompleteMultipartUploadRequest completeRequest =
                new CompleteMultipartUploadRequest
                {
                    BucketName = targetBucket,
                    Key = targetObjectKey,
                    UploadId = initResponse.UploadId
                };
                completeRequest.AddPartETags(copyResponses);

                // Complete the copy.
                CompleteMultipartUploadResponse completeUploadResponse =
                    await s3Client.CompleteMultipartUploadAsync(completeRequest);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
        }
    }
```

```
}
```

Tutorial: Unggah objek melalui unggahan multipart dan verifikasi integritas datanya

Unggahan multibagian memungkinkan Anda untuk mengunggah satu objek sebagai kumpulan bagian. Setiap bagian merupakan bagian data objek yang saling berkaitan. Anda dapat mengunggah bagian-bagian objek tersebut secara independen dan dengan urutan apa pun. Jika ada transmisi bagian mana pun yang gagal, Anda dapat mentransmisikan ulang bagian tersebut tanpa memengaruhi bagian lainnya. Setelah semua bagian objek Anda diunggah, Amazon S3 merakit bagian-bagian ini dan menciptakan objek. Secara umum, saat ukuran objek Anda mencapai 100 MB, Anda harus mempertimbangkan untuk menggunakan unggahan multibagian daripada mengunggah objek tersebut dalam satu operasi. Untuk informasi lebih lanjut tentang unggahan multibagian, lihat Mengunggah dan menyalin objek menggunakan unggahan multipart di Amazon S3. Untuk batasan yang terkait dengan unggahan multipart, lihat. Batas unggahan multibagian Amazon S3

Anda dapat menggunakan checksum untuk memverifikasi bahwa aset tidak diubah saat disalin. Melakukan checksum terdiri dari menggunakan algoritma untuk iterasi secara berurutan atas setiap byte dalam file. Amazon S3 menawarkan beberapa opsi checksum untuk memeriksa integritas data. Kami menyarankan Anda melakukan pemeriksaan integritas ini sebagai praktik terbaik daya tahan dan untuk mengonfirmasi bahwa setiap byte ditransfer tanpa perubahan. Amazon S3 juga mendukung algoritma berikut: SHA-1, SHA-256, dan C. CRC32 CRC32 Amazon S3 menggunakan satu atau lebih algoritme ini untuk menghitung nilai checksum tambahan dan menyimpannya sebagai bagian dari metadata objek. Untuk informasi selengkapnya tentang checksum, lihat Memeriksa integritas objek di Amazon S3.

Tujuan

Dalam tutorial ini, Anda akan belajar cara mengunggah objek ke Amazon S3 dengan menggunakan unggahan multipart dan checksum SHA-256 tambahan melalui Command AWS Line Interface (CLI).AWS Anda juga akan belajar cara memeriksa integritas data objek dengan menghitung MD5 hash dan checksum SHA-256 dari objek yang diunggah.

Topik

- Prasyarat
- Langkah 1: Buat file besar
- Langkah 2: Pisahkan file menjadi beberapa file
- Langkah 3: Buat unggahan multipart dengan checksum tambahan

- Langkah 4: Unggah bagian dari unggahan multipart Anda
- Langkah 5: Buat daftar semua bagian dari unggahan multipart Anda
- Langkah 6: Selesaikan unggahan multipart
- Langkah 7: Konfirmasikan bahwa objek diunggah ke bucket Anda
- Langkah 8: Verifikasi integritas objek dengan MD5 checksum
- Langkah 9: Verifikasi integritas objek dengan checksum tambahan
- Langkah 10: Bersihkan sumber daya Anda

Prasyarat

- Sebelum memulai tutorial ini, pastikan Anda memiliki akses ke bucket Amazon S3 yang dapat Anda unggah. Untuk informasi selengkapnya, lihat Membuat ember tujuan umum.
- Anda harus menginstal dan AWS mengkonfigurasi CLI. Jika AWS CLI belum diinstal, lihat Menginstal atau memperbarui ke versi terbaru dari AWS CLIAWS Command Line Interface Panduan Pengguna.
- Atau, Anda dapat menjalankan perintah AWS CLI dari konsol dengan menggunakan. AWS CloudShell AWS CloudShell adalah shell berbasis browser dan pra-otentikasi yang dapat Anda luncurkan langsung dari file. AWS Management Console Untuk informasi lebih lanjut, lihat Apa itu CloudShell? dan Memulai dengan AWS CloudShell di Panduan AWS CloudShell Pengguna.

Langkah 1: Buat file besar

Jika Anda sudah memiliki file yang siap diunggah, Anda dapat menggunakan file untuk tutorial ini. Jika tidak, buat file 15 MB menggunakan langkah-langkah berikut. Untuk batasan yang terkait dengan unggahan multipart, lihat. Batas unggahan multibagian Amazon S3

Untuk membuat file besar

Gunakan salah satu perintah berikut untuk membuat file Anda, tergantung pada sistem operasi yang Anda gunakan.

Linux atau macOS

Untuk membuat file 15 MB, buka terminal lokal Anda dan jalankan perintah berikut:

```
dd if=/dev/urandom of=census-data.bin bs=1M count=15
```

Perintah ini membuat file bernama `census-data.bin` diisi dengan byte acak, dengan ukuran 15 MB.

Windows

Untuk membuat file 15 MB, buka terminal lokal Anda dan jalankan perintah berikut:

```
fsutil file createnew census-data.bin 15728640
```

Perintah ini membuat file bernama `census-data.bin` dengan ukuran 15 MB data arbitrer (15728640 byte).

Langkah 2: Pisahkan file menjadi beberapa file

Untuk melakukan pengunggahan multipart, Anda harus membagi file besar Anda menjadi bagian-bagian yang lebih kecil. Anda kemudian dapat mengunggah bagian-bagian yang lebih kecil dengan menggunakan proses upload multipart. Langkah ini menunjukkan cara membagi file besar yang dibuat di Langkah 1 menjadi bagian-bagian yang lebih kecil. Contoh berikut menggunakan file 15 MB bernama`census-data.bin`.

Untuk membagi file besar menjadi beberapa bagian

Linux atau macOS

Untuk membagi file besar menjadi bagian 5 MB, gunakan `split` perintah. Buka terminal Anda dan jalankan yang berikut ini:

```
split -b 5M -d census-data.bin census-part
```

Perintah ini dibagi `census-data.bin` menjadi 5 MB bagian bernama`census-part**`, di mana `**` sufiks numerik dimulai dari. `00`

Windows

Untuk membagi file besar, gunakan PowerShell. Buka Powershell, dan jalankan skrip berikut:

```
$inputFile = "census-data.bin"
$outputFilePrefix = "census-part"
$chunkSize = 5MB

$fs = [System.IO.File]::OpenRead($inputFile)
```

```
$buffer = New-Object byte[] $chunkSize
$fileNumber = 0

while ($fs.Position -lt $fs.Length) {
$bytesRead = $fs.Read($buffer, 0, $chunkSize)
$outputFile = "{0}{1:D2}" -f $outputFilePrefix, $fileNumber
$fileStream = [System.IO.File]::Create($outputFile)
$fileStream.Write($buffer, 0, $bytesRead)
$fileStream.Close()
$fileNumber++
}

$fs.Close()
```

PowerShell Script ini membaca file besar dalam potongan 5 MB dan menulis setiap potongan ke file baru dengan akhiran numerik.

Setelah menjalankan perintah yang sesuai, Anda akan melihat bagian-bagian di direktori tempat Anda menjalankan perintah. Setiap bagian akan memiliki akhiran yang sesuai dengan nomor bagiannya, misalnya:

```
census-part00 census-part01 census-part02
```

Langkah 3: Buat unggahan multipart dengan checksum tambahan

Untuk memulai proses upload multipart, Anda perlu membuat permintaan upload multipart. Langkah ini melibatkan memulai unggahan multibagian dan menentukan checksum tambahan untuk integritas data. Contoh berikut menggunakan checksum SHA-256. Jika Anda ingin memberikan metadata apa pun yang menjelaskan objek yang sedang diunggah, Anda harus menyediakannya dalam permintaan untuk memulai unggahan multibagian.

> **Note**
>
> Pada langkah ini dan langkah selanjutnya, tutorial ini menggunakan algoritma tambahan SHA-256. Anda mungkin secara opsional menggunakan checksum tambahan lain untuk langkah-langkah ini, seperti, CRC32 C CRC32, atau SHA-1. Jika Anda menggunakan algoritma yang berbeda, Anda harus menggunakannya di seluruh langkah-langkah tutorial.

Untuk memulai unggahan multipart

Di terminal Anda, gunakan `create-multipart-upload` perintah berikut untuk memulai unggahan multipart untuk bucket Anda. Ganti *amzn-s3-demo-bucket1* dengan nama bucket Anda yang sebenarnya. Juga, ganti `census_data_file` dengan nama file yang Anda pilih. Nama file ini menjadi kunci objek saat unggahan selesai.

```
aws s3api create-multipart-upload --bucket amzn-s3-demo-bucket1 --key
 'census_data_file' --checksum-algorithm sha256
```

Jika permintaan Anda berhasil, Anda akan melihat output JSON seperti berikut:

```
{
    "ServerSideEncryption": "AES256",
    "ChecksumAlgorithm": "SHA256",
    "Bucket": "amzn-s3-demo-bucket1",
    "Key": "census_data_file",
    "UploadId":
 "cNV6KCSNANFZapz1LUGPC5XwUVi1n6yUoIeSP138sNOKPeMhpKQRrbT9k0ePmgoOTCj9K83T4e2Gb5hQvNoNpCKqyb8m3
}
```

> **Note**
>
> Saat Anda mengirim permintaan untuk memulai unggahan multibagian, Amazon S3 mengirimkan respons dengan ID unggahan, yang merupakan pengidentifikasi unik untuk unggahan multibagian Anda. Anda harus menyertakan ID unggahan ini setiap kali Anda mengunggah bagian, mendaftar bagian, menyelesaikan unggahan, atau menghentikan sebuah unggahan. Anda harus menggunakan `Bucket` nilai`UploadId`,`Key`, dan untuk langkah-langkah selanjutnya, jadi pastikan untuk menyimpannya.
> Selain itu, jika Anda menggunakan unggahan multibagian dengan checksum tambahan, nomor bagian harus berurutan. Jika Anda menggunakan nomor bagian yang tidak berurutan, `complete-multipart-upload` permintaan dapat menghasilkan HTTP. `500 Internal Server Error`

Langkah 4: Unggah bagian dari unggahan multipart Anda

Pada langkah ini, Anda akan mengunggah bagian dari unggahan multipart Anda ke bucket S3 Anda. Gunakan `upload-part` perintah untuk mengunggah setiap bagian satu per satu. Proses

ini memerlukan penentuan ID unggahan, nomor bagian, dan file yang akan diunggah untuk setiap bagian.

Untuk mengunggah bagian-bagiannya

1. Saat mengunggah bagian, selain ID unggahan, Anda harus menentukan nomor bagian dengan menggunakan `--part-number` argumen. Anda dapat memilih nomor bagian antara 1 hingga 10.000. Nomor bagian secara unik mengidentifikasi sebuah bagian dan posisinya dalam objek yang Anda unggah. Nomor bagian yang Anda pilih harus dalam urutan berurutan (misalnya, bisa 1, 2, atau 3). Jika Anda mengunggah sebuah bagian baru menggunakan nomor yang sama dengan bagian yang diunggah sebelumnya, bagian yang diunggah sebelumnya akan ditimpa.
2. Gunakan `upload-part` perintah untuk mengunggah setiap bagian dari unggahan multipart Anda. Sama seperti pada output yang dibuat oleh `create-multipart-upload` perintah di Langkah 3. `--upload-id` Untuk mengunggah bagian pertama data Anda, gunakan perintah berikut:

   ```
   aws s3api upload-part --bucket amzn-s3-demo-bucket1 --key
    'census_data_file' --part-number 1 --body census-part00 --upload-id
    "cNV6KCSNANFZapz1LUGPC5XwUVi1n6yUoIeSP138sNOKPeMhpKQRrbT9k0ePmgoOTCj9K83T4e2Gb5hQvNoNpCKqy
    --checksum-algorithm SHA256
   ```

   Setelah menyelesaikan setiap `upload-part` perintah, Anda akan melihat output seperti contoh berikut:

   ```
   {
       "ServerSideEncryption": "AES256",
       "ETag": "\"e611693805e812ef37f96c9937605e69\"",
       "ChecksumSHA256": "QLl8R4i4+SaJlrl8ZIcutc5TbZtwt2NwB8lTXkd3GH0="
   }
   ```

3. Untuk bagian selanjutnya, tambahkan nomor bagian yang sesuai:

   ```
   aws s3api upload-part --bucket amzn-s3-demo-bucket1 --key 'census_data_file' --
   part-number <part-number> --body <file-path> --upload-id "<your-upload-id>" --
   checksum-algorithm SHA256
   ```

   Misalnya, gunakan perintah berikut untuk mengunggah bagian kedua:

```
aws s3api upload-part --bucket amzn-s3-demo-bucket1 --key
 'census_data_file' --part-number 2 --body census-part01 --upload-id
 "cNV6KCSNANFZapz1LUGPC5XwUVi1n6yUoIeSP138sNOKPeMhpKQRrbT9k0ePmgoOTCj9K83T4e2Gb5hQvNoNpCKqy
 --checksum-algorithm SHA256
```

Amazon S3 mengembalikan tag entitas (ETag) dan checksum tambahan untuk setiap bagian yang diunggah sebagai header dalam respons.

4. Lanjutkan menggunakan `upload-part` perintah sampai Anda mengunggah semua bagian objek Anda.

Langkah 5: Buat daftar semua bagian dari unggahan multipart Anda

Untuk menyelesaikan unggahan multibagian, Anda memerlukan daftar semua bagian yang telah diunggah untuk unggahan multibagian tertentu. Output dari `list-parts` perintah memberikan informasi seperti nama bucket, kunci, ID unggah, nomor bagian, checksum tambahan ETag, dan banyak lagi. Sangat membantu untuk menyimpan output ini dalam file sehingga Anda dapat menggunakannya untuk langkah berikutnya saat menyelesaikan proses pengunggahan multibagian. Anda dapat membuat file keluaran JSON yang disebut `parts.json` dengan menggunakan metode berikut.

Untuk membuat file yang mencantumkan semua bagian

1. Untuk menghasilkan file JSON dengan rincian semua bagian yang diunggah, gunakan perintah berikut`list-parts`. Ganti ***amzn-s3-demo-bucket1*** dengan nama bucket Anda yang sebenarnya dan **<your-upload-id>** dengan ID unggahan yang Anda terima di Langkah 3. Untuk informasi selengkapnya tentang `list-parts` perintah, lihat list-partsdi Panduan AWS Command Line Interface Pengguna.

   ```
   aws s3api list-parts --bucket amzn-s3-demo-bucket1 --key 'census_data_file' --
   upload-id <your-upload-id> --query '{Parts: Parts[*].{PartNumber: PartNumber, ETag:
    ETag, ChecksumSHA256: ChecksumSHA256}}' --output json > parts.json
   ```

   File baru bernama `parts.json` dihasilkan. File tersebut berisi informasi berformat JSON untuk semua bagian yang Anda unggah. `parts.json`File tersebut menyertakan informasi penting untuk setiap bagian dari unggahan multibagian Anda, seperti nomor bagian dan ETag nilai yang sesuai, yang diperlukan untuk menyelesaikan proses pengunggahan multibagian.

2. Buka `parts.json` dengan menggunakan editor teks apa pun atau melalui terminal. Berikut contoh outputnya:

```
{
    "Parts": [
        {
            "PartNumber": 1,
            "ETag": "\"3c3097f89e2a2fece47ac54b243c9d97\"",
            "ChecksumSHA256": "fTPVHfyNHdv5VkR4S3EewdyioXECv7JBxN+d4FXYYTw="
        },
        {
            "PartNumber": 2,
            "ETag": "\"03c71cc160261b20ab74f6d2c476b450\"",
            "ChecksumSHA256": "VDWTa8enjOvULBAO3W2a6C+5/7ZnNjrnLApa1QVc3FE="
        },
        {
            "PartNumber": 3,
            "ETag": "\"81ae0937404429a97967dffa7eb4affb\"",
            "ChecksumSHA256": "cVVkXehUlzcwrBrXgPIM+EKQXPUvWist8mlUTCs4bg8="
        }
    ]
}
```

Langkah 6: Selesaikan unggahan multipart

Setelah mengunggah semua bagian dari unggahan multipart Anda dan mencantumkannya, langkah terakhir adalah menyelesaikan unggahan multipart. Langkah ini menggabungkan semua bagian yang diunggah menjadi satu objek di bucket S3 Anda.

> **Note**
>
> Anda dapat menghitung checksum objek sebelum menelepon `complete-multipart-upload` dengan memasukkan `--checksum-sha256` dalam permintaan Anda. Jika checksum tidak cocok, Amazon S3 gagal permintaan. Untuk informasi selengkapnya, lihat complete-multipart-upload di AWS Command Line Interface Panduan Pengguna.

Untuk menyelesaikan unggahan multipart

Untuk menyelesaikan unggahan multipart, gunakan perintah. `complete-multipart-upload` Perintah ini memerlukan `parts.json` file yang dibuat di Langkah 5, nama bucket Anda, dan ID unggahan. Ganti **<*amzn-s3-demo-bucket1*>** dengan nama bucket Anda dan **<your-upload-id>** dengan ID unggah dari`parts.json`.

```
aws s3api complete-multipart-upload --multipart-upload file://parts.json --bucket amzn-s3-demo-bucket1 --key 'census_data_file' --upload-id <your-upload-id>
```

Berikut contoh outputnya:

```
{
    "ServerSideEncryption": "AES256",
    "Location": "https://amzn-s3-demo-bucket1.s3.us-east-2.amazonaws.com/census_data_file",
    "Bucket": "amzn-s3-demo-bucket1",
    "Key": "census_data_file",
    "ETag": "\"f453c6dccca969c457efdf9b1361e291-3\"",
    "ChecksumSHA256": "aI8EoktCdotjU8Bq46DrPCxQCGuGcPIhJ51noWs6hvk=-3"
}
```

> **Note**
>
> Jangan menghapus file bagian individual. Anda akan memerlukan bagian-bagian individual sehingga Anda dapat melakukan checksum pada mereka untuk memverifikasi integritas objek yang digabungkan bersama.

Langkah 7: Konfirmasikan bahwa objek diunggah ke bucket Anda

Setelah menyelesaikan pengunggahan multipart, Anda dapat memverifikasi bahwa objek telah berhasil diunggah ke bucket S3 Anda. Untuk membuat daftar objek di bucket Anda dan mengonfirmasi keberadaan file yang baru diunggah, gunakan perintah `list-objects-v2`

Untuk daftar objek yang diunggah

Untuk membuat daftar objek di Anda, gunakan bucket `list-objects-v2` perintah. Ganti ***amzn-s3-demo-bucket1*** dengan nama bucket Anda yang sebenarnya:

```
aws s3api list-objects-v2 --bucket amzn-s3-demo-bucket1
```

Perintah ini mengembalikan daftar objek di bucket Anda. Cari file yang Anda unggah (misalnya,`census_data_file`) dalam daftar objek.

Untuk informasi selengkapnya, lihat bagian Contoh untuk `list-objects-v2` perintah di Panduan AWS Command Line Interface Pengguna.

Langkah 8: Verifikasi integritas objek dengan MD5 checksum

Saat mengunggah objek, Anda dapat menentukan algoritma checksum untuk Amazon S3 untuk digunakan. Secara default, Amazon S3 menyimpan MD5 intisari byte sebagai objek. ETag Untuk unggahan multipart, ETag ini bukan checksum untuk seluruh objek, melainkan gabungan checksum untuk setiap bagian individu.

Untuk memverifikasi integritas objek dengan menggunakan MD5 checksum

1. Untuk mengambil objek ETag yang diunggah, lakukan permintaan: `head-object`

   ```
   aws s3api head-object --bucket amzn-s3-demo-bucket1 --key census_data_file
   ```

   Berikut contoh outputnya:

   ```
   {
       "AcceptRanges": "bytes",
       "LastModified": "2024-07-26T19:04:13+00:00",
       "ContentLength": 16106127360,
       "ETag": "\"f453c6dccca969c457efdf9b1361e291-3\"",
       "ContentType": "binary/octet-stream",
       "ServerSideEncryption": "AES256",
       "Metadata": {}
   }
   ```

   Ini ETag memiliki “-3" ditambahkan ke akhir. Ini menunjukkan bahwa objek diunggah dalam tiga bagian menggunakan unggahan multipart.

2. Selanjutnya, hitung MD5 checksum setiap bagian menggunakan `md5sum` perintah. Pastikan Anda memberikan jalur yang benar ke file bagian Anda:

```
md5sum census-part*
```

Berikut contoh outputnya:

```
e611693805e812ef37f96c9937605e69 census-part00
63d2d5da159178785bfd6b6a5c635854 census-part01
95b87c7db852451bb38b3b44a4e6d310 census-part02
```

3. Untuk langkah ini, gabungkan MD5 hash secara manual menjadi satu string. Kemudian, jalankan perintah berikut untuk mengubah string menjadi biner dan menghitung MD5 checksum dari nilai biner:

```
echo
 "e611693805e812ef37f96c9937605e6963d2d5da159178785bfd6b6a5c63585495b87c7db852451bb38b3b44a
 | xxd -r -p | md5sum
```

Berikut contoh outputnya:

```
f453c6dccca969c457efdf9b1361e291 -
```

Nilai hash ini harus sesuai dengan nilai hash dari ETag nilai asli di Langkah 1, yang memvalidasi integritas objek. `census_data_file`

Saat Anda menginstruksikan Amazon S3 untuk menggunakan checksum tambahan, Amazon S3 menghitung nilai checksum untuk setiap bagian dan menyimpan nilainya. Jika Anda ingin mengambil nilai checksum untuk masing-masing bagian dari unggahan multibagian yang masih berlangsung, Anda dapat menggunakannya. `list-parts`

Untuk informasi selengkapnya tentang cara kerja checksum dengan objek unggahan multibagian, lihat. Memeriksa integritas objek di Amazon S3

Langkah 9: Verifikasi integritas objek dengan checksum tambahan

Pada langkah ini, tutorial ini menggunakan SHA-256 sebagai checksum tambahan untuk memvalidasi integritas objek. Jika Anda telah menggunakan checksum tambahan yang berbeda, gunakan nilai checksum itu sebagai gantinya.

Untuk memverifikasi integritas objek dengan SHA256

1. Jalankan perintah berikut di terminal Anda, termasuk `--checksum-mode enabled` argumen, untuk menampilkan `ChecksumSHA256` nilai objek Anda:

```
aws s3api head-object --bucket amzn-s3-demo-bucket1 --key census_data_file --
checksum-mode enabled
```

   Berikut contoh outputnya:

```
{
    "AcceptRanges": "bytes",
    "LastModified": "2024-07-26T19:04:13+00:00",
    "ContentLength": 16106127360,
    "ChecksumSHA256": "aI8EoktCdotjU8Bq46DrPCxQCGuGcPIhJ51noWs6hvk=-3",
    "ETag": "\"f453c6dccca969c457efdf9b1361e291-3\"",
    "ContentType": "binary/octet-stream",
    "ServerSideEncryption": "AES256",
    "Metadata": {}
}
```

2. Gunakan perintah berikut untuk memecahkan kode `ChecksumSHA256` nilai untuk masing-masing bagian menjadi base64 dan menyimpannya ke dalam file biner yang disebut. `outfile` Nilai-nilai ini dapat ditemukan di `parts.json` file Anda. Ganti contoh string base64 dengan nilai aktual `ChecksumSHA256` Anda.

```
echo "QLl8R4i4+SaJlrl8ZIcutc5TbZtwt2NwB8lTXkd3GH0=" | base64 --decode >> outfile
echo "xCdgs1K5Bm4jWETYw/CmGYr+m6O2DcGfpckx5NVokvE=" | base64 --decode >> outfile
echo "f5wsfsa5bB+yXuwzqG1Bst91uYneqGD3CCidpb54mAo=" | base64 --decode >> outfile
```

3. Jalankan perintah berikut untuk menghitung SHA256 checksum dari`outfile`:

```
sha256sum outfile
```

   Berikut contoh outputnya:

```
688f04a24b42768b6353c06ae3a0eb3c2c50086b8670f221279d67a16b3a86f9 outfile
```

Pada langkah berikutnya, ambil nilai hash dan ubah menjadi nilai biner. Nilai biner ini harus sesuai dengan `ChecksumSHA256` nilai dari Langkah 1.

4. Ubah SHA256 checksum dari Langkah 3 menjadi biner, lalu kodekan ke base64 untuk memverifikasi bahwa itu cocok dengan `ChecksumSHA256` nilai dari Langkah 1:

```
echo "688f04a24b42768b6353c06ae3a0eb3c2c50086b8670f221279d67a16b3a86f9" | xxd -r -p
 | base64
```

Berikut contoh outputnya:

```
aI8EoktCdotjU8Bq46DrPCxQCGuGcPIhJ51noWs6hvk=
```

Output ini harus mengkonfirmasi bahwa output base64 cocok dengan `ChecksumSHA256` nilai dari output `head-object` perintah. Jika output cocok dengan nilai checksum, maka objek tersebut valid.

> **Important**
>
> - Saat Anda menginstruksikan Amazon S3 untuk menggunakan checksum tambahan, Amazon S3 menghitung nilai checksum untuk setiap bagian dan menyimpan nilai ini.
> - Jika Anda ingin mengambil nilai checksum untuk masing-masing bagian dari unggahan multibagian yang masih berlangsung, Anda dapat menggunakan perintah tersebut. `list-parts`

Langkah 10: Bersihkan sumber daya Anda

Jika Anda ingin membersihkan file yang dibuat dalam tutorial ini, gunakan metode berikut. Untuk petunjuk cara menghapus file yang diunggah ke bucket S3 Anda, lihat. Menghapus objek Amazon S3

Hapus file lokal yang dibuat di Langkah 1:

Untuk menghapus file yang Anda buat untuk upload multipart Anda, jalankan perintah berikut dari direktori kerja Anda:

```
rm census-data.bin census-part* outfile parts.json
```

Batas unggahan multibagian Amazon S3

Unggahan multibagian memungkinkan Anda untuk mengunggah satu objek sebagai kumpulan bagian. Setiap bagian merupakan bagian data objek yang saling berkaitan. Setelah semua bagian objek Anda diunggah, Amazon S3 merakit bagian-bagian ini dan menciptakan objek. Secara umum, saat ukuran objek Anda mencapai 100 MB, Anda harus mempertimbangkan untuk menggunakan unggahan multibagian daripada mengunggah objek tersebut dalam satu operasi. Untuk informasi lebih lanjut tentang unggahan multibagian, lihat Mengunggah dan menyalin objek menggunakan unggahan multipart di Amazon S3.

Tabel berikut ini menyediakan spesifikasi inti unggahan multibagian. Ini termasuk ukuran objek maksimum, jumlah maksimum bagian, ukuran bagian maksimum, dan banyak lagi. Tidak ada batas ukuran minimum pada bagian terakhir dari unggahan multibagian Anda.

Item	Spesifikasi
Ukuran objek maksimum	5 TiB
Jumlah maksimum bagian per unggahan	10.000
Nomor bagian	1 hingga 10.000 (inklusif)
Ukuran bagian	5 MiB hingga 5 GiB. Tidak ada batas ukuran minimum pada bagian terakhir dari unggahan multibagian Anda.
Jumlah maksimum bagian yang ditampilkan untuk permintaan daftar bagian	1000
Jumlah maksimum unggahan multibagian yang dikembalikan dalam permintaan unggahan multibagian daftar	1000

Tambahkan prasyarat ke operasi S3 dengan permintaan bersyarat

Anda dapat menggunakan permintaan bersyarat untuk menambahkan prasyarat ke operasi S3 Anda. Untuk menggunakan permintaan bersyarat, Anda menambahkan header tambahan ke operasi Amazon S3 API Anda. Header ini menentukan kondisi yang, jika tidak terpenuhi, akan mengakibatkan operasi S3 gagal.

Pembacaan bersyarat didukung untuk`GET`,`HEAD`, dan `COPY` permintaan. Anda dapat menambahkan prasyarat untuk mengembalikan atau menyalin objek berdasarkan tag Entity (ETag) atau tanggal modifikasi terakhir. Ini dapat membatasi operasi S3 ke objek yang diperbarui sejak tanggal yang ditentukan. Anda juga dapat membatasi operasi S3 ke yang spesifik ETag. Ini dapat memastikan Anda hanya mengembalikan atau menyalin versi objek tertentu. Untuk informasi selengkapnya tentang metadata objek, lihat Bekerja dengan metadata objek.

Penulisan bersyarat dapat memastikan tidak ada objek yang ada dengan nama kunci yang sama di bucket Anda selama `PUT` operasi. Ini mencegah penimpaan objek yang ada dengan nama kunci yang identik. Demikian pula, Anda dapat menggunakan penulisan bersyarat untuk memeriksa apakah objek tidak berubah sebelum memperbarui objek. ETag Ini mencegah penimpaan yang tidak disengaja pada suatu objek tanpa mengetahui keadaan kontennya. Anda dapat menggunakan penulisan bersyarat untuk PutObjectatau CompleteMultipartUploadpermintaan. Untuk informasi selengkapnya tentang nama kunci, lihat Menamai objek Amazon S3.

Tidak ada biaya tambahan untuk pembacaan bersyarat atau penulisan bersyarat. Anda hanya dikenakan tarif yang ada untuk permintaan yang berlaku, termasuk untuk permintaan yang gagal. Untuk informasi tentang fitur dan harga Amazon S3, lihat Harga Amazon S3.

Topik

- Cara mengambil atau menyalin objek berdasarkan metadata dengan pembacaan bersyarat
- Bagaimana mencegah penimpaan objek dengan penulisan bersyarat

Cara mengambil atau menyalin objek berdasarkan metadata dengan pembacaan bersyarat

Dengan pembacaan bersyarat, Anda dapat menambahkan header tambahan ke permintaan baca Anda untuk menambahkan prasyarat ke operasi S3 Anda. Jika prasyarat ini tidak terpenuhi, permintaan baca akan gagal.

Anda dapat menggunakan pembacaan bersyarat pada`GET`,`HEAD`, atau `COPY` permintaan untuk hanya mengembalikan objek berdasarkan metadatanya.

Saat Anda mengunggah objek, Amazon S3 membuat metadata yang dikontrol sistem yang hanya dapat dimodifikasi oleh S3. Entity tags (ETags) dan Last-Modified adalah contoh metadata yang dikendalikan sistem. Objek ETag adalah string yang mewakili versi tertentu dari suatu objek. Tanggal Modifikasi Terakhir adalah metadata yang mewakili tanggal pembuatan objek atau tanggal modifikasi terakhir, mana yang terbaru.

Dengan pembacaan bersyarat, Anda dapat mengembalikan objek berdasarkan tanggal objek ETag atau Terakhir Modifikasi. Anda dapat menentukan ETag nilai dengan permintaan Anda dan mengembalikan objek hanya jika ETag nilainya cocok. Ini dapat memastikan Anda hanya mengembalikan atau menyalin versi tertentu dari suatu objek. Anda dapat menentukan nilai Modifikasi Terakhir dengan permintaan baca Anda dan mengembalikan objek hanya jika objek tersebut telah dimodifikasi sejak tanggal yang Anda berikan.

Didukung APIs

APIs Dukungan S3 berikut menggunakan pembacaan bersyarat:

- GetObject
- HeadObject
- CopyObject

Anda dapat menggunakan header berikut untuk mengembalikan objek yang bergantung pada tag entitas (ETag) atau tanggal modifikasi terakhir. Untuk informasi selengkapnya tentang metadata objek seperti ETags dan Last-Modified, lihat. the section called “Metadata objek yang ditentukan sistem”

GetObject

- `If-Match`— Kembalikan objek hanya jika ETag cocok dengan yang disediakan.
- `If-Modified-Since`— Kembalikan objek hanya jika telah dimodifikasi sejak waktu yang ditentukan.
- `If-None-Match`— Kembalikan objek hanya jika ETag tidak cocok dengan yang disediakan.
- `If-Unmodified-Since`— Kembalikan objek hanya jika belum dimodifikasi sejak waktu yang ditentukan.

Untuk informasi selengkapnya tentang header ini, error yang ditampilkan, dan urutan S3 menangani beberapa header bersyarat dalam satu permintaan, lihat di Referensi API Amazon Simple Storage Service. GetObject

HeadObject

- `If-Match`— Kembalikan objek hanya jika ETag cocok dengan yang disediakan.
- `If-Modified-Since`— Kembalikan objek hanya jika telah dimodifikasi sejak waktu yang ditentukan.
- `If-None-Match`— Kembalikan objek hanya jika ETag tidak cocok dengan yang disediakan.
- `If-Unmodified-Since`— Kembalikan objek hanya jika belum dimodifikasi sejak waktu yang ditentukan.

Untuk informasi selengkapnya tentang header ini, error yang ditampilkan, dan urutan S3 menangani beberapa header bersyarat dalam satu permintaan, lihat di Referensi API Amazon Simple Storage Service. HeadObject

CopyObject

- `x-amz-copy-source-if-match`— Menyalin objek sumber hanya jika ETag cocok dengan yang disediakan.
- `x-amz-copy-source-if-modified-since`— Menyalin objek sumber hanya jika telah dimodifikasi sejak waktu yang ditentukan.
- `x-amz-copy-source-if-none-match`— Menyalin objek sumber hanya jika ETag tidak cocok dengan yang disediakan.
- `x-amz-copy-source-if-unmodified-since`— Menyalin objek sumber hanya jika belum dimodifikasi sejak waktu yang ditentukan.

Untuk informasi selengkapnya tentang header ini, error yang ditampilkan, dan urutan S3 menangani beberapa header bersyarat dalam satu permintaan, lihat di Referensi API Amazon Simple Storage Service. CopyObject

Bagaimana mencegah penimpaan objek dengan penulisan bersyarat

Dengan menggunakan penulisan bersyarat, Anda dapat menambahkan header tambahan ke `WRITE` permintaan Anda untuk menentukan prasyarat untuk operasi Amazon S3 Anda. Untuk menulis objek secara kondisional, tambahkan HTTP `If-None-Match` atau `If-Match` header.

`If-None-Match`Header mencegah penimpaan data yang ada dengan memvalidasi bahwa tidak ada objek dengan nama kunci yang sama di bucket Anda.

Atau, Anda dapat menambahkan `If-Match` header untuk memeriksa tag entitas objek (ETag) sebelum menulis objek. Dengan header ini, Amazon S3 membandingkan nilai yang diberikan dengan ETag nilai objek di ETag S3. Jika ETag nilainya tidak cocok, operasi gagal.

Pemilik bucket dapat menggunakan kebijakan bucket untuk menerapkan penulisan bersyarat untuk objek yang diunggah. Untuk informasi selengkapnya, lihat the section called “Menegakkan penulisan bersyarat”.

> **Note**
>
> Untuk menggunakan penulisan bersyarat, Anda harus membuat permintaan melalui HTTPS (TLS) atau menggunakan AWS Signature Version 4 untuk menandatangani permintaan.

Topik

- Bagaimana mencegah penimpaan objek berdasarkan nama kunci
- Bagaimana mencegah penimpaan jika objek telah berubah
- Perilaku menulis bersyarat
- Skenario tulis bersyarat
- Menegakkan penulisan bersyarat di ember Amazon S3

Bagaimana mencegah penimpaan objek berdasarkan nama kunci

Anda dapat menggunakan header `If-None-Match` bersyarat HTTP untuk memeriksa apakah objek sudah ada di bucket yang ditentukan berdasarkan nama kuncinya sebelum membuatnya. Saat Anda mengunggah objek ke Amazon S3, tentukan nama kunci: pengenal unik dan peka huruf besar/kecil dari objek dalam bucket. Tanpa `If-None-Match` header HTTP, jika Anda mengunggah objek

dengan nama kunci yang identik dalam bucket yang tidak berversi atau ditangguhkan versi, objek akan ditimpa. Dalam bucket berversi, objek yang paling baru diunggah menjadi versi objek saat ini. Penulisan bersyarat dengan `If-None-Match` header HTTP memeriksa keberadaan objek selama `WRITE` operasi. Jika nama kunci yang identik ditemukan di bucket, operasi gagal. Untuk informasi selengkapnya tentang menggunakan nama kunci, lihatthe section called “Penamaan objek”.

Untuk melakukan penulisan bersyarat dengan `If-None-Match` header HTTP Anda harus memiliki `s3:PutObject` izin. Hal ini memungkinkan penelepon untuk memeriksa keberadaan objek dalam ember. `If-None-Match`Header mengharapkan nilai * (tanda bintang).

Anda dapat menggunakan `If-None-Match` header dengan yang berikut APIs:

- PutObject
- CompleteMultipartUpload

Menggunakan AWS CLI

`put-object`Contoh perintah berikut mencoba untuk melakukan penulisan bersyarat untuk objek dengan nama `dir-1/my_images.tar.bz2` kunci.

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key dir-1/my_images.tar.bz2 --body
 my_images.tar.bz2 --if-none-match "*"
```

Untuk informasi selengkapnya, lihat put-object dalam AWS CLI Referensi Perintah.

Untuk informasi tentang AWS CLI, lihat Apa itu AWS Command Line Interface? dalam AWS Command Line Interface User Guide.

Bagaimana mencegah penimpaan jika objek telah berubah

Objek ETag adalah string yang unik untuk objek dan mencerminkan perubahan pada konten objek. Anda dapat menggunakan `If-Match` header untuk membandingkan ETag nilai objek di bucket Amazon S3 dengan yang Anda berikan selama operasi. `WRITE` Jika ETag nilainya tidak cocok, operasi gagal. Untuk informasi lebih lanjut tentang ETags, lihatthe section called “Menggunakan Konten- MD5 dan ETag untuk memverifikasi objek yang diunggah”.

Untuk melakukan penulisan bersyarat dengan `If-Match` header HTTP Anda harus memiliki `s3:PutObject` dan `s3:GetObject` izin. Hal ini memungkinkan penelepon untuk memeriksa ETag

dan memverifikasi status objek dalam ember. `If-Match`Header mengharapkan ETag nilai sebagai string.

Anda dapat menggunakan `If-Match` header dengan yang berikut APIs:

- PutObject
- CompleteMultipartUpload

Menggunakan AWS CLI

`put-object`Contoh perintah berikut mencoba untuk melakukan penulisan bersyarat dengan ETag nilai `6805f2cfc46c0f04559748bb039d69ae` yang disediakan.

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key dir-1/my_images.tar.bz2 --body
 my_images.tar.bz2 --if-match "6805f2cfc46c0f04559748bb039d69ae"
```

Untuk informasi selengkapnya, lihat put-object dalam AWS CLI Referensi Perintah.

Untuk informasi tentang AWS CLI, lihat Apa itu AWS Command Line Interface? dalam AWS Command Line Interface User Guide.

Perilaku menulis bersyarat

Menulis bersyarat dengan header **`If-None-Match`**

Penulisan bersyarat dengan `If-None-Match` header mengevaluasi objek yang ada dalam ember. Jika tidak ada objek yang ada dengan nama kunci yang sama di bucket, operasi penulisan berhasil, menghasilkan `200 OK` respons. Jika ada objek yang ada, operasi tulis gagal, menghasilkan `412 Precondition Failed` respons.

Untuk bucket dengan versi diaktifkan, jika tidak ada versi objek saat ini dengan nama yang sama, atau jika versi objek saat ini adalah penanda hapus, operasi tulis berhasil. Jika tidak, itu menghasilkan operasi penulisan yang gagal dengan `412 Precondition Failed` respons.

Jika beberapa penulisan bersyarat terjadi untuk nama objek yang sama, operasi penulisan pertama yang selesai berhasil. Amazon S3 kemudian gagal menulis berikutnya dengan tanggapan. `412 Precondition Failed`

Anda juga dapat menerima `409 Conflict` respons dalam kasus permintaan bersamaan jika permintaan penghapusan ke objek berhasil sebelum operasi tulis bersyarat pada

objek tersebut selesai. Saat menggunakan penulisan bersyarat dengan`PutObject`, unggahan dapat dicoba lagi setelah menerima kesalahan. `409 Conflict` Saat menggunakan`CompleteMultipartUpload`, seluruh unggahan multibagian harus dimulai ulang `CreateMultipartUpload` untuk mengunggah objek lagi setelah menerima kesalahan. `409 Conflict`

Menulis bersyarat dengan header **`If-Match`**

`If-Match`Header mengevaluasi terhadap objek yang ada dalam ember. Jika ada objek yang ada dengan nama kunci dan pencocokan yang sama ETag, operasi penulisan berhasil, menghasilkan `200 OK` respons. Jika ETag tidak cocok, operasi tulis gagal dengan `412 Precondition Failed` respons.

Anda juga dapat menerima `409 Conflict` tanggapan dalam kasus permintaan bersamaan.

Anda akan menerima `404 Not Found` respons jika permintaan penghapusan bersamaan ke objek berhasil sebelum operasi tulis bersyarat pada objek tersebut selesai, karena kunci objek tidak ada lagi. Anda harus mengunggah ulang objek ketika Anda menerima `404 Not Found` respons.

Jika tidak ada versi objek saat ini dengan nama yang sama, atau jika versi objek saat ini adalah penanda hapus, operasi gagal dengan `404 Not Found` kesalahan.

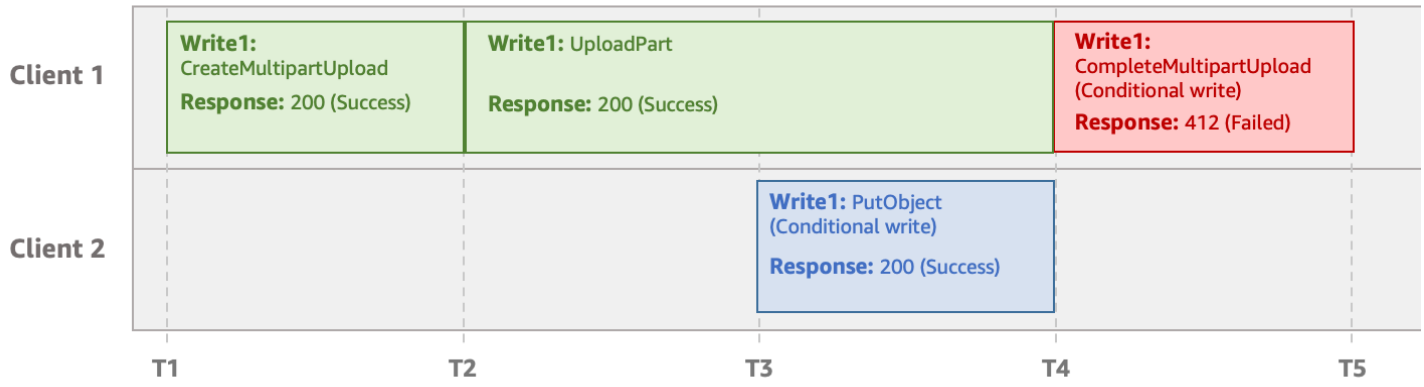
Skenario tulis bersyarat

Pertimbangkan skenario berikut di mana dua klien menjalankan operasi pada bucket yang sama.

Penulisan bersyarat selama unggahan multipart

Penulisan bersyarat tidak mempertimbangkan permintaan unggahan multibagian yang sedang berlangsung karena itu belum sepenuhnya ditulis objek. Perhatikan contoh berikut di mana Klien 1 mengunggah objek menggunakan unggahan multipart. Selama pengunggahan multipart, Klien 2 berhasil menulis objek yang sama dengan operasi penulisan bersyarat. Selanjutnya, ketika Klien 1 mencoba menyelesaikan unggahan multibagian menggunakan penulisan bersyarat, unggahan gagal.

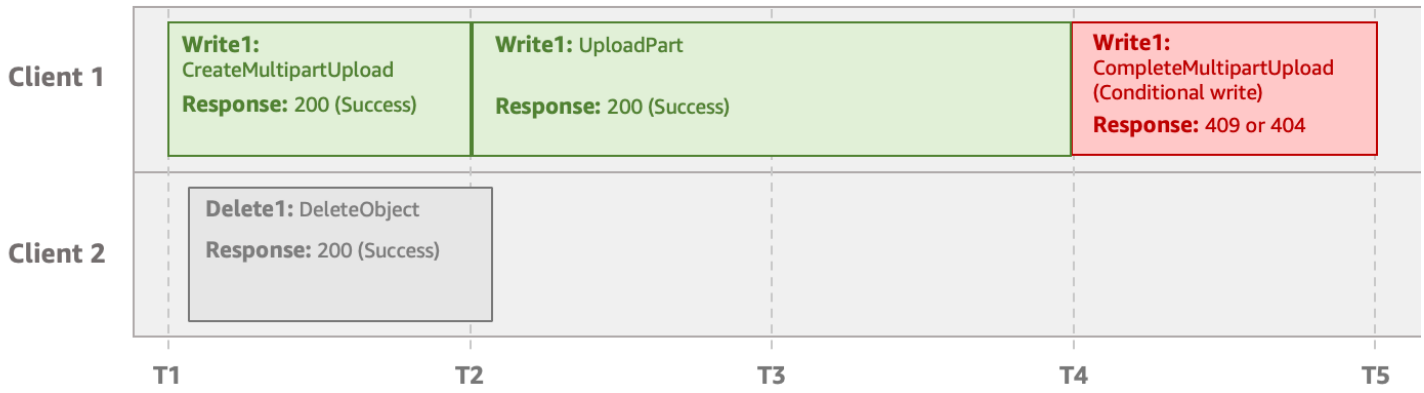
> **Note**
>
> Skenario ini akan menghasilkan `412 Precondition Failed` respons untuk keduanya `If-None-Match` dan `If-Match` header.



Penghapusan bersamaan selama unggahan multipart

Jika permintaan hapus berhasil sebelum permintaan penulisan bersyarat dapat diselesaikan, Amazon S3 mengembalikan `409 Conflict` respons `404 Not Found` atau untuk operasi tulis. Ini karena permintaan hapus yang dimulai sebelumnya lebih diutamakan daripada operasi tulis bersyarat. Dalam kasus seperti itu, Anda harus memulai unggahan multipart baru.

> **Note**
>
> Skenario ini akan menghasilkan `409 Conflict` respons untuk `If-None-Match` header dan `404 Not Found` respons untuk `If-Match` header.



> **Note**
>
> Untuk meminimalkan biaya penyimpanan, kami menyarankan untuk Anda mengonfigurasi aturan siklus hidup untuk menghapus unggahan multibagian yang tidak lengkap setelah beberapa hari tertentu dengan menggunakan tindakan `AbortIncompleteMultipartUpload`. Untuk informasi selengkapnya tentang membuat aturan siklus hidup untuk menghapus unggahan multibagian yang tidak lengkap, lihat Mengonfigurasi konfigurasi siklus hidup bucket untuk menghapus unggahan multibagian yang tidak lengkap.

Menegakkan penulisan bersyarat di ember Amazon S3

Dengan menggunakan kebijakan bucket Amazon S3, Anda dapat menerapkan penulisan bersyarat untuk unggahan objek di bucket tujuan umum.

Kebijakan bucket adalah kebijakan berbasis sumber daya yang dapat Anda gunakan untuk memberikan izin akses ke bucket Amazon S3 dan objek di dalamnya. Hanya pemilik bucket yang dapat mengaitkan kebijakan dengan bucket. Untuk informasi lebih lanjut tentang kebijakan bucket, lihat the section called “Kebijakan bucket”.

Anda dapat menggunakan tombol kondisi `s3:if-match` atau `s3:if-none-match` sebagai `Condition` elemen opsional atau `Condition` blok untuk menentukan kapan kebijakan berlaku. Untuk unggahan multibagian, Anda harus menentukan kunci `s3:ObjectCreationOperation` kondisi untuk mengecualikan,, dan `UploadPartCopy` operasi `CreateMultipartUploadUploadPart`, karena ini APIs tidak menerima header bersyarat. Untuk informasi selengkapnya tentang penggunaan kondisi dalam kebijakan bucket, lihatthe section called “Contoh kunci kondisi”.

> **Note**
>
> Jika Anda menggunakan kebijakan bucket untuk menerapkan penulisan bersyarat, Anda tidak dapat melakukan operasi penyalinan ke bucket atau awalan yang ditentukan dalam kebijakan bucket Anda. `CopyObject`permintaan tanpa header `If-None-Match` atau `If-Match` HTTP gagal dengan `403 Access Denied` kesalahan. `CopyObject`permintaan yang dibuat dengan header HTTP tersebut gagal dengan `501 Not Implemented` respons.

Contoh berikut menunjukkan cara menggunakan kondisi dalam kebijakan bucket untuk memaksa klien menggunakan header `If-None-Match` atau `If-Match` HTTP.

Topik

- Contoh 1: Hanya izinkan unggahan objek menggunakan PutObject dan CompleteMultipartUpload permintaan yang menyertakan header if-none-match
- Contoh 2: Hanya izinkan unggahan objek menggunakan PutObject dan CompleteMultipartUpload permintaan yang menyertakan header if-match
- Contoh 3: Hanya izinkan permintaan unggah objek yang menyertakan if-none-match atau if-match header

Contoh 1: Hanya izinkan unggahan objek menggunakan **`PutObject`** dan **`CompleteMultipartUpload`** permintaan yang menyertakan header **`if-none-match`**

Kebijakan ini memungkinkan akun 111122223333, pengguna Alice, untuk menulis ke *`amzn-s3-demo-bucket1`* bucket jika permintaan menyertakan `if-none-match` header, memastikan bahwa kunci objek belum ada di bucket. Semua `PutObject` dan `CompleteMultipartUpload` permintaan ke bucket yang ditentukan harus menyertakan `if-none-match` header agar berhasil. Dengan menggunakan header ini, pelanggan dapat menulis ke bucket ini hanya jika kunci objek tidak ada di bucket.

> **Note**
>
> Kebijakan ini juga menetapkan kunci `s3:ObjectCreationOperation` kondisi yang memungkinkan pengunggahan multibagian menggunakan`CreateMultipartUpload`,`UploadPart`, dan. `UploadPartCopy` APIs

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowConditionalPut",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:user/Alice"
            },
            "Action": "s3:PutObject",
```

```
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*",
            "Condition": {
                "Null": {
                    "s3:if-none-match": "false"
                }
            }
        },
        {
            "Sid": "AllowConditionalPutwithMPUs",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:user/Alice"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*",
            "Condition": {
                "Bool": {
                    "s3:ObjectCreationOperation": "false"
                }
            }
        }
    ]
}
```

Contoh 2: Hanya izinkan unggahan objek menggunakan **`PutObject`** dan **`CompleteMultipartUpload`** permintaan yang menyertakan header **`if-match`**

Kebijakan ini memungkinkan akun 111122223333, pengguna Alice untuk menulis *amzn-s3-demo-bucket1* hanya jika permintaan menyertakan header. `if-match` Header ini membandingkan ETag nilai objek di S3 dengan yang Anda berikan selama operasi. `WRITE` Jika ETag nilainya tidak cocok, operasi akan gagal. Semua `PutObject` dan `CompleteMultipartUpload` permintaan ke bucket yang ditentukan harus menyertakan `if-match` header agar berhasil.

> **Note**
>
> Kebijakan ini juga menetapkan kunci `s3:ObjectCreationOperation` kondisi yang memungkinkan pengunggahan multibagian menggunakan`CreateMultipartUpload`,`UploadPart`, dan. `UploadPartCopy` APIs

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutObject",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:user/Alice"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*",
        },
        {
            "Sid": "BlockNonConditionalObjectCreation",
            "Effect": "Deny",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:user/Alice"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*",
            "Condition": {
                "Null": {
                    "s3:if-match": "true"
                },
                "Bool": {
                    "s3:ObjectCreationOperation": "true"
                }
            }
        },
        {
            "Sid": "AllowGetObjectBecauseConditionalPutIfMatchETag",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:user/Alice"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*"
        }
    ]
}
```

Contoh 3: Hanya izinkan permintaan unggah objek yang menyertakan **`if-none-match`** atau **`if-match`** header

Kebijakan ini memungkinkan akun 111122223333, pengguna Alice untuk menulis *amzn-s3-demo-bucket1* jika permintaan menyertakan atau header. `if-none-match` `if-match` Hal ini memungkinkan Alice untuk mengunggah objek jika nama kunci tidak ada di bucket, atau jika nama kunci memang ada, Alice dapat menimpa objek jika objek ETag cocok dengan yang ETag disediakan dalam permintaan. PUT

> **Note**
>
> Kebijakan ini juga menetapkan kunci `s3:ObjectCreationOperation` kondisi yang memungkinkan pengunggahan multibagian menggunakan`CreateMultipartUpload`,`UploadPart`, dan. `UploadPartCopy` APIs

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": " AllowConditionalPutifAbsent",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:user/Alice"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*",
            "Condition": {
                "Null": {
                    "s3:if-none-match": "false"
                }
            }
        },
        {
            "Sid": "AllowConditionalPutIfMatchEtag",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:user/Alice"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*",
```

```
            "Condition": {
                "Null": {
                    "s3:if-match": "false"
                }
            }
        },
        {
            "Sid": "AllowConditionalObjectCreation",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:user/Alice"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*",
            "Condition": {
                "Bool": {
                    "s3:ObjectCreationOperation": "false"
                }
            }
        },
        {
            "Sid": " AllowGetObjectBecauseConditionalPutIfMatchETag",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:user/Alice"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*"
        }
    ]
}
```

Menyalin, memindahkan, dan mengganti nama objek

`CopyObject`Operasi membuat salinan objek yang sudah disimpan di Amazon S3.

Anda dapat membuat salinan objek hingga 5 GB dalam satu operasi atom. Namun, untuk menyalin objek yang lebih besar dari 5 GB, Anda harus menggunakan unggahan multipart menggunakan AWS CLI or AWS SDKs. Untuk informasi selengkapnya, lihat Menyalin objek menggunakan unggahan multibagian.

> **Note**
>
> Untuk mempertahankan manfaat kinerja objek yang Anda unggah menggunakan unggahan multibagian, Anda harus menyalin objek menggunakan unggahan multibagian menggunakan AWS CLI atau AWS SDK alih-alih konsol S3. Untuk informasi selengkapnya, lihat Menyalin objek menggunakan unggahan multibagian.

Dengan menggunakan operasi `CopyObject`, Anda dapat:

- Buat salinan objek tambahan.
- Ganti nama objek dengan menyalinnya dan menghapus yang asli.
- Salin atau pindahkan objek dari satu ember ke ember lainnya, termasuk melintasi Wilayah AWS (misalnya, dari `us-west-1` keeu-`west-2`). Saat Anda memindahkan objek, Amazon S3 menyalin objek ke tujuan yang ditentukan dan kemudian menghapus objek sumber.

  > **Note**
  >
  > Menyalin atau memindahkan objek di seluruh Wilayah AWS menimbulkan biaya bandwidth. Untuk informasi selengkapnya, lihat Harga Amazon S3.

- Ubah metadata objek. Setiap objek Amazon S3 memiliki metadata. Metadata ini adalah satu set pasangan nama-nilai. Anda dapat mengatur metadata objek pada saat Anda mengunggah objek. Setelah mengunggah objek, Anda tidak dapat mengubah metadata objek. Satu-satunya cara untuk memodifikasi metadata objek adalah membuat salinan objek, dan mengatur metadatanya. Untuk melakukannya, dalam operasi penyalinan, atur objek yang sama dengan sumber dan target.

  Beberapa metadata objek adalah metadata sistem dan lainnya ditentukan pengguna. Anda dapat mengontrol beberapa metadata sistem. Misalnya, Anda dapat mengontrol kelas penyimpanan dan jenis enkripsi sisi server yang akan digunakan untuk objek. Saat Anda menyalin sebuah objek, metadata sistem yang dikontrol pengguna dan metadata yang ditentukan pengguna juga disalin. Amazon S3 mengatur ulang metadata yang dikontrol sistem. Misalnya, saat Anda menyalin suatu objek, Amazon S3 mengatur ulang tanggal penciptaan objek yang disalin. Anda tidak perlu menyetel salah satu nilai metadata yang dikendalikan sistem ini dalam permintaan salinan Anda.

  Saat menyalin sebuah objek, Anda mungkin memutuskan untuk memperbarui beberapa nilai metadata. Misalnya, jika objek sumber Anda dikonfigurasi untuk menggunakan penyimpanan Standar S3, Anda dapat memilih untuk menggunakan S3 Intelligent-Tiering untuk salinan objek

tersebut. Anda juga dapat memutuskan untuk mengubah beberapa nilai metadata yang ditentukan pengguna yang terdapat pada objek sumber tersebut. Jika Anda memilih untuk memperbarui salah satu metadata objek yang dapat dikonfigurasi pengguna (sistem atau yang ditentukan pengguna) selama penyalinan, maka Anda harus secara eksplisit menentukan semua metadata yang dapat dikonfigurasi pengguna yang ada pada objek sumber dalam permintaan Anda, bahkan jika Anda mengubahnya hanya satu dari nilai metadata.

> **Note**
>
> Saat menyalin objek menggunakan konsol Amazon S3, Anda mungkin menerima pesan kesalahan “Metadata yang disalin tidak dapat diverifikasi.” Konsol menggunakan header untuk mengambil dan mengatur metadata untuk objek Anda. Jika konfigurasi jaringan atau browser Anda mengubah permintaan jaringan Anda, perilaku ini dapat menyebabkan metadata yang tidak diinginkan (seperti `Cache-Control` header yang dimodifikasi) ditulis ke objek yang Anda salin. Amazon S3 tidak dapat memverifikasi metadata yang tidak diinginkan ini.
> Untuk mengatasi masalah ini, periksa konfigurasi jaringan dan browser Anda untuk memastikan tidak mengubah header, seperti`Cache-Control`. Untuk informasi selengkapnya, lihat Model Tanggung Jawab Bersama.

Untuk informasi selengkapnya tentang pemberian metadata objek, lihat Bekerja dengan metadata objek.

Menyalin objek yang diarsipkan dan dipulihkan

Jika objek sumber diarsipkan di S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive, Anda harus memulihkan salinan sementara terlebih dahulu sebelum dapat menyalin objek ke bucket lain. Untuk informasi tentang pengarsipan objek, lihat Bekerja dengan objek yang diarsipkan.

Operasi Salin di konsol Amazon S3 tidak didukung untuk objek yang dipulihkan di kelas penyimpanan S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive. Untuk menyalin objek yang dipulihkan ini, gunakan AWS Command Line Interface (AWS CLI), the AWS SDKs, atau Amazon S3 REST API.

Menyalin objek terenkripsi

Amazon S3 secara otomatis mengenkripsi semua objek baru yang disalin ke bucket S3. Jika Anda tidak menentukan informasi enkripsi dalam permintaan penyalinan, pengaturan enkripsi objek target

diatur ke konfigurasi enkripsi default bucket tujuan. Secara default, semua bucket memiliki konfigurasi enkripsi tingkat dasar yang menggunakan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3). Jika bucket tujuan memiliki konfigurasi enkripsi default yang menggunakan enkripsi di sisi server dengan kunci AWS Key Management Service (AWS KMS) (SSE-KMS), atau kunci enkripsi yang disediakan pelanggan (SSE-C), Amazon S3 menggunakan kunci KMS yang sesuai, atau kunci yang disediakan pelanggan untuk mengenkripsi salinan objek target.

Saat menyalin objek, jika Anda ingin menggunakan jenis pengaturan enkripsi yang berbeda untuk objek target, Anda dapat meminta Amazon S3 mengenkripsi objek target dengan kunci KMS, kunci terkelola Amazon S3, atau kunci yang disediakan pelanggan. Jika pengaturan enkripsi dalam permintaan Anda berbeda dari konfigurasi enkripsi default bucket tujuan, pengaturan enkripsi dalam permintaan Anda akan lebih diutamakan. Jika objek sumber untuk salinan dienkripsi dengan SSE-C, Anda harus memberikan informasi enkripsi yang diperlukan dalam permintaan Anda sehingga Amazon S3 dapat mendekripsi objek untuk disalin. Untuk informasi selengkapnya, lihat Melindungi data dengan enkripsi.

Menggunakan checksum saat menyalin objek

Saat menyalin objek, Anda dapat memilih untuk menggunakan algoritma checksum yang berbeda untuk objek tersebut. Apakah Anda memilih untuk menggunakan algoritma yang sama atau yang baru, Amazon S3 menghitung nilai checksum baru setelah objek disalin. Amazon S3 tidak secara langsung menyalin nilai checksum. Semua objek yang disalin tanpa checksum dan algoritma checksum tujuan tertentu secara otomatis mendapatkan algoritma checksum. `CRC-64NVME` Untuk informasi selengkapnya tentang bagaimana checksum dihitung, lihat Mengunggah dan menyalin objek menggunakan unggahan multipart di Amazon S3.

Menyalin beberapa objek dalam satu permintaan

Untuk menyalin lebih dari satu objek Amazon S3 dengan satu permintaan, Anda juga dapat menggunakan Operasi Batch S3. Anda menyediakan daftar objek yang akan dioperasikan kepada Operasi Batch S3. Operasi Batch S3 akan memanggil masing-masing operasi API untuk melakukan operasi tertentu. Satu tugas Operasi Batch dapat melakukan operasi tertentu pada miliaran objek yang berisi data sebesar eksabita.

Operasi Batch S3 memiliki fitur melacak progres, mengirimkan notifikasi, dan menyimpan laporan penyelesaian terperinci dari semua tindakan, menyediakan pengalaman yang dikelola sepenuhnya, dapat diaudit, dan nirserver. Anda dapat menggunakan Operasi Batch S3 melalui konsol Amazon S3 AWS CLI, AWS SDKs, atau REST API. Untuk informasi selengkapnya, lihat the section called “Dasar-dasar Operasi Batch”.

Menyalin objek ke ember direktori

Untuk informasi tentang menyalin objek ke bucket direktori, lihatMenyalin objek dari atau ke ember direktori. Untuk informasi tentang menggunakan kelas penyimpanan Amazon S3 Express One Zone dengan bucket direktori, lihat dan. S3 Express One Zone Bekerja dengan bucket direktori

Untuk menyalin objek

Untuk menyalin objek, gunakan metode berikut.

Menggunakan konsol S3

> **Note**
>
> Batasan dan batasan saat Anda menyalin objek dengan konsol adalah sebagai berikut:
>
> - Anda dapat menyalin objek jika objek Anda kurang dari 5 GB. Jika objek Anda lebih besar dari 5 GB, Anda harus menggunakan AWS CLIatau AWS SDKsuntuk menyalin objek.
> - Untuk daftar izin tambahan yang diperlukan untuk menyalin objek, lihatthe section called "Izin yang diperlukan untuk operasi API S3". Misalnya kebijakan yang memberikan izin ini, lihatthe section called "Contoh kebijakan berbasis identitas".
> - `Copy`Tindakan ini berlaku untuk semua objek dalam folder yang ditentukan (awalan). Objek yang ditambahkan ke folder ini saat tindakan sedang berlangsung mungkin terpengaruh.
> - Penyalinan objek Lintas Wilayah yang dienkripsi dengan SSE-KMS tidak didukung oleh konsol Amazon S3. Untuk menyalin objek yang dienkripsi dengan SSE-KMS di seluruh Wilayah, gunakan, AWS SDK AWS CLI, atau Amazon S3 REST API.
> - Objek yang dienkripsi dengan kunci enkripsi yang disediakan pelanggan (SSE-C) tidak dapat disalin dengan menggunakan konsol S3. Untuk menyalin objek yang dienkripsi dengan SSE-C, gunakan, AWS SDK AWS CLI, atau Amazon S3 REST API.
> - Objek yang disalin tidak akan mempertahankan pengaturan Object Lock dari objek asli.
> - Jika bucket tempat Anda menyalin objek menggunakan pengaturan yang diberlakukan pemilik bucket untuk Kepemilikan Objek S3, objek tidak ACLs akan disalin ke tujuan yang ditentukan.
> - Jika Anda ingin menyalin objek ke bucket yang menggunakan setelan diberlakukan pemilik bucket untuk Kepemilikan Objek S3, pastikan bucket sumber juga menggunakan setelan yang diberlakukan pemilik bucket, atau hapus semua hibah ACL objek ke akun dan grup lain. AWS

Untuk menyalin objek

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum atau ember Direktori.
3. Dalam daftar ember, pilih nama bucket yang berisi objek yang ingin Anda salin.
4. Centang kotak di sebelah kiri nama objek yang ingin Anda salin.
5. Pada menu Tindakan, pilih Salin dari daftar opsi yang muncul.
6. Pilih jenis tujuan dan akun tujuan. Untuk menentukan jalur tujuan, pilih Jelajahi S3, navigasi ke tujuan, dan pilih kotak centang di sebelah kiri tujuan. Pilih Pilih tujuan di sudut kanan bawah.

   Atau, masukkan jalur tujuan.
7. Jika Anda tidak mengaktifkan versi bucket, Anda akan melihat peringatan yang merekomendasikan Anda mengaktifkan Bucket Versioning untuk membantu melindungi dari penimpaan atau penghapusan objek secara tidak sengaja. Jika Anda ingin menyimpan semua versi objek dalam bucket ini, pilih Aktifkan Penentuan Versi Bucket. Anda juga dapat melihat enkripsi default dan properti Kunci Objek S3 di Detail tujuan.
8. Di bawah Pengaturan salinan tambahan, pilih apakah Anda ingin Menyalin setelan sumber, Jangan tentukan pengaturan, atau Tentukan pengaturan. Pengaturan sumber salin adalah opsi default. Jika Anda hanya ingin menyalin objek tanpa atribut pengaturan sumber, pilih Jangan tentukan pengaturan. Pilih Tentukan pengaturan untuk menentukan pengaturan untuk kelas penyimpanan, tag objek ACLs, metadata, enkripsi sisi server, dan checksum tambahan.
9. Pilih Salin di sudut kanan bawah. Amazon S3 menyalin objek Anda ke tujuan.

Menggunakan AWS SDKs

Contoh di bagian ini memperlihatkan cara untuk menyalin objek hingga 5 GB dalam satu operasi. Untuk menyalin objek yang lebih besar dari 5 GB, Anda harus menggunakan unggahan multibagian. Untuk informasi selengkapnya, lihat Menyalin objek menggunakan unggahan multibagian.

Java

Example

Contoh berikut menyalin objek di Amazon S3 menggunakan AWS SDK untuk Java. Untuk petunjuk cara membuat dan menguji sampel yang berfungsi, lihat Memulai di Panduan AWS SDK untuk Java Pengembang.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.CopyObjectRequest;

import java.io.IOException;

public class CopyObjectSingleOperation {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String sourceKey = "*** Source object key *** ";
        String destinationKey = "*** Destination object key ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            // Copy the object into a new object in the same bucket.
            CopyObjectRequest copyObjRequest = new CopyObjectRequest(bucketName,
 sourceKey, bucketName, destinationKey);
            s3Client.copyObject(copyObjRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

Contoh C # berikut menggunakan tingkat tinggi SDK untuk .NET untuk menyalin objek yang sebesar 5 GB dalam satu operasi. Untuk objek yang lebih besar dari 5 GB, gunakan contoh salinan unggahan multibagian yang dijelaskan dalam Menyalin objek menggunakan unggahan multibagian.

Contoh ini menampilkan pembuatan salinan objek yang berukuran maksimum sebesar 5 GB. Untuk informasi tentang menyiapkan dan menjalankan contoh kode, lihat Memulai SDK for .NET di AWSAWS SDK for .NET Developer Guide.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class CopyObjectTest
    {
        private const string sourceBucket = "*** provide the name of the bucket with
 source object ***";
        private const string destinationBucket = "*** provide the name of the bucket
 to copy the object to ***";
        private const string objectKey = "*** provide the name of object to copy
 ***";
        private const string destObjectKey = "*** provide the destination object key
 name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            Console.WriteLine("Copying an object");
            CopyingObjectAsync().Wait();
        }

        private static async Task CopyingObjectAsync()
        {
```

```
            try
            {
                CopyObjectRequest request = new CopyObjectRequest
                {
                    SourceBucket = sourceBucket,
                    SourceKey = objectKey,
                    DestinationBucket = destinationBucket,
                    DestinationKey = destObjectKey
                };
                CopyObjectResponse response = await
 s3Client.CopyObjectAsync(request);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
        }
    }
}
```

PHP

Topik ini memandu Anda menggunakan kelas dari versi 3 AWS SDK untuk PHP untuk menyalin satu objek dan beberapa objek dalam Amazon S3, dari satu bucket ke bucket lainnya atau dalam bucket yang sama.

Untuk informasi selengkapnya tentang AWS SDK for Ruby API, AWS buka SDK for Ruby - Versi 2.

Contoh PHP berikut menggambarkan penggunaan `copyObject()` metode untuk menyalin satu objek dalam Amazon S3. Hal ini juga menunjukkan bagaimana untuk membuat beberapa salinan dari objek dengan menggunakan batch panggilan ke `CopyObject` dengan `getcommand()` metode.

Menyalin objek

1	Buat instans dari klien Amazon S3 dengan menggunakan konstruktor kelas `Aws\S3\S3Client` .
2	Untuk membuat beberapa salinan objek, Anda menjalankan sekumpulan panggilan ke getCommand()metode klien Amazon S3, yang diwarisi dari kelas. Aws\CommandInterface Anda menyediakan perintah `CopyObject` sebagai argumen pertama dan array yang berisi bucket sumber, nama kunci sumber, bucket target, dan nama kunci target sebagai argumen kedua.

```
 require 'vendor/autoload.php';

use Aws\CommandPool;
use Aws\Exception\AwsException;
use Aws\ResultInterface;
use Aws\S3\S3Client;

$sourceBucket = '*** Your Source Bucket Name ***';
$sourceKeyname = '*** Your Source Object Key ***';
$targetBucket = '*** Your Target Bucket Name ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region' => 'us-east-1'
]);

// Copy an object.
$s3->copyObject([
    'Bucket' => $targetBucket,
    'Key' => "$sourceKeyname-copy",
    'CopySource' => "$sourceBucket/$sourceKeyname",
]);

// Perform a batch of CopyObject operations.
$batch = array();
for ($i = 1; $i <= 3; $i++) {
    $batch[] = $s3->getCommand('CopyObject', [
        'Bucket' => $targetBucket,
        'Key' => "{targetKeyname}-$i",
        'CopySource' => "$sourceBucket/$sourceKeyname",
```

```
    ]);
}
try {
    $results = CommandPool::batch($s3, $batch);
    foreach ($results as $result) {
        if ($result instanceof ResultInterface) {
            // Result handling here
        }
        if ($result instanceof AwsException) {
            // AwsException handling here
        }
    }
} catch (Exception $e) {
    // General error handling here
}
```

Python

```
class ObjectWrapper:
    """Encapsulates S3 object actions."""

    def __init__(self, s3_object):
        """
        :param s3_object: A Boto3 Object resource. This is a high-level resource in
 Boto3
                          that wraps object actions in a class-like structure.
        """
        self.object = s3_object
        self.key = self.object.key
```

```
    def copy(self, dest_object):
        """
        Copies the object to another bucket.

        :param dest_object: The destination object initialized with a bucket and
 key.
                            This is a Boto3 Object resource.
        """
        try:
            dest_object.copy_from(
```

```
                CopySource={"Bucket": self.object.bucket_name, "Key":
self.object.key}
            )
            dest_object.wait_until_exists()
            logger.info(
                "Copied object from %s:%s to %s:%s.",
                self.object.bucket_name,
                self.object.key,
                dest_object.bucket_name,
                dest_object.key,
            )
        except ClientError:
            logger.exception(
                "Couldn't copy object from %s/%s to %s/%s.",
                self.object.bucket_name,
                self.object.key,
                dest_object.bucket_name,
                dest_object.key,
            )
            raise
```

Ruby

Tugas berikut memandu Anda menggunakan Ruby class untuk menyalin objek di Amazon S3 dari satu bucket ke bucket lainnya atau dalam bucket yang sama.

Menyalin objek

1	Gunakan permata termodulasi Amazon S3 untuk versi 3 dari, require AWS SDK untuk Ruby`aws-sdk-s3` , dan berikan kredensyal Anda. AWS Untuk informasi selengkapnya tentang cara memberikan kredensil, lihat Membuat permintaan menggunakan kredensil pengguna AWS akun atau IAM di Referensi API Amazon S3.
2	Berikan informasi permintaan, seperti nama bucket sumber, nama kunci sumber, nama bucket tujuan, dan kunci tujuan.

Contoh Ruby kode berikut menunjukkan tugas-tugas sebelumnya dengan menggunakan `#copy_object` metode untuk menyalin objek dari satu ember ke ember lainnya.

```
require 'aws-sdk-s3'

# Wraps Amazon S3 object actions.
class ObjectCopyWrapper
  attr_reader :source_object

  # @param source_object [Aws::S3::Object] An existing Amazon S3 object. This is
 used as the source object for
  #                                       copy actions.
  def initialize(source_object)
    @source_object = source_object
  end

  # Copy the source object to the specified target bucket and rename it with the
 target key.
  #
  # @param target_bucket [Aws::S3::Bucket] An existing Amazon S3 bucket where the
 object is copied.
  # @param target_object_key [String] The key to give the copy of the object.
  # @return [Aws::S3::Object, nil] The copied object when successful; otherwise,
 nil.
  def copy_object(target_bucket, target_object_key)
    @source_object.copy_to(bucket: target_bucket.name, key: target_object_key)
    target_bucket.object(target_object_key)
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't copy #{@source_object.key} to #{target_object_key}. Here's why:
 #{e.message}"
  end
end

# Example usage:
def run_demo
  source_bucket_name = "amzn-s3-demo-bucket1"
  source_key = "my-source-file.txt"
  target_bucket_name = "amzn-s3-demo-bucket2"
  target_key = "my-target-file.txt"

  source_bucket = Aws::S3::Bucket.new(source_bucket_name)
  wrapper = ObjectCopyWrapper.new(source_bucket.object(source_key))
  target_bucket = Aws::S3::Bucket.new(target_bucket_name)
  target_object = wrapper.copy_object(target_bucket, target_key)
  return unless target_object
```

```
  puts "Copied #{source_key} from #{source_bucket_name} to
 #{target_object.bucket_name}:#{target_object.key}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Penggunaan API REST

Contoh ini menjelaskan cara menyalin objek dengan menggunakan Amazon S3 REST API. Untuk informasi selengkapnya tentang API REST, lihat CopyObject.

Contoh ini menyalin objek `flotsam` dari bucket *amzn-s3-demo-bucket1* ke objek `jetsam` dari bucket *amzn-s3-demo-bucket2*, dengan mempertahankan metadatanya.

```
PUT /jetsam HTTP/1.1
Host: amzn-s3-demo-bucket2.s3.amazonaws.com
x-amz-copy-source: /amzn-s3-demo-bucket1/flotsam
Authorization: AWS AKIAIOSFODNN7EXAMPLE:ENoSbxYByFA0UGLZUqJN5EUnLDg=
Date: Wed, 20 Feb 2008 22:12:21 +0000
```

Tanda tangannya dihasilkan dari informasi berikut.

```
PUT\r\n
\r\n
\r\n
Wed, 20 Feb 2008 22:12:21 +0000\r\n

x-amz-copy-source:/amzn-s3-demo-bucket1/flotsam\r\n
/amzn-s3-demo-bucket2/jetsam
```

Amazon S3 mengembalikan respons berikut yang menentukan ETag objek dan kapan terakhir diubah.

```
HTTP/1.1 200 OK
x-amz-id-2: Vyaxt7qEbzv34BnSu5hctyyNSlHTYZFMWK4FtzO+iX8JQNyaLdTshL0KxatbaOZt
x-amz-request-id: 6B13C3C5B34AF333
Date: Wed, 20 Feb 2008 22:13:01 +0000

Content-Type: application/xml
Transfer-Encoding: chunked
Connection: close
```

```
Server: AmazonS3
<?xml version="1.0" encoding="UTF-8"?>

<CopyObjectResult>
   <LastModified>2008-02-20T22:13:01</LastModified>
   <ETag>"7e9c608af58950deeb370c98608ed097"</ETag>
</CopyObjectResult>
```

Menggunakan AWS CLI

Anda juga dapat menggunakan AWS Command Line Interface (AWS CLI) untuk menyalin objek S3. Untuk informasi selengkapnya, lihat copy-object dalam AWS CLI Referensi Perintah.

Untuk informasi tentang AWS CLI, lihat Apa itu AWS Command Line Interface? dalam AWS Command Line Interface User Guide.

Untuk memindahkan objek

Untuk memindahkan objek, gunakan metode berikut.

Menggunakan konsol S3

> **Note**
>
> - Anda dapat memindahkan objek jika objek Anda kurang dari 5 GB. Jika objek Anda lebih besar dari 5 GB, Anda harus menggunakan AWS CLIatau AWS SDKsuntuk memindahkan objek.
> - Untuk daftar izin tambahan yang diperlukan untuk memindahkan objek, lihatthe section called “Izin yang diperlukan untuk operasi API S3”. Misalnya kebijakan yang memberikan izin ini, lihatthe section called “Contoh kebijakan berbasis identitas”.
> - Objek yang dienkripsi dengan kunci enkripsi yang disediakan pelanggan (SSE-C) tidak dapat dipindahkan dengan menggunakan konsol Amazon S3. Untuk memindahkan objek yang dienkripsi dengan SSE-C, gunakan, atau Amazon S3 REST API. AWS CLI AWS SDKs
> - Saat memindahkan folder, tunggu operasi Pindahkan selesai sebelum membuat perubahan tambahan di folder.
> - Anda tidak dapat menggunakan alias titik akses S3 sebagai sumber atau tujuan untuk operasi Pindah di konsol Amazon S3.

Untuk memindahkan objek

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket. Navigasikan ke bucket atau folder Amazon S3 yang berisi objek yang ingin Anda pindahkan.
3. Pilih kotak centang untuk objek yang ingin Anda pindahkan.
4. Pada menu Tindakan, pilih Pindahkan.
5. Untuk menentukan jalur tujuan, pilih Browse S3, navigasikan ke tujuan, dan pilih kotak centang tujuan. Pilih Pilih tujuan.

   Atau, masukkan jalur tujuan.
6. Jika Anda tidak mengaktifkan versi bucket, Anda akan melihat peringatan yang merekomendasikan Anda mengaktifkan Bucket Versioning untuk membantu melindungi dari penimpaan atau penghapusan objek secara tidak sengaja. Jika Anda ingin menyimpan semua versi objek dalam bucket ini, pilih Aktifkan Penentuan Versi Bucket. Anda juga dapat melihat enkripsi default dan properti Object Lock di Detail tujuan.
7. Di bawah Pengaturan salinan tambahan, pilih apakah Anda ingin Menyalin setelan sumber, Jangan tentukan pengaturan, atau Tentukan pengaturan. Pengaturan sumber salin adalah opsi default. Jika Anda hanya ingin menyalin objek tanpa atribut pengaturan sumber, pilih Jangan tentukan pengaturan. Pilih Tentukan pengaturan untuk menentukan pengaturan untuk kelas penyimpanan, tag objek ACLs, metadata, enkripsi sisi server, dan checksum tambahan.
8. Pilih Pindahkan di sudut kanan bawah. Amazon S3 memindahkan objek Anda ke tujuan.

Menggunakan AWS CLI

Anda juga dapat menggunakan AWS Command Line Interface (AWS CLI) untuk memindahkan objek S3. Untuk informasi selengkapnya, lihat mv dalam AWS CLI Referensi Perintah.

Untuk informasi tentang AWS CLI, lihat Apa itu AWS Command Line Interface? dalam AWS Command Line Interface User Guide.

Untuk mengganti nama objek

Untuk mengganti nama objek, gunakan prosedur berikut.

Note

- Anda dapat mengganti nama objek jika objek Anda kurang dari 5 GB. Untuk mengganti nama objek yang lebih besar dari 5 GB, Anda harus menggunakan AWS CLIatau AWS SDKsuntuk menyalin objek Anda dengan nama baru dan kemudian menghapus objek asli.
- Untuk daftar izin tambahan yang diperlukan untuk menyalin objek, lihatthe section called "Izin yang diperlukan untuk operasi API S3". Misalnya kebijakan yang memberikan izin ini, lihatthe section called "Contoh kebijakan berbasis identitas".
- Mengganti nama objek membuat salinan objek dengan tanggal modifikasi terakhir baru, dan kemudian menambahkan penanda hapus ke objek asli.
- Pengaturan bucket untuk enkripsi default diterapkan secara otomatis ke objek tertentu yang tidak dienkripsi.
- Anda tidak dapat menggunakan konsol Amazon S3 untuk mengganti nama objek dengan kunci enkripsi yang disediakan pelanggan (SSE-C). Untuk mengganti nama objek yang dienkripsi dengan SSE-C, gunakan, atau Amazon S3 REST API untuk menyalin objek AWS SDKs tersebut dengan nama baru. AWS CLI
- Jika bucket ini menggunakan setelan yang diberlakukan pemilik bucket untuk Kepemilikan Objek S3, daftar kontrol akses objek (ACLs) tidak akan disalin.

Untuk mengganti nama objek

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi, pilih Bucket, lalu pilih tab Bucket tujuan umum. Arahkan ke bucket Amazon S3 atau folder yang berisi objek yang ingin Anda ganti namanya.
3. Pilih kotak centang untuk objek yang ingin Anda ganti namanya.
4. Pada menu Tindakan, pilih Ganti nama objek.
5. Di kotak Nama objek baru, masukkan nama baru untuk objek tersebut.
6. Di bawah Pengaturan salinan tambahan, pilih apakah Anda ingin Menyalin setelan sumber, Jangan tentukan pengaturan, atau Tentukan pengaturan. Pengaturan sumber salin adalah opsi default. Jika Anda hanya ingin menyalin objek tanpa atribut pengaturan sumber, pilih Jangan tentukan pengaturan. Pilih Tentukan pengaturan untuk menentukan pengaturan untuk kelas penyimpanan, tag objek ACLs, metadata, enkripsi sisi server, dan checksum tambahan.

7. Pilih Simpan perubahan. Amazon S3 mengganti nama objek Anda.

Mengunggah objek

Bagian ini menjelaskan cara untuk mengunduh objek dari bucket Amazon S3. Dengan Amazon S3, Anda dapat menyimpan objek dalam satu atau lebih bucket, dan setiap objek dapat berukuran hingga 5 TB. Objek Amazon S3 apa pun yang tidak diarsipkan dapat diakses secara real time. Namun, objek yang diarsipkan harus dipulihkan sebelum dapat diunduh. Untuk informasi tentang objek yang diarsipkan, lihat the section called “Mengunduh objek yang diarsipkan”.

Anda dapat mengunduh satu objek dengan menggunakan konsol Amazon S3, AWS Command Line Interface (AWS CLI) AWS SDKs, atau Amazon S3 REST API. Untuk mengunduh objek dari S3 tanpa menulis kode atau menjalankan perintah apa pun, gunakan konsol S3. Untuk informasi selengkapnya, lihat the section called “Mengunduh objek”.

Untuk mengunduh beberapa objek AWS CloudShell, gunakan AWS CLI, atau file AWS SDKs. Untuk informasi selengkapnya, lihat the section called “Mengunduh beberapa objek”.

Jika Anda perlu mengunduh bagian dari suatu objek, Anda menggunakan parameter tambahan dengan AWS CLI atau REST API untuk menentukan hanya byte yang ingin Anda unduh. Untuk informasi selengkapnya, lihat the section called “Mengunduh bagian dari suatu objek”.

Jika Anda perlu mengunduh objek yang bukan milik Anda, mintalah pemilik objek untuk membuat URL yang telah ditentukan sebelumnya yang memungkinkan Anda mengunduh objek tersebut. Untuk informasi selengkapnya, lihat the section called “Mengunduh objek dari Akun AWS yang lain”.

Saat Anda mengunduh objek di luar AWS jaringan, biaya transfer data berlaku. Transfer data dalam AWS jaringan gratis dalam hal yang sama Wilayah AWS, tetapi Anda akan dikenakan biaya untuk `GET` permintaan apa pun. Untuk informasi selengkapnya tentang biaya transfer data dan pengambilan data, lihat Harga Amazon S3.

Topik

- Mengunduh objek
- Mengunduh beberapa objek
- Mengunduh bagian dari suatu objek
- Mengunduh objek dari Akun AWS yang lain
- Mengunduh objek yang diarsipkan

- Mengunduh objek berdasarkan metadata
- Memecahkan masalah pengunduhan objek

Mengunduh objek

Anda dapat mengunduh objek dengan menggunakan konsol Amazon S3, AWS CLI AWS SDKs, atau REST API.

Menggunakan konsol S3

Bagian ini menjelaskan cara menggunakan konsol Amazon S3 untuk mengunduh objek dari bucket S3.

> **Note**
>
> - Anda hanya dapat mengunduh satu objek dalam satu waktu.
> - Jika Anda menggunakan konsol Amazon S3 untuk mengunduh objek yang nama kuncinya diakhiri dengan titik (`.`), titik tersebut dihapus dari nama kunci objek yang diunduh. Untuk mempertahankan periode di akhir nama objek yang diunduh, Anda harus menggunakan AWS Command Line Interface (AWS CLI) AWS SDKs, atau Amazon S3 REST API.

Untuk mengunduh objek dari bucket S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum atau ember Direktori.
3. Dalam daftar bucket, pilih nama bucket tempat Anda ingin mengunduh objek.
4. Anda dapat mengunduh objek dari bucket S3 dengan cara berikut:
   - Pilih kotak centang di samping objek, dan pilih Unduh. Jika Anda ingin mengunduh objek ke folder tertentu, pada menu Tindakan, pilih Unduh sebagai.
   - Jika Anda ingin mengunduh versi objek tertentu, aktifkan Tampilkan versi (terletak di samping kotak pencarian). Centang kotak di samping versi objek yang Anda inginkan, dan pilih Unduh. Jika Anda ingin mengunduh objek ke folder tertentu, pada menu Tindakan, pilih Unduh sebagai.

Menggunakan AWS CLI

Contoh perintah `get-object` berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk mengunduh objek dari Amazon S3. Perintah ini mendapatkan objek *`folder/my_image`* dari bucket *`amzn-s3-demo-bucket1`*. Anda harus menyertakan`outfile`, yang merupakan nama file untuk objek yang diunduh, seperti*`my_downloaded_image.jpg`*.

```
aws s3api get-object --bucket amzn-s3-demo-bucket1 --key folder/
my_image my_downloaded_image.jpg
```

Untuk informasi dan contoh selengkapnya, lihat get-object di AWS CLI Referensi Perintah.

Menggunakan AWS SDKs

Untuk contoh cara mengunduh objek dengan AWS SDKs, lihat Contoh kode di Referensi API Amazon S3.

Untuk informasi umum tentang penggunaan yang berbeda AWS SDKs, lihat Mengembangkan dengan Amazon S3 menggunakan Referensi API Amazon S3 AWS SDKs di Amazon S3.

Penggunaan API REST

Anda dapat menggunakan API REST untuk mengambil objek dari Amazon S3. Untuk informasi selengkapnya, lihat GetObject dalam Referensi API Amazon Simple Storage Service.

Mengunduh beberapa objek

Anda dapat mengunduh beberapa objek dengan menggunakan AWS CloudShell, file AWS CLI, atau file AWS SDKs.

Menggunakan AWS CloudShell di AWS Management Console

AWS CloudShell adalah shell berbasis browser dan pra-otentikasi yang dapat Anda luncurkan langsung dari file. AWS Management Console

Untuk informasi lebih lanjut tentang AWS CloudShell, lihat Apa itu CloudShell? dalam AWS CloudShell User Guide.

> **Important**
>
> Dengan AWS CloudShell, direktori home Anda memiliki penyimpanan hingga 1GB per Wilayah AWS. Maka, Anda tidak dapat menyinkronkan bucket dengan objek yang berjumlah

lebih dari jumlah ini. Untuk pembatasan yang lebih lengkap, lihat Kuota layanan dan batasan di AWS CloudShell Panduan Pengguna.

Untuk mengunduh objek dengan menggunakan AWS CloudShell

1. Masuk ke AWS Management Console dan buka CloudShell konsol di https://console.aws.amazon.com/cloudshell/.
2. Jalankan perintah berikut untuk menyinkronkan objek di bucket Anda CloudShell. Perintah berikut menyinkronkan objek dari bucket bernama *amzn-s3-demo-bucket1* dan membuat folder bernama*temp*. CloudShell CloudShell menyinkronkan objek Anda ke folder ini. Untuk menggunakan perintah ini, ganti *user input placeholders* dengan informasi Anda sendiri.

   ```
   aws s3 sync s3://amzn-s3-demo-bucket1 ./temp
   ```

   > **Note**
   >
   > `sync`Perintah ini tidak kompatibel dengan bucket direktori.
   > Untuk melakukan pencocokan pola untuk mengecualikan atau menyertakan objek tertentu, Anda dapat menggunakan parameter `--exclude "`*value*`"` dan `--include "`*value*`"` dengan perintah `sync`.

3. Jalankan perintah berikut untuk meng-zip objek Anda di folder bernama *temp* ke file bernama *temp.zip*.

   ```
   zip temp.zip -r temp/
   ```

4. Pilih Tindakan, lalu pilih Unduh file.
5. Masukkan nama file **`temp.zip`** lalu pilih Unduh.
6. (Opsional) Hapus *temp.zip* file dan objek yang disinkronkan ke *temp* folder di CloudShell. Dengan AWS CloudShell, Anda memiliki penyimpanan persisten hingga 1 GB untuk masing-masing Wilayah AWS.

   Anda dapat menggunakan contoh perintah berikut untuk menghapus file `.zip` dan folder Anda. Untuk menggunakan contoh perintah ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
rm temp.zip && rm -rf temp/
```

Menggunakan AWS CLI

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk men-download semua file atau objek di bawah direktori tertentu atau awalan. Perintah ini menyalin semua objek dari bucket *amzn-s3-demo-bucket1* ke direktori Anda saat ini. Untuk menggunakan perintah contoh ini, gunakan nama bucket Anda sebagai pengganti *amzn-s3-demo-bucket1*.

```
aws s3 cp s3://amzn-s3-demo-bucket1 . --recursive
```

Perintah berikut mengunduh semua objek di bawah prefiks *logs* di bucket *amzn-s3-demo-bucket1* ke direktori Anda saat ini. Ini juga menggunakan parameter `--exclude` dan `--include` untuk menyalin hanya objek dengan sufiks *.log*. Untuk menggunakan contoh perintah ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3 cp s3://amzn-s3-demo-bucket1/logs/ . --recursive --exclude "*" --include "*.log"
```

Untuk informasi dan contoh selengkapnya, lihat cp di AWS CLI Referensi Perintah.

Menggunakan AWS SDKs

Untuk contoh cara mengunduh semua objek di bucket Amazon S3 dengan AWS SDKs, lihat Contoh kode di Referensi API Amazon S3.

Untuk informasi umum tentang penggunaan yang berbeda AWS SDKs, lihat Mengembangkan dengan Amazon S3 menggunakan Referensi API Amazon S3 AWS SDKs di Amazon S3.

Mengunduh bagian dari suatu objek

Anda dapat mengunduh bagian dari suatu objek dengan menggunakan AWS CLI atau REST API. Untuk melakukannya, Anda menggunakan parameter tambahan untuk menentukan bagian mana dari objek yang ingin Anda unduh.

Menggunakan AWS CLI

Perintah contoh berikut ini melakukan permintaan `GET` untuk rentang byte dalam objek bernama *folder/my_data* dalam bucket yang bernama *amzn-s3-demo-bucket1*. Dalam permintaan

tersebut, rentang byte harus diawali dengan `bytes=`. Objek parsial diunduh ke file output bernama *`my_data_range`*. Untuk menggunakan contoh perintah ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3api get-object --bucket amzn-s3-demo-bucket1 --key folder/my_data --range
 bytes=0-500 my_data_range
```

Untuk informasi dan contoh selengkapnya, lihat get-object di AWS CLI Referensi Perintah.

Untuk informasi selengkapnya tentang header `Range` HTTP, lihat RFC 9110 di situs web RFC Editor.

> **Note**
>
> Amazon S3 tidak mendukung pengambilan beberapa rentang data dalam satu permintaan `GET`.

Penggunaan API REST

Anda dapat menggunakan parameter `partNumber` dan `Range` di API REST untuk mengambil bagian objek dari Amazon S3. Untuk informasi selengkapnya, lihat GetObject dalam Referensi API Amazon Simple Storage Service.

Mengunduh objek dari Akun AWS yang lain

Anda dapat menggunakan URL yang telah ditandatangani sebelumnya untuk memberi orang lain akses yang dibatasi waktu ke objek Anda tanpa memperbarui kebijakan bucket Anda.

URL yang telah ditandatangani sebelumnya dapat dimasukkan di dalam browser, atau digunakan oleh program untuk mengunduh objek. Kredensi yang digunakan oleh URL adalah milik AWS pengguna yang membuat URL. Setelah URL dibuat, siapa pun yang memiliki URL yang telah ditentukan sebelumnya dapat mengunduh objek terkait hingga URL tersebut kedaluwarsa.

Menggunakan URL yang telah ditandatangani sebelumnya di konsol S3

Anda dapat menggunakan konsol Amazon S3 untuk membuat URL yang telah ditetapkan sebelumnya untuk berbagi objek dalam bucket tujuan umum dengan mengikuti langkah-langkah berikut. Saat menggunakan konsol, waktu kedaluwarsa maksimum untuk URL yang telah ditandatangani sebelumnya adalah 12 jam dari waktu pembuatan.

Untuk menghasilkan URL yang telah ditandatangani sebelumnya menggunakan konsol Amazon S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Dalam daftar bucket, pilih nama bucket yang berisi objek yang Anda inginkan untuk URL yang telah ditetapkan sebelumnya.
4. Dalam daftar Objek, pilih objek yang ingin Anda buatkan URL yang telah ditandatangani.
5. Pada menu Tindakan objek, pilih Bagikan dengan URL yang telah ditandatangani.
6. Tentukan berapa lama Anda menginginkan URL yang telah ditandatangani tersebut valid.
7. Pilih Buat URL yang telah ditandatangani.
8. Ketika pesan konfirmasi muncul, URL secara otomatis disalin ke clipboard Anda. Anda akan melihat tombol untuk menyalin URL yang telah ditandatangani jika Anda perlu menyalinnya lagi.
9. Untuk mengunduh objek, tempel URL ke browser apa pun, dan objek akan mencoba mengunduh.

Untuk informasi selengkapnya tentang presigned URLs dan metode lain untuk membuatnya, lihatUnduh dan unggah objek dengan presigned URLs.

Mengunduh objek yang diarsipkan

Untuk mengurangi biaya penyimpanan objek yang jarang diakses, Anda dapat mengarsipkan objek tersebut. Ketika Anda mengarsipkan objek, objek tersebut dipindahkan ke penyimpanan berbiaya rendah, yang berarti Anda tidak dapat mengaksesnya secara real time. Untuk mengunduh objek yang diarsipkan, Anda harus memulihkannya terlebih dahulu.

Anda dapat memulihkan objek yang diarsipkan dalam hitungan menit atau jam, tergantung pada kelas penyimpanan. Anda dapat memulihkan objek yang diarsipkan menggunakan konsol Amazon S3, Operasi Batch S3, API REST Amazon S3, AWS SDKs the, dan (). AWS Command Line Interface AWS CLI

Untuk petunjuk, lihat Memulihkan objek yang diarsipkan. Setelah Anda memulihkan objek yang diarsipkan, Anda dapat mengunduhnya.

Mengunduh objek berdasarkan metadata

Anda dapat menambahkan prasyarat untuk mengunduh objek berdasarkan metadatanya menggunakan permintaan baca bersyarat. Anda dapat mengembalikan objek berdasarkan tag Entity (ETag) atau tanggal modifikasi terakhir. Ini dapat membatasi operasi S3 ke objek yang diperbarui sejak tanggal tertentu atau hanya mengembalikan versi objek tertentu.

Anda dapat menggunakan penulisan bersyarat untuk GetObjectatau HeadObjectpermintaan.

Untuk informasi selengkapnya tentang permintaan bersyarat, lihat,Tambahkan prasyarat ke operasi S3 dengan permintaan bersyarat.

Memecahkan masalah pengunduhan objek

Izin yang tidak memadai atau kebijakan pengguna bucket atau AWS Identity and Access Management (IAM) yang salah dapat menyebabkan kesalahan saat Anda mencoba mengunduh objek dari Amazon S3. Masalah ini sering dapat menyebabkan kesalahan Akses Ditolak (403 Dilarang). di mana Amazon S3 tidak bisa memberikan akses ke sumber daya.

Untuk mengetahui penyebab umum kesalahan Akses Ditolak (403 Forbidden), lihat Memecahkan masalah kesalahan akses ditolak (403 Forbidden) di Amazon S3.

Jika Anda mendapatkan NoSuchKey kesalahan 404 saat mencoba mengakses objek, lihat Bagaimana saya bisa memecahkan masalah kesalahan 404 dari NoSuchKey Amazon S3? di pusat AWS re:Post pengetahuan.

Memeriksa integritas objek di Amazon S3

Amazon S3 menggunakan nilai checksum untuk memverifikasi integritas data yang Anda unggah atau unduh. Selain itu, Anda dapat meminta agar nilai checksum lain dihitung untuk objek apa pun yang Anda simpan di Amazon S3. Anda dapat memilih algoritma checksum yang akan digunakan saat mengunggah, menyalin, atau menyalin data secara batch.

Saat mengunggah data, Amazon S3 menggunakan algoritme yang Anda pilih untuk menghitung checksum di sisi server dan memvalidasinya dengan nilai yang diberikan sebelum menyimpan objek dan menyimpan checksum sebagai bagian dari metadata objek. Validasi ini bekerja secara konsisten di seluruh mode enkripsi, ukuran objek, dan kelas penyimpanan untuk unggahan satu bagian dan multibagian. Namun, ketika Anda menyalin atau menyalin data secara batch, Amazon S3 menghitung checksum pada objek sumber dan memindahkannya ke objek tujuan.

> **Note**
>
> Saat Anda melakukan unggahan satu bagian atau multibagian, Anda dapat secara opsional menyertakan checksum yang telah dihitung sebelumnya sebagai bagian dari permintaan Anda, dan menggunakan jenis checksum objek lengkap. Untuk menggunakan nilai yang telah dihitung sebelumnya dengan beberapa objek, gunakan AWS CLI atau AWS SDKs.

Menggunakan algoritma checksum yang didukung

Dengan Amazon S3, Anda dapat memilih algoritma checksum untuk memvalidasi data Anda selama pengunggahan. Algoritma checksum yang ditentukan kemudian disimpan dengan objek Anda dan dapat digunakan untuk memvalidasi integritas data selama unduhan. Anda dapat memilih salah satu algoritma checksum Secure Hash Algorithms (SHA) atau Cyclic Redundancy Check (CRC) berikut untuk menghitung nilai checksum:

- CRC-64/NVME () `CRC64NVME`
- CRC-32 () `CRC32`
- CRC-32C () `CRC32C`
- SHA-1 () `SHA1`
- SHA-256 () `SHA256`

Selain itu, Anda dapat memberikan checksum dengan setiap permintaan menggunakan MD5 header Content-.

Saat Anda mengunggah objek, Anda menentukan algoritme yang ingin Anda gunakan:

- Saat Anda menggunakan AWS Management Console, pilih algoritma checksum yang ingin Anda gunakan. Anda dapat secara opsional menentukan nilai checksum objek. Ketika Amazon S3 menerima objek, itu menghitung checksum dengan menggunakan algoritma yang Anda tentukan. Jika kedua nilai checksum tidak cocok, Amazon S3 menghasilkan kesalahan.
- Saat Anda menggunakan SDK, perhatikan hal berikut:
  - Atur `ChecksumAlgorithm` parameter ke algoritme yang Anda inginkan Amazon S3 gunakan. Jika Anda sudah memiliki checksum yang telah dihitung sebelumnya, Anda meneruskan nilai checksum ke AWS SDK, dan SDK menyertakan nilai dalam permintaan. Jika Anda tidak meneruskan nilai checksum atau tidak menentukan algoritma checksum, SDK secara otomatis

menghitung nilai checksum untuk Anda dan menyertakannya dengan permintaan untuk memberikan perlindungan integritas. Jika nilai checksum individual tidak cocok dengan nilai set algoritma checksum, Amazon S3 gagal permintaan dengan kesalahan. `BadDigest`

- Jika Anda menggunakan SDK yang ditingkatkan, AWS SDK akan memilih algoritma checksum untuk Anda. Namun, Anda dapat mengganti algoritma checksum ini.
- Jika Anda tidak menentukan algoritma checksum dan SDK juga tidak menghitung checksum untuk Anda, maka S3 secara otomatis memilih algoritma checksum CRC-64/NVME (). `CRC64NVME`

- Saat Anda menggunakan REST API, jangan gunakan `x-amz-sdk-checksum-algorithm` parameternya. Sebagai gantinya, gunakan salah satu header khusus algoritme (misalnya,). `x-amz-checksum-crc32`

Untuk menerapkan salah satu nilai checksum ini ke objek yang sudah diunggah ke Amazon S3, Anda dapat menyalin objek dan menentukan apakah Anda ingin menggunakan algoritma checksum yang ada atau yang baru. Jika Anda tidak menentukan algoritma, S3 menggunakan algoritma yang ada. Jika objek sumber tidak memiliki algoritma checksum atau nilai checksum tertentu, Amazon S3 menggunakan algoritma CRC-64/NVME untuk menghitung nilai checksum untuk objek tujuan. Anda juga dapat menentukan algoritma checksum saat menyalin objek menggunakan Operasi Batch S3.

> **Important**
>
> Jika Anda menggunakan unggahan multibagian dengan Checksum untuk checksum komposit (atau tingkat bagian), nomor bagian unggahan multibagian harus berurutan dan dimulai dengan 1. Jika Anda mencoba menyelesaikan permintaan unggahan multibagian dengan nomor bagian yang tidak berurutan, Amazon S3 menghasilkan kesalahan. `HTTP 500 Internal Server`

Objek lengkap dan jenis checksum komposit

Di Amazon S3, ada dua jenis checksum yang didukung:

- Checksum objek lengkap: Checksum objek lengkap dihitung berdasarkan semua konten unggahan multipart, yang mencakup semua data dari byte pertama bagian pertama hingga byte terakhir dari bagian terakhir.

> **Note**
>
> Semua permintaan PUT memerlukan jenis checksum objek lengkap.

- Checksum komposit: Checksum komposit dihitung berdasarkan checksum individual dari setiap bagian dalam unggahan multibagian. Alih-alih menghitung checksum berdasarkan semua konten data, pendekatan ini menggabungkan checksum tingkat suku cadang (dari bagian pertama hingga yang terakhir) untuk menghasilkan checksum gabungan tunggal untuk objek lengkap.

> **Note**
>
> Ketika sebuah objek diunggah sebagai unggahan multipart, tag entitas (ETag) untuk objek bukanlah MD5 intisari dari seluruh objek. Sebagai gantinya, Amazon S3 menghitung MD5 intisari setiap bagian saat diunggah. MD5 Intisari digunakan ETag untuk menentukan objek akhir. Amazon S3 menggabungkan byte untuk MD5 intisari bersama-sama dan kemudian menghitung intisari nilai gabungan ini. MD5 Selama langkah ETag pembuatan terakhir, Amazon S3 menambahkan tanda hubung dengan jumlah total bagian sampai akhir.

Amazon S3 mendukung objek lengkap berikut dan jenis algoritma checksum komposit:

- CRC-64/NVME (`CRC64NVME`): Mendukung tipe algoritma objek lengkap saja.
- CRC-32 (`CRC32`): Mendukung objek penuh dan jenis algoritma komposit.
- CRC-32C (`CRC32C`): Mendukung objek penuh dan jenis algoritma komposit.
- SHA-1 (`SHA1`): Mendukung objek penuh dan jenis algoritma komposit.
- SHA-256 (`SHA256`): Mendukung objek penuh dan jenis algoritma komposit.

> **Note**
>
> Jika Anda menggunakan algoritma `SHA-1` atau `SHA-256` checksum untuk upload multipart, Anda harus menggunakan jenis algoritma komposit. Jika Anda menggunakan algoritma `SHA-1` atau `SHA-256` checksum untuk satu bagian upload, hanya tipe algoritma objek lengkap yang didukung.

Unggahan satu bagian

Checksum objek yang diunggah dalam satu bagian (menggunakan `PutObject`) diperlakukan sebagai checksum objek lengkap. Saat mengunggah objek di konsol Amazon S3, Anda dapat memilih algoritma checksum yang ingin digunakan S3 dan juga (opsional) memberikan nilai yang telah dihitung sebelumnya. Amazon S3 kemudian memvalidasi checksum ini sebelum menyimpan objek dan nilai checksum-nya. Anda dapat memverifikasi integritas data objek saat meminta nilai checksum selama pengunduhan objek.

Unggahan multipart

Saat mengunggah objek di beberapa bagian menggunakan `MultipartUpload`API, Anda dapat menentukan algoritma checksum yang ingin digunakan Amazon S3 dan jenis checksum (objek lengkap atau komposit).

Tabel berikut menunjukkan jenis algoritma checksum yang didukung untuk setiap algoritma checksum dalam unggahan multipart:

Algoritma checksum	Objek penuh	Komposit
CRC-64/NVME () `CRC64NVME`	Ya	Tidak
CRC-32 () `CRC32`	Ya	Ya
CRC-32C () `CRC32C`	Ya	Ya
SHA-1 () `SHA1`	Tidak	Ya
SHA-256 () `SHA256`	Tidak	Ya

Menggunakan checksum objek lengkap untuk unggahan multipart

Saat membuat atau melakukan unggahan multibagian, Anda dapat menggunakan checksum objek lengkap untuk validasi saat diunggah. Ini berarti Anda dapat menyediakan algoritme checksum untuk `MultipartUpload`API, menyederhanakan alat validasi integritas karena Anda tidak perlu lagi melacak batas bagian untuk objek yang diunggah. Anda dapat memberikan checksum seluruh objek dalam `CompleteMultipartUpload`permintaan, bersama dengan ukuran objek.

Saat Anda memberikan checksum objek lengkap selama pengunggahan multibagian, AWS SDK meneruskan checksum ke Amazon S3, dan S3 memvalidasi sisi server integritas objek, membandingkannya dengan nilai yang diterima. Kemudian, Amazon S3 menyimpan objek jika nilainya cocok. Jika kedua nilai tidak cocok, S3 gagal permintaan dengan `BadDigest` kesalahan. Checksum objek Anda juga disimpan dalam metadata objek yang Anda gunakan nanti untuk memvalidasi integritas data objek.

Untuk checksum objek lengkap, Anda dapat menggunakan algoritma checksum CRC-64/NVME (`CRC64NVME`), CRC-32 (), atau CRC-32C (`CRC32`) di S3. `CRC32C` Checksum objek lengkap dalam unggahan multibagian hanya tersedia untuk checksum berbasis CRC karena mereka dapat linier menjadi checksum objek lengkap. Linearisasi ini memungkinkan Amazon S3 untuk memparalelkan permintaan Anda untuk meningkatkan kinerja. Secara khusus, S3 dapat menghitung checksum seluruh objek dari checksum tingkat bagian. Jenis validasi ini tidak tersedia untuk algoritma lain, seperti SHA dan. MD5 Karena S3 memiliki perlindungan integritas default, jika objek diunggah tanpa checksum, S3 secara otomatis melampirkan algoritma checksum objek penuh CRC-64/NVME () yang direkomendasikan ke objek. `CRC64NVME`

> **Note**
>
> Untuk memulai unggahan multipart, Anda dapat menentukan algoritma checksum dan jenis checksum objek lengkap. Setelah Anda menentukan algoritma checksum dan jenis checksum objek lengkap, Anda dapat memberikan nilai checksum objek lengkap untuk unggahan multipart.

Menggunakan checksum tingkat suku cadang untuk unggahan multipart

Ketika objek diunggah ke Amazon S3, mereka dapat diunggah sebagai objek tunggal atau diunggah dalam beberapa bagian dengan proses pengunggahan multibagian. Anda dapat memilih jenis Checksum untuk unggahan multipart Anda. Untuk checksum tingkat suku cadang unggahan multibagian (atau checksum komposit), Amazon S3 menghitung checksum untuk setiap bagian individual dengan menggunakan algoritma checksum yang ditentukan. Anda dapat menggunakan `UploadPart`untuk memberikan nilai checksum untuk setiap bagian. Jika objek yang Anda coba unggah di konsol Amazon S3 diatur untuk menggunakan algoritma checksum CRC-64/NVME (`CRC64NVME`) dan melebihi 16 MB, objek tersebut secara otomatis ditetapkan sebagai checksum objek lengkap.

Amazon S3 kemudian menggunakan nilai checksum tingkat bagian yang disimpan untuk mengonfirmasi bahwa setiap bagian diunggah dengan benar. Ketika checksum setiap bagian (untuk seluruh objek) disediakan, S3 menggunakan nilai checksum yang disimpan dari setiap bagian untuk menghitung checksum objek lengkap secara internal, membandingkannya dengan nilai checksum yang disediakan. Ini meminimalkan biaya komputasi karena S3 dapat menghitung checksum dari seluruh objek menggunakan checksum bagian-bagiannya. Untuk informasi selengkapnya tentang unggahan multipart, lihat Mengunggah dan menyalin objek menggunakan unggahan multipart di Amazon S3 dan. Menggunakan checksum objek lengkap untuk unggahan multipart

Ketika objek benar-benar diunggah, Anda dapat menggunakan checksum terhitung akhir untuk memverifikasi integritas data objek.

Saat mengunggah bagian dari unggahan multipart, perhatikan hal-hal berikut:

- Untuk mengambil informasi tentang objek, termasuk berapa banyak bagian yang membentuk seluruh objek, Anda dapat menggunakan `GetObjectAttributes`operasi. Dengan checksum tambahan, Anda juga dapat memulihkan informasi untuk setiap bagian yang mencakup nilai checksum bagian.
- Untuk upload selesai, Anda bisa mendapatkan checksum bagian individual dengan menggunakan `HeadObject`operasi `GetObject`atau dan menentukan nomor bagian atau rentang byte yang sejajar dengan satu bagian. Jika Anda ingin mengambil nilai checksum untuk masing-masing bagian dari unggahan multibagian yang masih berlangsung, Anda dapat menggunakannya. `ListParts`
- Oleh karena cara Amazon S3 menghitung checksum untuk objek multibagian, nilai checksum untuk objek mungkin berubah jika Anda menyalinnya. Jika Anda menggunakan SDK atau REST API dan menelepon `CopyObject`, Amazon S3 menyalin objek apa pun hingga batasan ukuran `CopyObject` operasi API. Amazon S3 melakukan penyalinan ini sebagai tindakan tunggal, terlepas dari apakah objek diunggah dalam satu permintaan atau sebagai bagian dari unggahan multibagian. Dengan perintah salin, checksum objek adalah checksum langsung dari objek penuh. Jika objek awalnya diunggah menggunakan unggahan multibagian, nilai checksum berubah meskipun datanya tidak.
- Objek yang lebih besar dari batasan ukuran operasi `CopyObject` API harus menggunakan perintah salinan unggah multibagian.
- Saat Anda melakukan beberapa operasi menggunakan AWS Management Console, Amazon S3 menggunakan unggahan multibagian jika objek berukuran lebih besar dari 16 MB.

Operasi Checksum

Setelah mengunggah objek, Anda bisa mendapatkan nilai checksum dan membandingkannya dengan nilai checksum yang telah dihitung sebelumnya atau disimpan sebelumnya dari jenis algoritma yang sama. Contoh berikut menunjukkan operasi atau metode checksum mana yang dapat Anda gunakan untuk memverifikasi integritas data.

Menggunakan konsol S3

Untuk mempelajari lebih lanjut tentang menggunakan konsol dan menentukan algoritma checksum yang akan digunakan saat mengunggah objek, lihat Mengunggah Objek dan Tutorial: Memeriksa integritas data di Amazon S3 dengan checksum tambahan.

Menggunakan AWS SDKs

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan AWS SDKs untuk meng-upload file besar dengan upload multipart, men-download file besar, dan memvalidasi file upload multipart, semua dengan menggunakan SHA-256 untuk validasi file.

Java

Example Contoh: Mengunggah, mengunduh, dan memverifikasi file yang berukuran besar dengan SHA-256

Untuk petunjuk cara membuat dan menguji sampel yang berfungsi, lihat Memulai di Panduan AWS SDK untuk Java Pengembang.

```
import software.amazon.awssdk.auth.credentials.AwsCredentials;
import software.amazon.awssdk.auth.credentials.AwsCredentialsProvider;
import software.amazon.awssdk.core.ResponseInputStream;
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.AbortMultipartUploadRequest;
import software.amazon.awssdk.services.s3.model.ChecksumAlgorithm;
import software.amazon.awssdk.services.s3.model.ChecksumMode;
import software.amazon.awssdk.services.s3.model.CompleteMultipartUploadRequest;
import software.amazon.awssdk.services.s3.model.CompleteMultipartUploadResponse;
import software.amazon.awssdk.services.s3.model.CompletedMultipartUpload;
import software.amazon.awssdk.services.s3.model.CompletedPart;
import software.amazon.awssdk.services.s3.model.CreateMultipartUploadRequest;
import software.amazon.awssdk.services.s3.model.CreateMultipartUploadResponse;
```

```
    import software.amazon.awssdk.services.s3.model.GetObjectAttributesRequest;
    import software.amazon.awssdk.services.s3.model.GetObjectAttributesResponse;
    import software.amazon.awssdk.services.s3.model.GetObjectRequest;
    import software.amazon.awssdk.services.s3.model.GetObjectResponse;
    import software.amazon.awssdk.services.s3.model.GetObjectTaggingRequest;
    import software.amazon.awssdk.services.s3.model.ObjectAttributes;
    import software.amazon.awssdk.services.s3.model.PutObjectTaggingRequest;
    import software.amazon.awssdk.services.s3.model.Tag;
    import software.amazon.awssdk.services.s3.model.Tagging;
    import software.amazon.awssdk.services.s3.model.UploadPartRequest;
    import software.amazon.awssdk.services.s3.model.UploadPartResponse;

    import java.io.File;
    import java.io.FileInputStream;
    import java.io.FileOutputStream;
    import java.io.IOException;
    import java.io.InputStream;
    import java.io.OutputStream;
    import java.nio.ByteBuffer;
    import java.security.MessageDigest;
    import java.security.NoSuchAlgorithmException;
    import java.util.ArrayList;
    import java.util.Base64;
    import java.util.List;

    public class LargeObjectValidation {
        private static String FILE_NAME = "sample.file";
        private static String BUCKET = "sample-bucket";
        //Optional, if you want a method of storing the full multipart object
 checksum in S3.
        private static String CHECKSUM_TAG_KEYNAME = "fullObjectChecksum";
        //If you have existing full-object checksums that you need to validate
 against, you can do the full object validation on a sequential upload.
        private static String SHA256_FILE_BYTES = "htCM5g7ZNdoSw8bN/
mkgiAhXt5MFoVowVg+LE9aIQmI=";
        //Example Chunk Size - this must be greater than or equal to 5MB.
        private static int CHUNK_SIZE = 5 * 1024 * 1024;

        public static void main(String[] args) {
            S3Client s3Client = S3Client.builder()
                    .region(Region.US_EAST_1)
                    .credentialsProvider(new AwsCredentialsProvider() {
                        @Override
                        public AwsCredentials resolveCredentials() {
```

```
                            return new AwsCredentials() {
                                @Override
                                public String accessKeyId() {
                                    return Constants.ACCESS_KEY;
                                }

                                @Override
                                public String secretAccessKey() {
                                    return Constants.SECRET;
                                }
                            };
                        }
                    })
                    .build();
            uploadLargeFileBracketedByChecksum(s3Client);
            downloadLargeFileBracketedByChecksum(s3Client);
            validateExistingFileAgainstS3Checksum(s3Client);
        }

        public static void uploadLargeFileBracketedByChecksum(S3Client s3Client) {
            System.out.println("Starting uploading file validation");
            File file = new File(FILE_NAME);
            try (InputStream in = new FileInputStream(file)) {
                MessageDigest sha256 = MessageDigest.getInstance("SHA-256");
                CreateMultipartUploadRequest createMultipartUploadRequest =
CreateMultipartUploadRequest.builder()
                        .bucket(BUCKET)
                        .key(FILE_NAME)
                        .checksumAlgorithm(ChecksumAlgorithm.SHA256)
                        .build();
                CreateMultipartUploadResponse createdUpload =
s3Client.createMultipartUpload(createMultipartUploadRequest);
                List<CompletedPart> completedParts = new ArrayList<CompletedPart>();
                int partNumber = 1;
                byte[] buffer = new byte[CHUNK_SIZE];
                int read = in.read(buffer);
                while (read != -1) {
                    UploadPartRequest uploadPartRequest =
UploadPartRequest.builder()

.partNumber(partNumber).uploadId(createdUpload.uploadId()).key(FILE_NAME).bucket(BUCKET).ch
                    UploadPartResponse uploadedPart =
s3Client.uploadPart(uploadPartRequest,
RequestBody.fromByteBuffer(ByteBuffer.wrap(buffer, 0, read)));
```

```
                    CompletedPart part =
CompletedPart.builder().partNumber(partNumber).checksumSHA256(uploadedPart.checksumSHA256()
                    completedParts.add(part);
                    sha256.update(buffer, 0, read);
                    read = in.read(buffer);
                    partNumber++;
                }
                String fullObjectChecksum =
Base64.getEncoder().encodeToString(sha256.digest());
                if (!fullObjectChecksum.equals(SHA256_FILE_BYTES)) {
                    //Because the SHA256 is uploaded after the part is uploaded; the
upload is bracketed and the full object can be fully validated.

s3Client.abortMultipartUpload(AbortMultipartUploadRequest.builder().bucket(BUCKET).key(FILE
                    throw new IOException("Byte mismatch between stored checksum and
upload, do not proceed with upload and cleanup");
                }
                CompletedMultipartUpload completedMultipartUpload =
CompletedMultipartUpload.builder().parts(completedParts).build();
                CompleteMultipartUploadResponse completedUploadResponse =
s3Client.completeMultipartUpload(

CompleteMultipartUploadRequest.builder().bucket(BUCKET).key(FILE_NAME).uploadId(createdUplo
                Tag checksumTag =
Tag.builder().key(CHECKSUM_TAG_KEYNAME).value(fullObjectChecksum).build();
                //Optionally, if you need the full object checksum stored with the
file; you could add it as a tag after completion.

s3Client.putObjectTagging(PutObjectTaggingRequest.builder().bucket(BUCKET).key(FILE_NAME).t
            } catch (IOException | NoSuchAlgorithmException e) {
                e.printStackTrace();
            }
            GetObjectAttributesResponse
                    objectAttributes =
s3Client.getObjectAttributes(GetObjectAttributesRequest.builder().bucket(BUCKET).key(FILE_N
                    .objectAttributes(ObjectAttributes.OBJECT_PARTS,
ObjectAttributes.CHECKSUM).build());
            System.out.println(objectAttributes.objectParts().parts());
            System.out.println(objectAttributes.checksum().checksumSHA256());
        }

        public static void downloadLargeFileBracketedByChecksum(S3Client s3Client) {
            System.out.println("Starting downloading file validation");
            File file = new File("DOWNLOADED_" + FILE_NAME);
```

```
            try (OutputStream out = new FileOutputStream(file)) {
                GetObjectAttributesResponse
                        objectAttributes =
s3Client.getObjectAttributes(GetObjectAttributesRequest.builder().bucket(BUCKET).key(FILE_N
                        .objectAttributes(ObjectAttributes.OBJECT_PARTS,
ObjectAttributes.CHECKSUM).build());
                //Optionally if you need the full object checksum, you can grab a
tag you added on the upload
                List<Tag> objectTags =
s3Client.getObjectTagging(GetObjectTaggingRequest.builder().bucket(BUCKET).key(FILE_NAME).b
                String fullObjectChecksum = null;
                for (Tag objectTag : objectTags) {
                    if (objectTag.key().equals(CHECKSUM_TAG_KEYNAME)) {
                        fullObjectChecksum = objectTag.value();
                        break;
                    }
                }
                MessageDigest sha256FullObject =
MessageDigest.getInstance("SHA-256");
                MessageDigest sha256ChecksumOfChecksums =
MessageDigest.getInstance("SHA-256");

                //If you retrieve the object in parts, and set the ChecksumMode to
enabled, the SDK will automatically validate the part checksum
                for (int partNumber = 1; partNumber <=
objectAttributes.objectParts().totalPartsCount(); partNumber++) {
                    MessageDigest sha256Part = MessageDigest.getInstance("SHA-256");
                    ResponseInputStream<GetObjectResponse> response =
s3Client.getObject(GetObjectRequest.builder().bucket(BUCKET).key(FILE_NAME).partNumber(part
                    GetObjectResponse getObjectResponse = response.response();
                    byte[] buffer = new byte[CHUNK_SIZE];
                    int read = response.read(buffer);
                    while (read != -1) {
                        out.write(buffer, 0, read);
                        sha256FullObject.update(buffer, 0, read);
                        sha256Part.update(buffer, 0, read);
                        read = response.read(buffer);
                    }
                    byte[] sha256PartBytes = sha256Part.digest();
                    sha256ChecksumOfChecksums.update(sha256PartBytes);
                    //Optionally, you can do an additional manual validation again
the part checksum if needed in addition to the SDK check
                    String base64PartChecksum =
Base64.getEncoder().encodeToString(sha256PartBytes);
```

```
                    String base64PartChecksumFromObjectAttributes =
 objectAttributes.objectParts().parts().get(partNumber - 1).checksumSHA256();
                    if (!
base64PartChecksum.equals(getObjectResponse.checksumSHA256()) || !
base64PartChecksum.equals(base64PartChecksumFromObjectAttributes)) {
                        throw new IOException("Part checksum didn't match for the
 part");
                    }
                    System.out.println(partNumber + " " + base64PartChecksum);
                }
                //Before finalizing, do the final checksum validation.
                String base64FullObject =
 Base64.getEncoder().encodeToString(sha256FullObject.digest());
                String base64ChecksumOfChecksums =
 Base64.getEncoder().encodeToString(sha256ChecksumOfChecksums.digest());
                if (fullObjectChecksum != null && !
fullObjectChecksum.equals(base64FullObject)) {
                    throw new IOException("Failed checksum validation for full
 object");
                }
                System.out.println(fullObjectChecksum);
                String base64ChecksumOfChecksumFromAttributes =
 objectAttributes.checksum().checksumSHA256();
                if (base64ChecksumOfChecksumFromAttributes != null && !
base64ChecksumOfChecksums.equals(base64ChecksumOfChecksumFromAttributes)) {
                    throw new IOException("Failed checksum validation for full
 object checksum of checksums");
                }
                System.out.println(base64ChecksumOfChecksumFromAttributes);
                out.flush();
            } catch (IOException | NoSuchAlgorithmException e) {
                //Cleanup bad file
                file.delete();
                e.printStackTrace();
            }
        }

        public static void validateExistingFileAgainstS3Checksum(S3Client s3Client)
 {
            System.out.println("Starting existing file validation");
            File file = new File("DOWNLOADED_" + FILE_NAME);
            GetObjectAttributesResponse
                    objectAttributes =
 s3Client.getObjectAttributes(GetObjectAttributesRequest.builder().bucket(BUCKET).key(FILE_N
```

```
                    .objectAttributes(ObjectAttributes.OBJECT_PARTS,
 ObjectAttributes.CHECKSUM).build());
            try (InputStream in = new FileInputStream(file)) {
                MessageDigest sha256ChecksumOfChecksums =
 MessageDigest.getInstance("SHA-256");
                MessageDigest sha256Part = MessageDigest.getInstance("SHA-256");
                byte[] buffer = new byte[CHUNK_SIZE];
                int currentPart = 0;
                int partBreak =
 objectAttributes.objectParts().parts().get(currentPart).size();
                int totalRead = 0;
                int read = in.read(buffer);
                while (read != -1) {
                    totalRead += read;
                    if (totalRead >= partBreak) {
                        int difference = totalRead - partBreak;
                        byte[] partChecksum;
                        if (totalRead != partBreak) {
                            sha256Part.update(buffer, 0, read - difference);
                            partChecksum = sha256Part.digest();
                            sha256ChecksumOfChecksums.update(partChecksum);
                            sha256Part.reset();
                            sha256Part.update(buffer, read - difference,
 difference);
                        } else {
                            sha256Part.update(buffer, 0, read);
                            partChecksum = sha256Part.digest();
                            sha256ChecksumOfChecksums.update(partChecksum);
                            sha256Part.reset();
                        }
                        String base64PartChecksum =
 Base64.getEncoder().encodeToString(partChecksum);
                        if (!
base64PartChecksum.equals(objectAttributes.objectParts().parts().get(currentPart).checksumSH
 {
                            throw new IOException("Part checksum didn't match S3");
                        }
                        currentPart++;
                        System.out.println(currentPart + " " + base64PartChecksum);
                        if (currentPart <
 objectAttributes.objectParts().totalPartsCount()) {
                            partBreak +=
 objectAttributes.objectParts().parts().get(currentPart - 1).size();
                        }
```

```
                    } else {
                        sha256Part.update(buffer, 0, read);
                    }
                    read = in.read(buffer);
                }
                if (currentPart != objectAttributes.objectParts().totalPartsCount())
 {
                    currentPart++;
                    byte[] partChecksum = sha256Part.digest();
                    sha256ChecksumOfChecksums.update(partChecksum);
                    String base64PartChecksum =
 Base64.getEncoder().encodeToString(partChecksum);
                    System.out.println(currentPart + " " + base64PartChecksum);
                }

                String base64CalculatedChecksumOfChecksums =
 Base64.getEncoder().encodeToString(sha256ChecksumOfChecksums.digest());
                System.out.println(base64CalculatedChecksumOfChecksums);
                System.out.println(objectAttributes.checksum().checksumSHA256());
                if (!
base64CalculatedChecksumOfChecksums.equals(objectAttributes.checksum().checksumSHA256()))
 {
                    throw new IOException("Full object checksum of checksums don't
 match S3");
                }

            } catch (IOException | NoSuchAlgorithmException e) {
                e.printStackTrace();
            }
        }
    }
```

Penggunaan API REST

Anda dapat mengirim permintaan REST untuk mengunggah objek dengan nilai checksum untuk memverifikasi integritas data dengan PutObject. Anda juga dapat mengambil nilai checksum untuk objek yang menggunakan GetObjectatau. HeadObject

Menggunakan AWS CLI

Anda dapat mengirim permintaan PUT untuk mengunggah objek hingga 5 GB dalam satu operasi. Untuk informasi selengkapnya, lihat `PutObject` di Referensi Perintah AWS CLI . Anda juga dapat

menggunakan `get-object` dan `head-object` untuk mengambil checksum dari objek yang sudah diunggah guna memverifikasi integritas data.

Untuk selengkapnya, lihat FAQ Amazon S3 CLI di Panduan Pengguna.AWS Command Line Interface

Menggunakan Konten- MD5 saat mengunggah objek

Cara lain untuk memverifikasi integritas objek Anda setelah mengunggah adalah dengan memberikan MD5 intisari objek saat Anda mengunggahnya. Jika Anda menghitung MD5 intisari untuk objek Anda, Anda dapat memberikan intisari dengan PUT perintah dengan menggunakan header. `Content-MD5`

Setelah mengunggah objek, Amazon S3 menghitung MD5 intisari objek dan membandingkannya dengan nilai yang Anda berikan. Permintaan hanya berhasil jika kedua intisari cocok.

Menyediakan MD5 intisari tidak diperlukan, tetapi Anda dapat menggunakannya untuk memverifikasi integritas objek sebagai bagian dari proses pengunggahan.

Menggunakan Konten- MD5 dan ETag untuk memverifikasi objek yang diunggah

Tag entitas (ETag) untuk objek mewakili versi tertentu dari objek tersebut. Perlu diingat bahwa ETag satu-satunya mencerminkan perubahan pada konten objek, bukan perubahan metadatanya. Jika hanya metadata suatu objek yang berubah, ETag tetap sama.

Tergantung pada objek, objek mungkin merupakan MD5 intisari dari data objek: ETag

- Jika objek dibuat oleh,, atau `CopyObject` operasi `PutObject` `PostObject`, atau melalui AWS Management Console, dan objek itu juga teks biasa atau dienkripsi oleh enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3), objek tersebut memiliki intisari data objeknya. ETag MD5
- Jika objek dibuat oleh,, atau `CopyObject` operasi `PutObject` `PostObject`, atau melalui AWS Management Console, dan objek itu dienkripsi oleh enkripsi sisi server dengan kunci yang disediakan pelanggan (SSE-C) atau enkripsi sisi server dengan () kunci (SSE-KMS AWS KMS), objek tersebut memiliki intisari data objeknya. AWS Key Management Service ETag MD5
- Jika objek dibuat oleh proses upload multipart atau `UploadPartCopy` operasi, objek ETag bukanlah MD5 intisari, terlepas dari metode enkripsi. Jika objek lebih besar dari 16 MB, AWS Management Console upload atau copy objek tersebut sebagai upload multipart, dan oleh karena itu ETag bukan MD5 intisari.

Untuk objek yang ETag merupakan `Content-MD5` intisari objek, Anda dapat membandingkan ETag nilai objek dengan `Content-MD5` intisari yang dihitung atau disimpan sebelumnya.

Menggunakan checksum trailing

Saat mengunggah objek ke Amazon S3, Anda dapat memberikan checksum yang telah dihitung sebelumnya untuk objek tersebut atau menggunakan SDK untuk secara otomatis membuat checksum AWS tambahan untuk unggahan yang dipotong, atas nama Anda. Jika Anda menggunakan checksum tambahan, Amazon S3 secara otomatis menghasilkan checksum dengan menggunakan algoritme yang ditentukan untuk memvalidasi integritas objek dalam unggahan terpotong, saat Anda mengunggah objek.

Untuk membuat checksum tambahan saat menggunakan AWS SDK, isi `ChecksumAlgorithm` parameter dengan algoritme pilihan Anda. SDK menggunakan algoritme tersebut untuk menghitung checksum untuk objek Anda (atau bagian objek) dan secara otomatis menambahkannya ke akhir permintaan unggahan yang terpotong. Perilaku ini menghemat waktu Anda, sebab Amazon S3 melakukan verifikasi dan pengunggahan data Anda dalam sekali jalan.

> **Important**
>
> Jika Anda menggunakan S3 Lambda Objek, semua permintaan ke S3 Lambda Objek ditandatangani menggunakan, `s3-object-lambda` dan bukan `s3`. Perilaku ini memengaruhi tanda tangan nilai checksum trailing. Untuk informasi selengkapnya tentang S3 Lambda Objek, lihat Mengubah objek dengan S3 Lambda Objek.

Header checksum tertinggal

Untuk membuat permintaan pengkodean konten yang terpotong, Amazon S3 mengharuskan klien menyertakan beberapa header untuk mengurai permintaan dengan benar. Klien harus menyertakan header berikut:

- **`x-amz-decoded-content-length`**: Header ini menunjukkan ukuran plaintext dari data aktual yang sedang diunggah ke Amazon S3 dengan permintaan.
- **`x-amz-content-sha256`**: Header ini menunjukkan jenis unggahan chunked yang disertakan dalam permintaan. Untuk unggahan terpotong dengan checksum tambahan, nilai header adalah `STREAMING-UNSIGNED-PAYLOAD-TRAILER` untuk permintaan yang tidak menggunakan penandatanganan payload dan `STREAMING-AWS4-HMAC-SHA256-PAYLOAD-TRAILER` untuk

permintaan yang menggunakan penandatanganan payload SigV4. (Untuk informasi selengkapnya tentang penerapan payload yang ditandatangani, lihat Perhitungan tanda tangan untuk header otorisasi: Mentransfer muatan dalam beberapa potongan.)

- **`x-amz-trailer`**: Header ini menunjukkan nama header trailing dalam permintaan. Jika checksum tambahan ada (di mana AWS SDKs menambahkan checksum ke badan permintaan yang dikodekan), nilai `x-amz-trailer` header menyertakan awalan dan diakhiri dengan nama algoritme. `x-amz-checksum-` `x-amz-trailer`Nilai-nilai berikut saat ini didukung:
  - `x-amz-checksum-crc32`
  - `x-amz-checksum-crc32c`
  - `x-amz-checksum-crc64nvme`
  - `x-amz-checksum-sha1`
  - `x-amz-checksum-sha256`

> **Note**
>
> Anda juga dapat menyertakan `Content-Encoding` header, dengan nilai chunked, dalam permintaan Anda. Meskipun header ini tidak diperlukan, termasuk header ini dapat meminimalkan masalah proxy HTTP saat mentransmisikan data yang dikodekan. Jika `Content-Encoding` header lain (seperti gzip) ada dalam permintaan, `Content-Encoding` header menyertakan nilai chunked dalam daftar pengkodean yang dipisahkan koma. Misalnya, `Content-Encoding: aws-chunked, gzip`.

Bagian yang terpotong

Saat Anda mengunggah objek ke Amazon S3 menggunakan encoding chunked, permintaan upload menyertakan jenis potongan berikut (diformat dalam urutan yang tercantum):

- Potongan badan objek: Mungkin ada satu, beberapa, atau nol potongan tubuh yang terkait dengan permintaan unggahan terpotong.
- Potongan penyelesaian: Mungkin ada satu, beberapa, atau nol potongan tubuh yang terkait dengan permintaan unggahan terpotong.
- Potongan trailing: Checksum trailing terdaftar setelah potongan penyelesaian. Hanya satu potongan trailing yang diizinkan.

> **Note**
>
> Setiap unggahan yang terpotong harus diakhiri dengan CRLF akhir (seperti`\r\n`) untuk menunjukkan akhir permintaan.

Untuk contoh pemformatan chunked, lihat. Contoh: Unggahan terpotong dengan checksum tambahan

Potongan tubuh objek

Potongan tubuh objek adalah potongan yang berisi data objek aktual yang sedang diunggah ke S3. Potongan ini memiliki batasan ukuran dan format yang konsisten.

Ukuran potongan tubuh objek

Potongan ini harus berisi setidaknya 8.192 byte (atau 8 KiB) data objek, kecuali untuk potongan tubuh akhir, yang bisa lebih kecil. Tidak ada ukuran potongan maksimum eksplisit tetapi Anda dapat mengharapkan semua potongan lebih kecil dari ukuran unggah maksimum 5 GB. Ukuran potongan dapat bervariasi dari satu potongan ke potongan berikutnya berdasarkan implementasi server klien Anda.

Format potongan tubuh objek

Potongan tubuh objek dimulai dengan pengkodean heksadesimal dari jumlah byte dalam potongan tubuh objek, diikuti oleh CRLF (Carriage Return Line Feed), byte objek untuk potongan itu, dan CRLF lainnya.

Misalnya:

```
hex-encoding-of-object-bytes-in-chunk\r\n
chunk-object-bytes\r\n
```

Namun, ketika potongan ditandatangani, potongan tubuh objek mengikuti format yang berbeda, di mana tanda tangan ditambahkan ke ukuran potongan dengan pembatas titik koma. Misalnya:

```
hex-encoding-of-object-bytes-in-chunk;chunk-signature\r\n
 chunk-object-bytes\r\n
```

Untuk informasi selengkapnya tentang penandatanganan potongan, lihat Perhitungan tanda tangan untuk Authorization Header: Mentransfer muatan dalam beberapa potongan (AWSVersi Tanda

Tangan 4). Untuk informasi selengkapnya tentang pemformatan potongan, lihat Pengkodean transfer terpotong di situs web Editor RFC.

Potongan penyelesaian

Potongan penyelesaian diperlukan untuk menjadi potongan tubuh objek terakhir dari setiap unggahan yang dipotong. Format potongan penyelesaian mirip dengan potongan tubuh, tetapi selalu berisi nol byte data objek. (Byte nol dari data objek menunjukkan bahwa semua data telah diunggah.) Unggahan terpotong harus menyertakan potongan penyelesaian sebagai potongan tubuh objek terakhirnya, mengikuti format seperti ini:

```
0\r\n
```

Namun, jika permintaan pengkodean konten menggunakan penandatanganan payload, itu mengikuti format ini sebagai gantinya:

```
0;chunk-signature\r\n
```

Potongan trailer

Potongan trailer menyimpan checksum yang dihitung untuk semua permintaan unggahan S3. Potongan trailer mencakup dua bidang: satu bidang nama header dan satu bidang nilai header. Bidang nama header untuk permintaan upload harus sesuai dengan nilai yang diteruskan ke header `x-amz-trailer` permintaan. Misalnya, jika permintaan berisi `x-amz-trailer: x-amz-checksum-crc32` dan potongan trailer memiliki nama header`x-amz-checksum-sha1`, permintaan gagal. Bidang nilai dalam potongan trailer mencakup pengkodean base64 dari nilai checksum endian besar untuk objek tersebut. (Pengurutan big-endian menyimpan byte data paling signifikan pada alamat memori terendah, dan byte paling tidak signifikan pada alamat memori terbesar). Algoritma yang digunakan untuk menghitung checksum ini sama dengan akhiran untuk nama header (misalnya,`crc32`).

Format potongan trailer

Potongan trailer menggunakan format berikut untuk permintaan muatan yang tidak ditandatangani:

```
x-amz-checksum-lowercase-checksum-algorithm-name:base64-checksum-value\n\r\n\r\n
```

Untuk permintaan dengan muatan yang ditandatangani SiGv4, potongan trailer menyertakan tanda tangan trailer setelah potongan trailer.

```
trailer-checksum\n\r\n
trailer-signature\r\n
```

Anda juga dapat menambahkan CRLF langsung ke akhir nilai checksum base64. Misalnya:

```
x-amz-checksum-lowercase-checksum-algorithm-name:base64-checksum-value\r\n\r\n
```

Contoh: Unggahan terpotong dengan checksum tambahan

Amazon S3 mendukung unggahan terpotong yang menggunakan pengkodean `aws-chunked` konten untuk `PutObject` dan permintaan dengan checksum tambahan. `UploadPart`

Example 1 — **`PutObject`** Permintaan potongan yang tidak ditandatangani dengan checksum CRC-32 tertinggal

Berikut ini adalah contoh `PutObject` permintaan chunked dengan checksum CRC-32 tertinggal. Dalam contoh ini, klien mengunggah objek 17 KB dalam tiga potongan yang tidak ditandatangani dan menambahkan potongan checksum CRC-32 tertinggal dengan menggunakan header. `x-amz-checksum-crc32`

```
PUT /Key+ HTTP/1.1
Host: amzn-s3-demo-bucket
Content-Encoding: aws-chunked
x-amz-decoded-content-length: 17408
x-amz-content-sha256: STREAMING-UNSIGNED-PAYLOAD-TRAILER
x-amz-trailer: x-amz-checksum-crc32

2000\r\n                                    // Object body chunk 1 (8192 bytes)
object-bytes\r\n
2000\r\n                                    // Object body chunk 2 (8192 bytes)
object-bytes\r\n
400\r\n                                     // Object body chunk 3 (1024 bytes)
object-bytes\r\n
0\r\n                                       // Completion chunk
x-amz-checksum-crc32:YABb/g==\n\r\n\r\n     // Trailer chunk (note optional \n
 character)
\r\n                                          // CRLF
```

Berikut adalah contoh respon:

```
HTTP/1.1 200
```

```
ETag: ETag
x-amz-checksum-crc32: YABb/g==
```

> **Note**
>
> Penggunaan linefeed \n di akhir nilai checksum mungkin berbeda antar klien.

Example 2 — **`PutObject`** Permintaan chunked yang ditandatangani SIGV4 dengan checksum CRC-32 () tertinggal **`CRC32`**

Berikut ini adalah contoh `PutObject` permintaan chunked dengan checksum CRC-32 tertinggal. Permintaan ini menggunakan penandatanganan payload SiGv4. Dalam contoh ini, klien mengunggah objek 17 KB dalam tiga potongan yang ditandatangani. Selain `object body` potongan, `completion chunk` dan `trailer chunk` juga ditandatangani.

```
PUT /Key+ HTTP/1.1
Host: amzn-s3-demo-bucket.s3.amazonaws.com
Content-Encoding: aws-chunked
x-amz-decoded-content-length: 17408
x-amz-content-sha256: STREAMING-AWS4-HMAC-SHA256-PAYLOAD-TRAILER
x-amz-trailer: x-amz-checksum-crc32

authorization-code                              // SigV4 headers authorization

2000;chunk-signature=signature-value...\r\n   // Object body chunk 1 (8192 bytes)
object-bytes\r\n
2000;chunk-signature\r\n                        // Object body chunk 2 (8192 bytes)
object-bytes\r\n
400;chunk-signature\r\n                         // Object body chunk 3 (1024 bytes)
object-bytes\r\n
0;chunk-signature\r\n                           // Completion chunk
x-amz-checksum-crc32:YABb/g==\n\r\n             // Trailer chunk (note optional \n
 character)
trailer-signature\r\n
\r\n                                            // CRLF
```

Berikut adalah contoh respon:

```
HTTP/1.1 200
ETag: ETag
```

```
x-amz-checksum-crc32: YABb/g==
```

Menghapus objek Amazon S3

Anda dapat menghapus satu atau beberapa objek langsung dari Amazon S3 menggunakan konsol Amazon S3,,AWS CLI() AWS SDKs AWS Command Line Interface , atau REST API. Misalnya, jika Anda mengumpulkan file log, sebaiknya hapus file tersebut saat tidak diperlukan lagi. Anda dapat mengatur aturan Siklus Hidup S3 untuk secara otomatis menghapus objek seperti file log.

Untuk menghapus objek, Anda dapat menggunakan salah satu operasi API berikut:

- Delete a single object–Amazon S3 menyediakan operasi API `DELETE` (`DeleteObject`) yang dapat Anda gunakan untuk menghapus satu objek dalam satu permintaan HTTP.
- Hapus beberapa objek–Amazon S3 menyediakan operasi API Multi-Object Delete (`DeleteObjects`) yang dapat Anda gunakan untuk menghapus hingga 1.000 objek dalam satu permintaan HTTP.

Saat menghapus objek dari bucket yang tidak diaktifkan versi, Anda hanya memberikan nama kunci objek. Namun, saat menghapus objek dari bucket berkemampuan versi, Anda dapat memberikan ID versi objek untuk menghapus versi objek tertentu.

Praktik terbaik untuk dipertimbangkan sebelum menghapus objek

Sebelum Anda menghapus objek, pertimbangkan praktik terbaik berikut:

- Aktifkan pembuatan versi bucket. S3 Versioning menambahkan perlindungan terhadap `DeleteObject` permintaan sederhana untuk mencegah penghapusan yang tidak disengaja. Untuk bucket berversi, jika Anda menghapus versi objek saat ini atau ketika permintaan penghapusan tidak menentukan Id versi tertentu, Amazon S3 tidak akan menghapus objek secara permanen. Sebagai gantinya, S3 menambahkan penanda hapus, mengeluarkan soft delete objek. Penanda hapus kemudian menjadi versi objek saat ini (atau terbaru) dengan ID versi baru. Untuk informasi selengkapnya, lihat Menghapus versi objek dari bucket berkemampuan versi.
- Jika Anda ingin menghapus sejumlah besar objek, atau untuk menghapus objek secara terprogram berdasarkan tanggal pembuatan objek, tetapkan konfigurasi Siklus Hidup S3 di bucket Anda. Untuk memantau penghapusan ini, kami sarankan Anda menggunakan pemberitahuan peristiwa Siklus Hidup S3. Saat Anda mengonfigurasi notifikasi Siklus Hidup S3, jenis `s3:LifecycleExpiration:Delete` acara akan memberi tahu Anda saat objek dalam

bucket dihapus. Ini juga memberi tahu Anda ketika versi objek dihapus secara permanen oleh konfigurasi Siklus Hidup S3. Jenis `s3:LifecycleExpiration:DeleteMarkerCreated` acara memberi tahu Anda saat Siklus Hidup S3 membuat penanda hapus. Penanda hapus dibuat saat versi objek saat ini dalam bucket berversi dihapus.

- Sebelum melakukan pembaruan apa pun pada konfigurasi Siklus Hidup S3 Anda, konfirmasikan bahwa Siklus Hidup telah menyelesaikan tindakan pada semua objek yang dimaksud. Untuk informasi selengkapnya, lihat bagian Memperbarui, menonaktifkan, atau menghapus aturan Siklus Hidup di Menyetel konfigurasi Siklus Hidup S3 pada bucket.

  > **Note**
  >
  > Aturan Siklus Hidup S3 harus diterapkan pada subset objek yang tepat untuk mencegah penghapusan yang tidak diinginkan. Anda dapat memfilter objek berdasarkan awalan, tag objek, atau ukuran objek saat membuat aturan Siklus Hidup.

- Pertimbangkan untuk membatasi pengguna untuk menghapus atau menghapus objek dari bucket Anda. Untuk membatasi pengguna, Anda harus secara eksplisit menolak izin pengguna untuk tindakan berikut dalam kebijakan bucket Amazon S3 Anda:
  - `s3:DeleteObject`, `s3:DeleteObjectVersion` (untuk mengontrol siapa yang dapat menghapus objek menggunakan permintaan API)
  - `s3:PutLifecycleConfiguration`(untuk mengontrol siapa yang dapat menambahkan aturan kedaluwarsa Siklus Hidup S3)
- Pertimbangkan untuk menggunakan Replikasi S3 untuk membuat beberapa salinan data Anda dan untuk mereplikasi mereka ke beberapa lokasi sekaligus. Anda dapat memilih ember tujuan sebanyak yang diperlukan. Selain itu, jika objek dihapus secara tidak sengaja, Anda masih akan memiliki salinan data.

Menghapus objek dari bucket berkemampuan versi

Jika bucket Anda mengaktifkan versi, beberapa versi objek yang sama dapat ada di bucket. Saat bekerja dengan bucket berkemampuan versi, operasi `Delete` API mengaktifkan opsi berikut:

- Tentukan permintaan penghapusan non-versi–Menentukan hanya kunci objek, dan bukan ID versi. Dalam hal ini, Amazon S3 membuat penanda hapus di atas versi objek saat ini dan mengembalikan ID versinya dalam respons. Ini akan menghilangkan objek Anda dari bucket

tersebut. Untuk informasi tentang pembuatan Penentuan Versi objek dan konsep penanda hapus, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.

- Tentukan permintaan penghapusan berversi - Tentukan kunci dan ID versi. Dalam hal ini, hasil berikut dimungkinkan:
  - Jika ID versi memetakan ke versi objek tertentu, Amazon S3 menghapus versi objek tertentu.
  - Jika ID versi memetakan ke penanda hapus suatu objek, Amazon S3 menghapus penanda hapus. Ketika penanda hapus dihapus, objek kemudian muncul kembali di ember Anda.

Menghapus objek dari bucket yang ditangguhkan versi

Jika bucket Anda ditangguhkan versi, operasi `Delete` API berperilaku sama untuk membuat versi bucket yang diaktifkan (kecuali saat versi saat ini memiliki ID versi null). Untuk informasi selengkapnya, lihat Menghapus objek dari bucket dengan Penentuan Versi ditangguhkan.

Menghapus objek dari bucket yang tidak berversi

Jika bucket tidak berversi, Anda dapat menentukan kunci objek dalam operasi `Delete` API dan Amazon S3 akan menghapus objek secara permanen. Untuk mencegah penghapusan permanen objek, aktifkan pembuatan versi bucket.

Menghapus objek dari bucket yang mengaktifkan MFA

Saat menghapus objek dari bucket yang mengaktifkan autentikasi multi-faktor (MFA), perhatikan hal berikut:

- Jika Anda menyediakan token MFA yang tidak valid, permintaan akan selalu gagal.
- Jika Anda memiliki bucket berkemampuan MFA dan Anda membuat permintaan penghapusan berversi (Anda memberikan kunci objek dan ID versi), permintaan tersebut akan gagal jika Anda tidak memberikan token MFA yang valid. Selain itu, saat menggunakan operasi `Delete` API multi-objek pada bucket berkemampuan MFA, jika salah satu penghapusan adalah permintaan penghapusan berversi (yaitu, Anda menentukan kunci objek dan ID versi), seluruh permintaan gagal jika Anda tidak memberikan token MFA.

Namun, dalam kasus berikut, permintaan berhasil:

- Jika Anda memiliki bucket berkemampuan MFA dan Anda membuat permintaan penghapusan yang tidak berversi (Anda tidak menghapus objek yang berversi), dan Anda tidak memberikan token MFA, maka penghapusannya berhasil.
- Jika Anda memiliki permintaan Penghapusan Multi-Objek yang hanya menentukan objek non-versi yang akan dihapus dari bucket yang mengaktifkan MFA dan Anda tidak memberikan token MFA, penghapusannya berhasil.

Untuk informasi tentang penghapusan MFA, lihat Mengonfigurasi penghapusan MFA.

Topik

- Menghapus satu objek
- Menghapus beberapa objek

Menghapus satu objek

Anda dapat menggunakan konsol Amazon S3 atau DELETE API untuk menghapus satu objek yang ada dari bucket S3. Untuk informasi selengkapnya tentang menghapus objek di Amazon S3, lihat Menghapus objek Amazon S3.

Karena semua objek dalam bucket S3 Anda mengeluarkan biaya penyimpanan, Anda sebaiknya menghapus objek yang tidak lagi dibutuhkan. Misalnya, jika Anda mengumpulkan file log, sebaiknya hapus file tersebut saat tidak diperlukan lagi. Anda dapat mengatur aturan siklus hidup untuk secara otomatis menghapus objek seperti file log. Untuk informasi selengkapnya, lihat the section called “Menetapkan konfigurasi siklus hidup”.

Untuk informasi tentang fitur dan harga Amazon S3, lihat Harga Amazon S3.

Menggunakan konsol S3

Ikuti langkah-langkah berikut ini untuk menggunakan konsol Amazon S3 guna menghapus satu objek dari bucket.

> **Warning**
>
> Saat Anda menghapus objek atau versi objek tertentu secara permanen di konsol Amazon S3, penghapusan tidak dapat dibatalkan.

Untuk menghapus objek yang memiliki versi diaktifkan atau ditangguhkan

> **Note**
>
> Jika ID versi untuk objek dalam bucket yang ditangguhkan versi ditandai sebagaiNULL, S3 akan menghapus objek secara permanen karena tidak ada versi sebelumnya. Namun, jika ID versi yang valid dicantumkan untuk objek dalam bucket yang ditangguhkan versi, maka S3 membuat penanda hapus untuk objek yang dihapus, sambil mempertahankan versi objek sebelumnya.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum atau ember Direktori.
3. Di bucket list, pilih nama bucket tempat Anda ingin menghapus objek.
4. Pilih objek dan kemudian pilih Hapus.
5. Untuk mengonfirmasi penghapusan daftar objek di bawah Objek yang ditentukan di objek Hapus? kotak teks, masukkan**`delete`**.

Untuk menghapus versi objek tertentu secara permanen dalam bucket berkemampuan versi

> **Warning**
>
> Saat Anda menghapus versi objek tertentu secara permanen di Amazon S3, penghapusan tidak dapat dibatalkan.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di dalam daftar Nama bucket, pilih nama bucket yang ingin Anda hapus objeknya.
3. Pilih objek yang ingin Anda hapus.
4. Pilih sakelar Tampilkan versi.
5. Pilih versi objek dan kemudian pilih Hapus.
6. Untuk mengonfirmasi penghapusan permanen dari versi objek tertentu yang tercantum di bawah Objek yang ditentukan, di objek Hapus? kotak teks, masukkan Hapus secara permanen. Amazon S3 secara permanen menghapus versi objek tertentu.

Untuk menghapus objek secara permanen di bucket Amazon S3 yang tidak mengaktifkan versi

> **Warning**
>
> Saat Anda menghapus objek secara permanen di Amazon S3, penghapusan tidak dapat dibatalkan. Juga, untuk ember apa pun tanpa mengaktifkan versi, penghapusan bersifat permanen.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Di bucket list, pilih nama bucket tempat Anda ingin menghapus objek.
4. Pilih objek dan kemudian pilih Hapus.
5. Untuk mengonfirmasi penghapusan permanen objek yang terdaftar di bawah Objek yang ditentukan, di objek Hapus? kotak teks, masukkan hapus secara permanen.

> **Note**
>
> Jika Anda mengalami masalah dengan menghapus objek Anda, lihatSaya ingin menghapus objek berversi secara permanen.

Menggunakan AWS CLI

Untuk menghapus satu objek per permintaan, gunakan API `DELETE`. Untuk informasi selengkapnya, lihat DELETE Object. Untuk informasi selengkapnya tentang menggunakan CLI untuk menghapus objek, lihat delete-object.

Penggunaan API REST

Anda dapat menggunakan AWS SDKs untuk menghapus objek. Namun, jika aplikasi Anda memerlukannya, Anda dapat mengirimkan permintaan REST secara langsung. Untuk informasi selengkapnya, lihat DELETE Object dalam Referensi API Amazon Simple Storage Service.

Menggunakan AWS SDKs

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan AWS SDKs untuk menghapus objek dari ember. Untuk informasi selengkapnya, lihat DELETE Object dalam Referensi API Amazon Simple Storage Service

Jika Penentuan Versi S3 Anda telah diaktifkan pada bucket tersebut, Anda memiliki opsi berikut:

- Menghapus versi objek tertentu dengan menentukan ID versi.
- Menghapus objek tanpa menentukan ID versi, di mana Amazon S3 menambahkan sebuah penanda hapus ke objek.

Untuk informasi selengkapnya tentang Penentuan Versi S3, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.

Untuk contoh selengkapnya, dan contoh dalam bahasa lain, lihat Menggunakan `DeleteObject` dengan AWS SDK atau CLI di referensi Amazon S3 API.

Java

Example Contoh 1: Menghapus sebuah objek (bucket non-versi)

Contoh berikut mengasumsikan bahwa bucket tidak diaktifkan versi dan objek tidak memiliki versi apa pun. IDs Dalam permintaan penghapusan, Anda hanya menentukan kunci objek dan bukan ID versi.

Untuk petunjuk cara membuat dan menguji sampel yang berfungsi, lihat Memulai di Panduan AWS SDK untuk Java Pengembang.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.DeleteObjectRequest;

import java.io.IOException;
```

```
public class DeleteObjectNonVersionedBucket {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String keyName = "*** Key name ****";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            s3Client.deleteObject(new DeleteObjectRequest(bucketName, keyName));
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Example Contoh 2: Menghapus sebuah objek (bucket berversi)

Contoh berikut menampilkan penghapusan sebuah objek dari sebuah bucket berversi. Contoh ini menghapus versi objek tertentu dengan menentukan nama kunci objek dan ID versi.

Contoh ini melakukan hal berikut:

1. Menambahkan sebuah objek sampel ke bucket. Amazon S3 menampilkan ID versi dari objek yang baru saja ditambahkan. Contohnya menggunakan ID versi ini dalam permintaan penghapusan.
2. Hapus versi objek dengan menentukan nama kunci objek maupun ID versi. Jika tidak ada versi lain dari objek tersebut, Amazon S3 akan sepenuhnya menghapus objek tersebut. Jika tidak, Amazon S3 hanya menghapus versi yang ditentukan.

> **Note**
>
> Anda bisa mendapatkan IDs versi objek dengan mengirimkan `ListVersions` permintaan.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.BucketVersioningConfiguration;
import com.amazonaws.services.s3.model.DeleteVersionRequest;
import com.amazonaws.services.s3.model.PutObjectResult;

import java.io.IOException;

public class DeleteObjectVersionEnabledBucket {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String keyName = "*** Key name ****";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            // Check to ensure that the bucket is versioning-enabled.
            String bucketVersionStatus =
 s3Client.getBucketVersioningConfiguration(bucketName).getStatus();
            if (!bucketVersionStatus.equals(BucketVersioningConfiguration.ENABLED))
 {
                System.out.printf("Bucket %s is not versioning-enabled.",
 bucketName);
            } else {
                // Add an object.
```

```
                PutObjectResult putResult = s3Client.putObject(bucketName, keyName,
                        "Sample content for deletion example.");
                System.out.printf("Object %s added to bucket %s\n", keyName,
 bucketName);

                // Delete the version of the object that we just created.
                System.out.println("Deleting versioned object " + keyName);
                s3Client.deleteVersion(new DeleteVersionRequest(bucketName, keyName,
 putResult.getVersionId()));
                System.out.printf("Object %s, version %s deleted\n", keyName,
 putResult.getVersionId());
            }
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

Contoh berikut menunjukkan cara untuk menghapus sebuah objek dari bucket berversi maupun non-versi. Untuk informasi selengkapnya tentang Penentuan Versi S3, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.

Example Penghapusan sebuah objek dari sebuah bucket non-versi

Contoh berikut menampilkan penghapusan sebuah objek dari sebuah bucket non-versi. Contoh mengasumsikan bahwa objek tidak memiliki versi IDs, jadi Anda tidak menentukan versi IDs. Anda hanya menentukan kunci objek.

Untuk informasi tentang menyiapkan dan menjalankan contoh kode, lihat Memulai SDK for .NET di AWSAWS SDK for .NET Developer Guide.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
```

```
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class DeleteObjectNonVersionedBucketTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string keyName = "*** object key ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            DeleteObjectNonVersionedBucketAsync().Wait();
        }
        private static async Task DeleteObjectNonVersionedBucketAsync()
        {
            try
            {
                var deleteObjectRequest = new DeleteObjectRequest
                {
                    BucketName = bucketName,
                    Key = keyName
                };

                Console.WriteLine("Deleting an object");
                await client.DeleteObjectAsync(deleteObjectRequest);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
 deleting an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 deleting an object", e.Message);
            }
        }
    }
```

```
}
```

Example Penghapusan sebuah objek dari sebuah bucket berversi

Contoh C# berikut menampilkan penghapusan sebuah objek dari sebuah bucket berversi. Ini menghapus versi objek tertentu dengan menentukan nama kunci objek dan ID versi.

Kode ini melakukan tugas-tugas berikut:

1. Mengaktifkan S3 Penentuan Versi pada bucket yang Anda tentukan (jika S3 Penentuan Versi sudah diaktifkan, hal ini tidak berpengaruh).
2. Menambahkan sebuah objek sampel ke bucket. Amazon S3 merespons dengan menampilkan ID versi dari objek yang baru saja ditambahkan. Contohnya menggunakan ID versi ini dalam permintaan penghapusan.
3. Hapus versi objek dengan menentukan nama kunci objek maupun ID versi.

> **Note**
>
> Anda dapat memperoleh ID versi dari sebuah objek dengan mengirimkan permintaan `ListVersions`.
>
> ```
> var listResponse = client.ListVersions(new ListVersionsRequest { BucketName
>  = bucketName, Prefix = keyName });
> ```

Untuk informasi tentang menyiapkan dan menjalankan contoh kode, lihat Memulai SDK for .NET di AWSAWS SDK for .NET Developer Guide.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class DeleteObjectVersion
    {
        private const string bucketName = "*** versioning-enabled bucket name ***";
        private const string keyName = "*** Object Key Name ***";
```

```
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            CreateAndDeleteObjectVersionAsync().Wait();
        }

        private static async Task CreateAndDeleteObjectVersionAsync()
        {
            try
            {
                // Add a sample object.
                string versionID = await PutAnObject(keyName);

                // Delete the object by specifying an object key and a version ID.
                DeleteObjectRequest request = new DeleteObjectRequest
                {
                    BucketName = bucketName,
                    Key = keyName,
                    VersionId = versionID
                };
                Console.WriteLine("Deleting an object");
                await client.DeleteObjectAsync(request);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
deleting an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
deleting an object", e.Message);
            }
        }

        static async Task<string> PutAnObject(string objectKey)
        {
            PutObjectRequest request = new PutObjectRequest
            {
```

```
                BucketName = bucketName,
                Key = objectKey,
                ContentBody = "This is the content body!"
            };
            PutObjectResponse response = await client.PutObjectAsync(request);
            return response.VersionId;
        }
    }
}
```

PHP

Contoh ini menunjukkan cara menggunakan kelas dari versi 3 AWS SDK untuk PHP untuk menghapus objek dari bucket yang tidak berversi. Untuk informasi tentang penghapusan sebuah objek dari sebuah bucket berversi, lihat Penggunaan API REST.

Untuk informasi selengkapnya tentang AWS SDK for Ruby API, AWS buka SDK for Ruby - Versi 2.

Contoh PHP berikut menghapus objek dari bucket. Karena contoh ini menunjukkan cara menghapus objek dari bucket yang non-versi, contoh ini hanya menyediakan nama bucket dan kunci objek (bukan ID versi) dalam permintaan penghapusan.

```
<?php

require 'vendor/autoload.php';

use Aws\S3\S3Client;
use Aws\S3\Exception\S3Exception;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// 1. Delete the object from the bucket.
try
{
    echo 'Attempting to delete ' . $keyname . '...' . PHP_EOL;
```

```
    $result = $s3->deleteObject([
        'Bucket' => $bucket,
        'Key'    => $keyname
    ]);

    if ($result['DeleteMarker'])
    {
        echo $keyname . ' was deleted or does not exist.' . PHP_EOL;
    } else {
        exit('Error: ' . $keyname . ' was not deleted.' . PHP_EOL);
    }
}
catch (S3Exception $e) {
    exit('Error: ' . $e->getAwsErrorMessage() . PHP_EOL);
}

// 2. Check to see if the object was deleted.
try
{
    echo 'Checking to see if ' . $keyname . ' still exists...' . PHP_EOL;

    $result = $s3->getObject([
        'Bucket' => $bucket,
        'Key'    => $keyname
    ]);

    echo 'Error: ' . $keyname . ' still exists.';
}
catch (S3Exception $e) {
    exit($e->getAwsErrorMessage());
}
```

Javascript

```
import { DeleteObjectCommand } from "@aws-sdk/client-s3";
import { s3Client } from "./libs/s3Client.js" // Helper function that creates Amazon
 S3 service client module.

export const bucketParams = { Bucket: "BUCKET_NAME", Key: "KEY" };

export const run = async () => {
  try {
    const data = await s3Client.send(new DeleteObjectCommand(bucketParams));
```

```
    console.log("Success. Object deleted.", data);
    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Menghapus beberapa objek

Karena semua objek dalam bucket S3 Anda mengeluarkan biaya penyimpanan, Anda sebaiknya menghapus objek yang tidak lagi dibutuhkan. Misalnya, jika Anda mengumpulkan file log, sebaiknya hapus file tersebut saat tidak diperlukan lagi. Anda dapat mengatur aturan siklus hidup untuk secara otomatis menghapus objek seperti file log. Untuk informasi selengkapnya, lihat the section called “Menetapkan konfigurasi siklus hidup”.

Untuk informasi tentang fitur dan harga Amazon S3, lihat Harga Amazon S3.

Anda dapat menggunakan konsol Amazon S3 AWS SDKs, atau REST API untuk menghapus beberapa objek secara bersamaan dari bucket S3.

Menggunakan konsol S3

Ikuti langkah-langkah berikut untuk menggunakan konsol Amazon S3 untuk menghapus beberapa objek dari bucket.

⚠ Warning

- Menghapus objek yang dipilih tadi tidak dapat diurungkan.
- Tindakan ini menghapus semua objek yang telah ditentukan. Saat menghapus folder, tunggu hingga tindakan penghapusannya selesai sebelum menambahkan objek baru ke folder tersebut. Jika tidak, objek baru mungkin juga terhapus.
- Saat menghapus objek dalam bucket tanpa mengaktifkan versi, termasuk bucket direktori, Amazon S3 akan menghapus objek secara permanen.
- Saat menghapus objek dalam bucket dengan versi bucket diaktifkan atau ditangguhkan, Amazon S3 membuat penanda hapus. Untuk informasi selengkapnya, lihat Bekerja dengan penanda hapus.

Untuk menghapus objek yang memiliki versi diaktifkan atau ditangguhkan

> **Note**
>
> Jika versi IDs untuk objek dalam bucket yang ditangguhkan versi ditandai sebagaiNULL, S3 akan menghapus objek secara permanen karena tidak ada versi sebelumnya. Namun, jika ID versi yang valid dicantumkan untuk objek dalam bucket yang ditangguhkan versi, maka S3 membuat penanda hapus untuk objek yang dihapus, sambil mempertahankan versi objek sebelumnya.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Di bucket list, pilih nama bucket tempat Anda ingin menghapus objeknya.
4. Pilih objek dan kemudian pilih Hapus.
5. Untuk mengonfirmasi penghapusan daftar objek di bawah Objek yang ditentukan di objek Hapus? kotak teks, masukkan**`delete`**.

Untuk menghapus versi objek tertentu secara permanen dalam bucket berkemampuan versi

> **Warning**
>
> Saat Anda menghapus versi objek tertentu secara permanen di Amazon S3, penghapusan tidak dapat dibatalkan.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Di bucket list, pilih nama bucket tempat Anda ingin menghapus objeknya.
4. Pilih objek yang ingin Anda hapus.
5. Pilih sakelar Tampilkan versi.
6. Pilih versi objek dan kemudian pilih Hapus.

7. Untuk mengonfirmasi penghapusan permanen dari versi objek tertentu yang tercantum di bawah Objek yang ditentukan, di objek Hapus? kotak teks, masukkan Hapus secara permanen. Amazon S3 secara permanen menghapus versi objek tertentu.

Untuk menghapus objek secara permanen di bucket Amazon S3 yang tidak mengaktifkan versi

> **Warning**
>
> Saat Anda menghapus objek secara permanen di Amazon S3, penghapusan tidak dapat dibatalkan. Selain itu, untuk bucket apa pun tanpa mengaktifkan versi, termasuk bucket direktori, penghapusan bersifat permanen.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum atau ember Direktori.
3. Di bucket list, pilih nama bucket tempat Anda ingin menghapus objeknya.
4. Pilih objek dan kemudian pilih Hapus.
5. Untuk mengonfirmasi penghapusan permanen objek yang terdaftar di bawah Objek yang ditentukan, di objek Hapus? kotak teks, masukkan hapus secara permanen.

> **Note**
>
> Jika Anda mengalami masalah dengan menghapus objek Anda, lihatSaya ingin menghapus objek berversi secara permanen.

Menggunakan AWS SDKs

Untuk contoh cara menghapus beberapa objek dengan AWS SDKs, lihat Contoh kode di Referensi API Amazon S3.

Untuk informasi umum tentang penggunaan yang berbeda AWS SDKs, lihat Mengembangkan dengan Amazon S3 menggunakan Referensi API Amazon S3 AWS SDKs di Amazon S3.

Penggunaan API REST

Anda dapat menggunakan AWS SDKs untuk menghapus beberapa objek menggunakan Multi-Object Delete API. Namun, jika aplikasi Anda memerlukannya, Anda dapat mengirimkan permintaan REST secara langsung.

Untuk informasi selengkapnya, lihat Menghapus Beberapa Objek dalam Referensi API Amazon Simple Storage Service.

Mengatur, membuat daftar, dan bekerja dengan objek Anda

Di Amazon S3, Anda dapat menggunakan prefiks untuk mengatur penyimpanan Anda. Prefiks adalah pembuatan grup logis dari objek dalam bucket. Nilai prefiks mirip dengan nama direktori yang memungkinkan Anda untuk menyimpan data serupa di bawah direktori yang sama dalam sebuah bucket. Ketika Anda mengunggah objek secara terprogram, Anda dapat menggunakan prefiks untuk organisasi data Anda.

Di konsol Amazon S3, prefiks disebut sebagai folder. Anda dapat melihat semua objek dan folder Anda di konsol S3 dengan menavigasi ke bucket. Anda juga dapat melihat informasi tentang setiap objek, termasuk properti objek.

Untuk informasi selengkapnya tentang daftar dan organisasi data Anda di Amazon S3, lihat topik berikut.

Topik

- Organisasi objek menggunakan prefiks
- Membuat daftar kunci objek secara terprogram
- Mengatur objek di konsol Amazon S3 dengan menggunakan folder
- Melihat properti objek di konsol Amazon S3
- Mengategorikan penyimpanan Anda menggunakan tag

Organisasi objek menggunakan prefiks

Anda dapat menggunakan prefiks untuk organisasi data yang Anda simpan di bucket Amazon S3. Prefiks adalah string karakter di bagian awal nama kunci objek. Prefiks dapat memiliki panjang berapa pun, tergantung pada panjang maksimum dari nama kunci objek (1.024 byte). Anda dapat menganggap prefiks sebagai cara untuk mengatur data Anda dengan cara yang mirip dengan direktori. Namun, prefiks bukan direktori.

Pencarian berdasarkan prefiks membatasi hasil hanya pada kunci yang dimulai dengan prefiks tertentu. Pembatas menyebabkan operasi daftar menggabungkan semua kunci yang memiliki prefiks yang sama ke dalam satu hasil daftar ringkasan.

Tujuan dari parameter prefiks dan pembatas adalah untuk membantu Anda mengatur dan kemudian menelusuri kunci Anda secara hierarki. Untuk melakukan hal ini, pertama-tama pilih sebuah pembatas untuk bucket Anda, seperti garis miring (/), yang tidak ada dalam nama kunci mana pun yang Anda antisipasi. Anda dapat menggunakan karakter lain sebagai pembatas. Tidak ada yang unik tentang karakter garis miring (/), tetapi ini adalah pembatas prefiks yang sangat umum. Selanjutnya, buat nama kunci Anda dengan menggabungkan semua tingkat hierarki yang memuatnya, pisahkan setiap tingkat dengan pembatas.

Misalnya, jika Anda menyimpan informasi tentang kota, Anda mungkin mengaturnya berdasarkan benua, lalu berdasarkan negara, lalu berdasarkan provinsi atau negara bagian. Karena nama-nama ini biasanya tidak mengandung tanda baca, Anda dapat menggunakan garis miring (/) sebagai pembatas. Contoh berikut menggunakan pembatas garis miring (/).

- Europe/France/Nouvelle-Aquitaine/Bordeaux
- Utara America/Canada/Quebec/Montreal
- Utara America/USA/Washington/Bellevue
- Utara America/USA/Washington/Seattle

Jika Anda menyimpan data untuk setiap kota di dunia dengan cara ini, maka pengelolaan namespace kunci datar akan terasa aneh. Dengan menggunakan `Prefix` dan `Delimiter` dengan operasi daftar, Anda dapat menggunakan hierarki yang telah Anda buat untuk mendaftarkan data Anda. Misalnya, untuk membuat daftar semua negara bagian di AS, atur `Delimiter='/'` dan `Prefix='North America/USA/'`. Untuk membuat daftar semua provinsi di Kanada yang datanya Anda miliki, atur `Delimiter='/'` dan `Prefix='North America/Canada/'`.

Untuk informasi selengkapnya tentang pembatas, prefiks, dan folder bersarang, lihat Perbedaan antara folder prefiks dan bersarang.

Membuat daftar objek menggunakan prefiks dan pembatas

Jika Anda mengeluarkan permintaan daftar dengan pembatas, Anda dapat menelusuri hierarki hanya pada satu tingkat, melompati dan meringkas (mungkin jutaan) kunci yang ada di tingkat yang lebih dalam. Misalnya, asumsikan bahwa Anda memiliki bucket (*`amzn-s3-demo-bucket`*) dengan kunci berikut:

`sample.jpg`

`photos/2006/January/sample.jpg`

`photos/2006/February/sample2.jpg`

`photos/2006/February/sample3.jpg`

`photos/2006/February/sample4.jpg`

Bucket sampel hanya memiliki objek `sample.jpg` pada tingkat root. Untuk mencantumkan objek tingkat root dalam bucket saja, Anda dapat mengirimkan permintaan GET pada bucket dengan karakter pembatas garis miring (`/`). Sebagai respons, Amazon S3 mengembalikan kunci objek `sample.jpg` karena tidak mengandung karakter pembatas `/`. Semua kunci lainnya mengandung karakter pembatas. Amazon S3 mengelompokkan kunci ini dan menampilkan satu `CommonPrefixes` elemen dengan nilai prefiks `photos/`, yang merupakan sebuah substring dari awal kunci ini hingga peristiwa pertama dari pembatas yang ditentukan.

Example

```
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>amzn-s3-demo-bucket</Name>
  <Prefix></Prefix>
  <Marker></Marker>
  <MaxKeys>1000</MaxKeys>
  <Delimiter>/</Delimiter>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>sample.jpg</Key>
    <LastModified>2011-07-24T19:39:30.000Z</LastModified>
    <ETag>&quot;d1a7fb5eab1c16cb4f7cf341cf188c3d&quot;</ETag>
    <Size>6</Size>
    <Owner>
      <ID>75cc57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
      <DisplayName>displayname</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <CommonPrefixes>
    <Prefix>photos/</Prefix>
  </CommonPrefixes>
</ListBucketResult>
```

Untuk informasi selengkapnya tentang membuat daftar kunci objek secara terprogram, lihat Membuat daftar kunci objek secara terprogram.

Membuat daftar kunci objek secara terprogram

Di Amazon S3, kunci dapat didaftarkan dengan prefiks. Anda dapat memilih prefiks umum untuk nama kunci terkait dan menandai kunci tersebut dengan karakter khusus yang membatasi hierarki. Anda kemudian dapat menggunakan operasi daftar untuk memilih dan menelusuri kunci secara hierarki. Hal ini mirip dengan bagaimana file disimpan dalam direktori dalam sistem file.

Amazon S3 mengekspos operasi daftar yang memungkinkan Anda menghitung kunci yang terdapat dalam bucket. Kunci yang dipilih untuk pendaftaran didasarkan pada bucket dan prefiks. Misalnya, bayangkan sebuah bucket bernama "`dictionary`" yang berisi kunci untuk setiap kata dalam bahasa Inggris. Anda dapat membuat panggilan untuk mencantumkan semua kunci dalam bucket yang dimulai dengan huruf "q". Hasil daftar selalu ditampilkan dalam urutan biner UTF-8.

Operasi daftar SOAP dan REST mengembalikan dokumen XML yang berisi nama kunci yang cocok dan informasi tentang objek yang diidentifikasi oleh setiap kunci.

> **Note**
>
> SOAP APIs for tidak tersedia untuk pelanggan baru, dan mendekati End of Life (EOL) pada 31 Agustus 2025. Kami menyarankan Anda menggunakan REST API atau AWS SDKs.

Kelompok kunci yang berbagi prefiks yang diakhiri oleh pembatas khusus dapat digabungkan dengan prefiks umum tersebut untuk tujuan pencatatan. Hal ini memungkinkan aplikasi untuk mengatur dan menelusuri kuncinya secara hierarki, seperti cara Anda mengatur file ke dalam direktori dalam sistem file.

Misalnya, untuk memperluas bucket kamus agar berisi lebih dari sekadar kata-kata bahasa Inggris, Anda dapat membentuk kunci dengan mengawali setiap kata dengan bahasanya dan pembatas, seperti "`French/logical`". Dengan menggunakan skema penamaan ini dan fitur daftar hierarki, Anda hanya dapat mengambil daftar kata-kata dalam bahasa Prancis. Anda juga dapat menelusuri daftar bahasa tingkat atas yang tersedia tanpa harus mengulangi semua kunci intervensi leksikografis. Untuk informasi selengkapnya tentang aspek pendaftaran ini, lihat Organisasi objek menggunakan prefiks.

API REST

Jika aplikasi Anda memerlukannya, Anda dapat mengirimkan permintaan REST secara langsung. Anda dapat mengirimkan permintaan GET untuk mengembalikan beberapa atau semua objek dalam bucket atau Anda dapat menggunakan kriteria pemilihan untuk mengembalikan subset objek dalam bucket. Untuk informasi selengkapnya, lihat GET Bucket (List Objects) Version 2 dalam Referensi API Amazon Simple Storage Service.

Membuat daftar implementasi efisiensi

Performa daftar tidak terpengaruh langsung oleh jumlah kunci dalam bucket Anda. Hal ini juga tidak terpengaruh dengan ada atau tidak adanya argumen `prefix`, `marker`, `maxkeys`, atau `delimiter`.

Iterasi melalui hasil multihalaman

Karena bucket dapat berisi kunci dalam jumlah yang hampir tidak terbatas, hasil lengkap dari kueri daftar bisa sangat besar. Untuk mengelola kumpulan hasil yang besar, API Amazon S3 mendukung penomoran halaman untuk membaginya menjadi beberapa respons. Setiap respons kunci daftar mengembalikan halaman hingga 1.000 kunci dengan indikator yang menunjukkan apakah respons terpotong. Anda mengirim serangkaian permintaan kunci daftar sampai Anda menerima semua kunci. AWS Pustaka pembungkus SDK menyediakan pagination yang sama.

Contoh

Saat mendaftarkan semua objek di dalam bucket Anda, perhatikan bahwa Anda harus memiliki izin `s3:ListBucket`.

CLI

daftar-objek

Contoh berikut menggunakan `list-objects` perintah untuk menampilkan nama-nama semua objek dalam bucket yang ditentukan:

```
aws s3api list-objects --bucket text-content --query 'Contents[].{Key: Key, Size:
 Size}'
```

Contoh menggunakan `--query` argumen untuk memfilter output `list-objects` turun ke nilai kunci dan ukuran untuk setiap objek

Untuk informasi selengkapnya tentang objek, lihat Bekerja dengan objek di Amazon S3.

- Untuk detail API, lihat ListObjectsdi Referensi AWS CLI Perintah.

ls

Contoh berikut mencantumkan semua objek dan awalan dalam ember dengan menggunakan perintah. `ls`

Untuk menggunakan perintah contoh ini, ganti *amzn-s3-demo-bucket* dengan nama bucket Anda.

```
$ aws s3 ls s3://amzn-s3-demo-bucket
```

- Untuk informasi selengkapnya tentang perintah tingkat tinggi`ls`, lihat Daftar bucket dan objek di AWS Command Line Interface Panduan Pengguna.

PowerShell

Alat untuk PowerShell V4

Contoh 1: Perintah ini mengambil informasi tentang semua item di bucket “test-files”.

```
Get-S3Object -BucketName amzn-s3-demo-bucket
```

Contoh 2: Perintah ini mengambil informasi tentang item "sample.txt" dari bucket “test-files”.

```
Get-S3Object -BucketName amzn-s3-demo-bucket -Key sample.txt
```

Contoh 3: Perintah ini mengambil informasi tentang semua item dengan awalan “sample” dari bucket “test-files”.

```
Get-S3Object -BucketName amzn-s3-demo-bucket -KeyPrefix sample
```

- Untuk detail API, lihat ListObjectsdi Referensi Alat AWS untuk PowerShell Cmdlet (V4).

Alat untuk PowerShell V5

Contoh 1: Perintah ini mengambil informasi tentang semua item di bucket “test-files”.

```
Get-S3Object -BucketName amzn-s3-demo-bucket
```

Contoh 2: Perintah ini mengambil informasi tentang item "sample.txt" dari bucket “test-files”.

```
Get-S3Object -BucketName amzn-s3-demo-bucket -Key sample.txt
```

Contoh 3: Perintah ini mengambil informasi tentang semua item dengan awalan “sample” dari bucket “test-files”.

```
Get-S3Object -BucketName amzn-s3-demo-bucket -KeyPrefix sample
```

- Untuk detail API, lihat ListObjectsdi Referensi Alat AWS untuk PowerShell Cmdlet (V5).

Mengatur objek di konsol Amazon S3 dengan menggunakan folder

Di bucket tujuan umum Amazon S3, objek adalah sumber daya utama, dan objek disimpan dalam ember. Bucket tujuan umum Amazon S3 memiliki struktur datar alih-alih hierarki seperti yang akan Anda lihat di sistem file. Namun, demi kemudahan organisasi, konsol Amazon S3 mendukung konsep folder sebagai pembuatan grup objek. Konsol melakukan ini dengan menggunakan prefiks nama bersama untuk objek yang dikelompokkan. Dengan kata lain, objek yang dikelompokkan memiliki nama yang dimulai dengan string umum. String umum ini, atau prefiks bersama, adalah nama folder. Nama objek juga disebut sebagai nama kunci.

Misalnya, Anda dapat membuat folder di bucket tujuan umum di konsol bernama `photos` dan menyimpan objek bernama `myphoto.jpg` di dalamnya. Objek kemudian disimpan dengan nama kunci `photos/myphoto.jpg`, tempat `photos/` adalah prefiksnya.

Berikut ini adalah dua contoh lainnya:

- Jika Anda memiliki tiga objek di bucket tujuan umum Anda— `logs/date1.txtlogs/date2.txt`,, dan `logs/date3.txt` —konsol akan menampilkan folder bernama`logs`. Jika Anda membuka folder di konsol tersebut, anda akan melihat tiga objek: `date1.txt`, `date2.txt`, dan `date3.txt`.
- Jika Anda memiliki objek bernama`photos/2017/example.jpg`, konsol akan menampilkan folder bernama `photos` yang berisi folder tersebut`2017`. Folder `2017` berisi objek`example.jpg`.

Anda dapat memiliki folder di dalam folder, tetapi bukan bucket di dalam bucket. Anda dapat mengunggah dan menyalin objek secara langsung ke dalam folder. Folder dapat dibuat, dihapus, dan dipublikasikan, tetapi tidak dapat diganti namanya. Objek dapat disalin dari satu folder ke folder lainnya.

Important

Saat Anda membuat folder di konsol Amazon S3, S3 membuat objek 0-byte. Kunci objek ini diatur ke nama folder yang Anda berikan ditambah karakter garis miring (/). Misalnya, di konsol Amazon S3, jika Anda membuat folder bernama `photos` di bucket, konsol Amazon S3 akan membuat objek 0-byte dengan kunci. `photos/` Konsol membuat objek ini untuk mendukung gagasan folder.

Selain itu, objek apa pun yang sudah ada sebelumnya yang diberi nama dengan karakter garis miring ke depan (/) muncul sebagai folder di konsol Amazon S3. Misalnya, objek dengan nama kunci `examplekeyname/` muncul sebagai folder di konsol Amazon S3 dan bukan sebagai objek. Jika tidak, ia berperilaku seperti objek lain dan dapat dilihat dan dimanipulasi melalui AWS Command Line Interface (AWS CLI), AWS SDKs, atau REST API.

Selain itu, Anda tidak dapat mengunggah objek yang memiliki nama kunci dengan karakter garis miring ke depan (/) dengan menggunakan konsol Amazon S3. Namun, Anda dapat mengunggah objek yang diberi nama dengan karakter trailing forward slash (/) dengan menggunakan AWS Command Line Interface (AWS CLI) AWS SDKs, atau REST API.

Selain itu, konsol Amazon S3 tidak menampilkan konten dan metadata untuk objek folder seperti halnya untuk objek lain. Saat Anda menggunakan konsol untuk menyalin objek bernama dengan karakter garis miring ke depan (/), folder baru dibuat di lokasi tujuan, tetapi data dan metadata objek tidak disalin. Juga, garis miring maju (/) dalam nama kunci objek mungkin memerlukan penanganan khusus. Untuk informasi selengkapnya, lihat Menamai objek Amazon S3.

Untuk membuat folder di ember direktori, unggah folder. Untuk informasi selengkapnya, lihat Mengunggah objek ke ember direktori.

Topik

- Membuat folder
- Menjadikan folder publik
- Menghitung ukuran folder
- Menghapus folder

Membuat folder

Bagian ini menjelaskan cara menggunakan konsol Amazon S3 untuk membuat folder.

> **Important**
>
> Jika kebijakan bucket Anda mencegah pengunggahan objek ke bucket ini tanpa tag, metadata, atau penerima daftar kontrol akses (ACL), Anda tidak dapat membuat folder dengan menggunakan prosedur berikut. Sebaliknya, unggah folder kosong dan tentukan pengaturan berikut di konfigurasi unggahan.

Untuk membuat folder

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Dalam daftar bucket, pilih nama bucket tempat Anda ingin membuat folder.
4. Pada tab Objek, pilih Buat folder.
5. Masukkan nama untuk folder (misalnya, **`favorite-pics`**).

   > **Note**
   >
   > Nama folder tunduk pada batasan dan pedoman tertentu, dan dianggap sebagai bagian dari nama kunci objek objek, yang dibatasi hingga 1.024 byte. Untuk informasi selengkapnya, lihat the section called “Penamaan objek”.

6. (Opsional) Jika kebijakan bucket Anda mengharuskan objek dienkripsi dengan kunci enkripsi tertentu, di bawah Enkripsi sisi server, Anda harus memilih Tentukan kunci enkripsi dan menentukan kunci enkripsi yang sama saat membuat folder. Jika tidak, pembuatan folder akan gagal.
7. Pilih Buat folder.

Menjadikan folder publik

Kami menyarankan untuk memblokir semua akses publik ke folder dan bucket Amazon S3 Anda, kecuali Anda secara khusus memerlukan folder atau bucket publik. Saat Anda membuat folder publik, siapa pun di internet dapat melihat semua objek yang dikelompokkan ke dalam folder itu.

Di konsol Amazon S3, Anda dapat membuat folder menjadi publik. Anda juga dapat menjadikan folder publik dengan membuat kebijakan bucket yang membatasi akses data berdasarkan prefiks. Untuk informasi selengkapnya, lihat Identity and Access Management untuk Amazon S3.

> **Warning**
>
> Setelah Anda menjadikan folder publik di konsol Amazon S3, Anda tidak dapat menjadikannya pribadi lagi. Sebaliknya, Anda harus mengatur izin untuk setiap objek individu di dalam folder publik sehingga objek tidak memiliki akses publik. Untuk informasi selengkapnya, lihat Mengkonfigurasi ACLs.

Topik

- Menghitung ukuran folder
- Menghapus folder

Menghitung ukuran folder

Bagian ini menjelaskan cara menggunakan konsol Amazon S3 untuk menghitung ukuran folder.

Untuk menghitung ukuran folder

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Dalam daftar Bucket tujuan umum, pilih nama bucket tempat folder Anda disimpan.
4. Dalam daftar Objek, pilih kotak centang di sebelah nama folder.
5. Pilih Tindakan, lalu pilih Hitung ukuran total.

> **Note**
>
> Saat Anda menjauh dari halaman, informasi folder (termasuk ukuran total) tidak lagi tersedia. Anda harus menghitung ukuran total lagi jika Anda ingin melihatnya lagi.

> **Important**
>
> Saat Anda menggunakan tindakan Hitung ukuran total pada objek atau folder yang ditentukan dalam bucket, Amazon S3 menghitung jumlah total objek dan ukuran penyimpanan total. Namun, unggahan multibagian yang tidak lengkap atau sedang berlangsung dan versi sebelumnya atau yang tidak terkini tidak dihitung dalam jumlah total objek atau ukuran total. Tindakan ini hanya menghitung jumlah total objek dan ukuran total untuk versi saat ini atau terbaru dari setiap objek yang disimpan dalam bucket.
> Contoh, jika ada dua versi objek di bucket Anda, maka kalkulator penyimpanan di Amazon S3 menghitungnya hanya sebagai satu objek. Akibatnya, jumlah total objek yang dihitung di konsol Amazon S3 dapat berbeda dari metrik Hitungan Objek yang ditunjukkan di Lensa Penyimpanan S3 dan dari jumlah yang dilaporkan oleh metrik Amazon CloudWatch . `NumberOfObjects` Demikian juga, ukuran penyimpanan total juga dapat berbeda dari metrik Total Storage yang ditunjukkan pada S3 Storage Lens dan dari `BucketSizeBytes` metrik yang ditunjukkan pada CloudWatch.

Menghapus folder

Bagian ini menjelaskan cara menggunakan konsol Amazon S3 untuk menghapus folder dari bucket S3.

Untuk informasi selengkapnya tentang fitur dan harga Amazon S3, lihat Amazon S3.

Untuk menghapus folder dari bucket S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Dalam daftar Bucket tujuan umum, pilih nama bucket tempat Anda ingin menghapus foldernya.
4. Dalam daftar Objek, pilih kotak centang di sebelah folder dan objek yang ingin Anda hapus.
5. Pilih Hapus.
6. Pada halaman Hapus objek, verifikasi bahwa nama folder dan objek yang Anda pilih untuk dihapus tercantum di bawah Objek yang ditentukan.
7. Di dalam kotak Hapus objek, masukkan **`delete`**, dan pilih Hapus objek.

> **Warning**
>
> Tindakan ini menghapus semua objek yang telah ditentukan. Saat menghapus folder, tunggu hingga tindakan penghapusannya selesai sebelum menambahkan objek baru ke folder tersebut. Jika tidak, objek baru mungkin juga terhapus.

Melihat properti objek di konsol Amazon S3

Anda dapat menggunakan konsol Amazon S3 untuk melihat properti sebuah objek, termasuk kelas penyimpanan, pengaturan enkripsi, tanda, dan metadata.

Untuk melihat properti sebuah objek

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum atau ember Direktori.
3. Di bucket list, pilih nama bucket yang berisi objek.
4. Di dalam daftar Objek, pilih nama objek yang ingin Anda lihat propertinya.

   Ikhtisar objek untuk objek Anda terbuka. Anda dapat menggulir ke bawah untuk melihat properti objek.
5. Pada halaman Ikhtisar objek, Anda dapat melihat atau mengonfigurasi properti berikut untuk objek tersebut.

   > **Note**
   >
   > - Jika Anda mengubah properti Kelas Penyimpanan, Enkripsi, atau Metadata, sebuah objek baru dibuat untuk menggantikan yang lama. Jika Penentuan Versi S3 diaktifkan, versi baru objek akan dibuat, dan objek yang sudah ada menjadi versi yang lebih lama. Peran yang mengubah properti juga menjadi pemilik objek baru (atau versi objek).
   > - Jika Anda mengubah properti Kelas Penyimpanan, Enkripsi, atau Metadata untuk objek yang memiliki tag yang ditentukan pengguna, Anda harus memiliki izin. `s3:GetObjectTagging` Jika Anda mengubah properti ini untuk objek yang tidak memiliki tag yang ditentukan pengguna tetapi berukuran lebih dari 16 MB, Anda juga harus memiliki izin. `s3:GetObjectTagging`

Jika kebijakan bucket tujuan menolak `s3:GetObjectTagging` tindakan, properti ini untuk objek akan diperbarui, tetapi tag yang ditentukan pengguna akan dihapus dari objek, dan Anda akan menerima kesalahan.

a. Kelas penyimpanan–Setiap objek di Amazon S3 memiliki kelas penyimpanan yang terkait dengannya. Kelas penyimpanan yang Anda pilih untuk digunakan tergantung pada seberapa sering Anda mengakses objeknya. Kelas penyimpanan default untuk objek S3 dalam bucket tujuan umum adalah STANDARD. Kelas penyimpanan default untuk objek S3 dalam bucket direktori adalah S3 Express One Zone. Anda memilih kelas penyimpanan mana yang akan digunakan saat mengunggah sebuah objek. Untuk informasi selengkapnya tentang kelas penyimpanan, lihat Memahami dan mengelola kelas penyimpanan Amazon S3.

   Untuk mengubah kelas penyimpanan setelah Anda mengunggah objek ke bucket tujuan umum, pilih kelas Storage. Pilih kelas penyimpanan yang Anda inginkan, lalu pilih Simpan.

   > **Note**
   >
   > Kelas penyimpanan objek dalam bucket direktori tidak dapat diubah.

b. Pengaturan enkripsi di sisi server–Anda dapat menggunakan enkripsi di sisi server untuk mengenkripsi objek S3 Anda. Untuk informasi selengkapnya, lihat Menentukan enkripsi di sisi server dengan AWS KMS (SSE-KMS) atau Menentukan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3).

c. Metadata–Setiap objek di Amazon S3 memiliki serangkaian pasangan nilai-nama yang mewakili metadata-nya. Untuk informasi tentang menambahkan metadata ke objek S3, lihat Mengedit metadata objek di konsol Amazon S3.

d. Tag — Anda mengkategorikan penyimpanan dengan menambahkan tag ke objek S3 dalam bucket tujuan umum. Untuk informasi selengkapnya, lihat Mengategorikan penyimpanan Anda menggunakan tag.

e. Penahanan dan penyimpanan legal kunci objek — Anda dapat mencegah objek dalam bucket tujuan umum dihapus. Untuk informasi selengkapnya, lihat Mengunci objek dengan Object Lock.

Mengategorikan penyimpanan Anda menggunakan tag

Gunakan pemberian tag objek untuk mengategorikan penyimpanan. Setiap tag adalah pasangan nilai kunci.

Anda dapat menambahkan tag ke objek baru saat mengunggahnya, atau Anda dapat menambahkannya ke objek yang sudah ada.

- Anda dapat mengaitkan hingga 10 tag dengan objek. Tag yang terkait dengan sebuah objek harus memiliki kunci tag unik.
- Kunci tanda dapat terdiri dari hingga 128 karakter Unicode, dan nilai tanda dapat terdiri dari hingga 256 karakter Unicode. Tag objek Amazon S3 diwakili secara internal dalam UTF-16. Perhatikan bahwa dalam UTF-16, karakter menggunakan 1 atau 2 posisi karakter.
- Kunci dan nilai peka terhadap huruf besar dan kecil.
- Untuk informasi selengkapnya tentang pembatasan tag, lihat Pembatasan tag yang ditentukan pengguna di Panduan Pengguna AWS Billing and Cost Management. Untuk pembatasan tag dasar, lihat Pembatasan tag di Panduan EC2 Pengguna Amazon.

Contoh

Pertimbangkan contoh pemberian tag berikut:

Example Informasi PHI

Misalkan objek berisi data informasi kesehatan yang dilindungi (PHI). Anda dapat memberi tag pada objek menggunakan pasangan nilai kunci berikut.

```
PHI=True
```

atau

```
Classification=PHI
```

Example File proyek

Misalkan, Anda menyimpan berkas proyek dalam bucket S3. Anda dapat memberi tag objek ini dengan kunci yang diberi nama `Project` dan nilai, seperti yang ditunjukkan berikut ini.

```
Project=Blue
```

Example Banyak tag

Anda dapat menambahkan beberapa tag ke sebuah objek, seperti yang ditunjukkan berikut.

```
Project=x
Classification=confidential
```

Prefiks nama kunci dan tag

Prefiks kunci objek juga memungkinkan Anda untuk mengategorikan penyimpanan. Namun, kategorisasi berbasis prefiks bersifat satu dimensi. Pertimbangkan nama kunci objek berikut:

```
photos/photo1.jpg
project/projectx/document.pdf
project/projecty/document2.pdf
```

Nama-nama kunci ini memiliki prefiks `photos/`, `project/projectx/`, dan `project/projecty/`. Prefiks ini mengaktifkan kategorisasi satu dimensi. Artinya, semua yang berada di bawah prefix adalah satu kategori. Misalnya, prefiks `project/projectx` mengidentifikasi semua dokumen yang terkait dengan proyek x.

Dengan pemberian tag, Anda sekarang memiliki dimensi lain. Jika Anda ingin foto1 dalam kategori proyek x, Anda dapat memberi tag objek yang sesuai.

Manfaat tambahan

Selain klasifikasi data, pemberian tag menawarkan manfaat seperti berikut ini:

- Tag objek memungkinkan kontrol izin akses yang detail. Misalnya, Anda dapat memberikan izin pengguna ke objek hanya-baca dengan tag tertentu.
- Tag objek memungkinkan manajemen siklus hidup objek yang detail yaitu Anda dapat menentukan filter berbasis tag, selain prefiks nama kunci, dalam aturan siklus hidup.
- Saat menggunakan analitik Amazon S3, Anda dapat mengonfigurasi filter agar membuat grup objek secara bersama-sama untuk analisis dengan tag objek, dengan prefiks nama kunci, atau dengan prefiks dan tag.
- Anda juga dapat menyesuaikan CloudWatch metrik Amazon untuk menampilkan informasi berdasarkan filter tag tertentu. Bagian berikut memberikan perincian.

> **Important**
>
> Dapat menggunakan tag untuk melabeli objek yang berisi data rahasia, seperti informasi pengenal pribadi (PII) atau informasi kesehatan terlindungi (PHI). Namun demikian, tag itu sendiri seharusnya tidak berisi informasi rahasia.

Menambahkan set tag objek ke beberapa objek Amazon S3 dengan permintaan tunggal

Untuk menambahkan set tag objek ke lebih dari satu objek Amazon S3 dengan permintaan tunggal, Anda dapat menggunakan Operasi Batch S3. Anda menyediakan daftar objek yang akan dioperasikan kepada Operasi Batch S3. Operasi Batch S3 akan memanggil masing-masing operasi API untuk melakukan operasi tertentu. Satu tugas Operasi Batch dapat melakukan operasi tertentu pada miliaran objek yang berisi data sebesar eksabita.

Operasi Batch S3 memiliki fitur melacak progres, mengirimkan notifikasi, dan menyimpan laporan penyelesaian terperinci dari semua tindakan, menyediakan pengalaman yang dikelola sepenuhnya, dapat diaudit, dan nirserver. Anda dapat menggunakan Operasi Batch S3 melalui konsol Amazon S3 AWS CLI, AWS SDKs, atau REST API. Untuk informasi selengkapnya, lihat the section called “Dasar-dasar Operasi Batch”.

Untuk informasi selengkapnya tentang tag objek, lihat Mengelola tag objek.

Operasi API terkait pemberian tag objek

Amazon S3 mendukung operasi API berikut yang khusus untuk pemberian tag objek:

Operasi API objek

- Pemberian tag PUT Objek–Mengganti tag pada satu objek. Anda menentukan tag di isi permintaan. Ada dua skenario berbeda dalam pengelolaan tag objek menggunakan API ini.
  - Objek tidak memiliki tag–Dengan menggunakan API ini, Anda dapat menambahkan satu set tag ke objek (objek tidak memiliki tag sebelumnya).
  - Objek memiliki satu set tag yang sudah ada–Untuk memodifikasi tag yang sudah ada, Anda harus mengambil set tag yang sudah ada terlebih dahulu, mengubahnya dari sisi klien, kemudian menggunakan API ini untuk mengganti set tag tersebut.

> **Note**
>
> Jika Anda mengirim permintaan ini dengan set tag kosong, Amazon S3 menghapus tag yang ada yang ditetapkan di objek. Jika Anda menggunakan metode ini, Anda akan dikenakan biaya untuk Permintaan Tingkat 1 (PUT). Untuk informasi selengkapnya, lihat harga Amazon S3.
> Permintaan pemberian tag DELETE Object lebih disarankan karena mencapai hasil yang sama tanpa dikenakan biaya.

- Pemberian tag GET Object–Mengembalikan set tag yang terkait dengan objek. Amazon S3 mengembalikan tag objek di dalam isi respons.
- Pemberian tag DELETE Object–Menghapus set tag yang terkait dengan objek.

Operasi API lainnya yang mendukung pemberian tag

- PUT Objek dan Memulai Pengunggahan Multibagian– Anda dapat menentukan tag saat membuat objek. Anda menentukan tag menggunakan header permintaan `x-amz-tagging`.
- GET Object–Alih-alih mengembalikan set tag, Amazon S3 mengembalikan jumlah tag objek dalam header `x-amz-tag-count` (hanya jika pemohon memiliki izin untuk membaca tag) karena ukuran respons header terbatas hingga 8 K byte. Jika Anda ingin melihat tag, Anda membuat permintaan lain untuk operasi API pemberian tag GET Object.
- Objek POST–Anda dapat menentukan tag di permintaan POST.

  Selama tag dalam permintaan Anda tidak melebihi batas ukuran header permintaan HTTP 8 K byte, Anda dapat menggunakan API `PUT Object` untuk membuat objek dengan tag. Jika tag yang Anda tetapkan melebihi batas ukuran header, Anda dapat menggunakan metode POST ini yaitu Anda menyertakan tag di dalam bodi.

  PUT Objek-Salin–Anda dapat menentukan `x-amz-tagging-directive` dalam permintaan Anda untuk mengarahkan Amazon S3 agar menyalin (perilaku default) tag atau mengganti tag dengan set tag baru yang diberikan dalam permintaan.

Perhatikan hal berikut:

- Pemberian Tag Objek S3 sangat konsisten. Untuk informasi selengkapnya, lihat Model konsistensi data Amazon S3.

Konfigurasi tambahan

Bagian ini menjelaskan bagaimana pemberian tag objek berkaitan dengan konfigurasi lain.

Pemberian tag objek dan manajemen siklus hidup

Dalam konfigurasi siklus operasional bucket, Anda dapat menetapkan filter untuk memilih subset objek yang aturannya berlaku. Anda dapat menentukan filter berdasarkan prefiks nama kunci, tag objek, atau keduanya.

Misalkan Anda menyimpan foto (format mentah dan yang sudah jadi) dalam bucket Amazon S3. Anda dapat memberi tag objek ini seperti yang ditunjukkan berikut ini.

```
phototype=raw
or
phototype=finished
```

Anda dapat mempertimbangkan untuk mengarsipkan foto mentah ke S3 Glacier beberapa saat setelah foto tersebut dibuat. Anda dapat mengonfigurasi aturan siklus aktif dengan filter yang mengidentifikasi subset objek dengan prefiks nama kunci (`photos/`) yang memiliki tag tertentu (`phototype=raw`).

Untuk informasi selengkapnya, lihat Mengelola siklus hidup objek.

Pemberian tag dan replikasi objek

Jika Anda mengonfigurasi Replikasi pada bucket, Amazon S3 mereplikasi tag, asalkan Anda memberikan izin Amazon S3 untuk membaca tag tersebut. Untuk informasi selengkapnya, lihat Menyiapkan ikhtisar replikasi langsung.

Pemberitahuan peristiwa pemberian tag objek

Anda dapat mengatur pemberitahuan peristiwa Amazon S3 untuk menerima pemberitahuan saat tag objek ditambahkan atau dihapus dari objek. Jenis peristiwa `s3:ObjectTagging:Put` memberi tahu Anda saat tag PUT pada objek atau saat tag yang ada diperbarui. Jenis peristiwa `s3:ObjectTagging:Delete` memberi tahu Anda saat tag dihapus dari objek. Untuk informasi selengkapnya, lihat Mengaktifkan pemberitahuan peristiwa.

Untuk informasi lebih lanjut tentang pemberian tag objek, lihat topik berikut ini:

Topik

- Kebijakan pemberian tag dan kontrol akses
- Mengelola tag objek

Kebijakan pemberian tag dan kontrol akses

Anda juga dapat menggunakan kebijakan izin (bucket dan kebijakan pengguna) untuk mengelola izin terkait pemberian tag objek. Untuk tindakan kebijakan, lihat topik berikut:

- Operasi objek
- Operasi bucket

Tag objek memungkinkan kontrol akses yang detail untuk mengelola izin. Anda dapat memberikan izin bersyarat berdasarkan tag objek. Amazon S3 mendukung kunci kondisi berikut yang dapat Anda gunakan untuk memberikan izin bersyarat berdasarkan tag objek:

- `s3:ExistingObjectTag/<tag-key>`–Gunakan kunci kondisi ini untuk memverifikasi bahwa tag objek yang ada memiliki kunci dan nilai tag spesifik.

  > **Note**
  >
  > Saat memberikan izin untuk operasi `PUT Object` dan `DELETE Object`, kunci kondisi ini tidak didukung. Artinya, Anda tidak dapat membuat kebijakan memberikan atau menolak izin pengguna untuk menghapus atau menimpa objek berdasarkan tag yang ada.

- `s3:RequestObjectTagKeys`–Gunakan kunci kondisi ini untuk membatasi kunci tag yang ingin Anda izinkan pada objek. Ini berguna saat menambahkan tag ke objek menggunakan permintaan objek PutObjectTagging dan PutObject, dan POST.
- `s3:RequestObjectTag/<tag-key>`–Gunakan kunci kondisi ini untuk membatasi kunci dan nilai tag yang ingin Anda izinkan pada objek. Ini berguna saat menambahkan tag ke objek menggunakan permintaan PutObjectTagging dan PutObject, dan POST Bucket.

Untuk daftar lengkap kunci kondisi Amazon S3, lihat Contoh kebijakan bucket menggunakan tombol kondisi. Kebijakan izin berikut menggambarkan cara pemberian tag objek yang memungkinkan manajemen izin akses yang detail.

Example 1: Mengizinkan pengguna untuk membaca hanya objek yang memiliki tag dan nilai kunci tertentu

Kebijakan izin berikut membatasi pengguna agar hanya membaca objek yang memiliki kunci dan nilai tag `environment: production`. Kebijakan ini menggunakan kunci kondisi `s3:ExistingObjectTag` untuk menentukan kunci tag dan nilai.

```
{
  "Version": "2012-10-17",
  "Statement": [
  {
    "Principal": {
      "AWS": [
        "arn:aws:iam::111122223333:role/JohnDoe"
      ]
    },
    "Effect": "Allow",
    "Action": ["s3:GetObject", "s3:GetObjectVersion"],
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
    "Condition": {
      "StringEquals":
        {"s3:ExistingObjectTag/environment": "production"}
    }
  }
  ]
}
```

Example 2: Batasi kunci tag objek mana yang dapat ditambahkan pengguna

Kebijakan izin berikut memberikan izin pengguna untuk melakukan tindakan `s3:PutObjectTagging`, yang memungkinkan pengguna menambahkan tag ke objek yang sudah ada. Kondisi ini menggunakan kunci kondisi `s3:RequestObjectTagKeys` untuk menentukan kunci tag yang diizinkan, seperti `Owner` atau `CreationDate`. Untuk informasi selengkapnya, lihat Membuat kondisi yang menguji beberapa nilai kunci dalam Panduan Pengguna IAM.

Kebijakan ini memastikan agar setiap kunci tag yang ditentukan dalam permintaan adalah kunci tag yang diotorisasi. Kualifikasi `ForAnyValue` dalam kondisi tersebut memastikan bahwa setidaknya ada satu kunci yang ditentukan dalam permintaan.

```
{
   "Version": "2012-10-17",
  "Statement": [
```

```
    {"Principal":{"AWS":[
            "arn:aws:iam::111122223333:role/JohnDoe"
         ]
       },
 "Effect": "Allow",
      "Action": [
        "s3:PutObjectTagging"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ],
      "Condition": {"ForAnyValue:StringEquals": {"s3:RequestObjectTagKeys": [
            "Owner",
            "CreationDate"
          ]
        }
      }
    }
  ]
}
```

Example 3: Memerlukan kunci dan nilai tag spesifik saat mengizinkan pengguna untuk menambahkan tag objek

Kebijakan contoh berikut memberikan izin pengguna untuk melakukan `s3:PutObjectTagging` tindakan, yang memungkinkan pengguna menambahkan tag ke objek yang sudah ada. Kondisi ini mengharuskan pengguna untuk menyertakan kunci tag tertentu (seperti *`Project`*) dengan set nilai ke *`X`*.

```
{
   "Version": "2012-10-17",
  "Statement": [
    {"Principal":{"AWS":[
       "arn:aws:iam::111122223333:user/JohnDoe"
        ]
       },
      "Effect": "Allow",
      "Action": [
        "s3:PutObjectTagging"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ],
```

```
      "Condition": {"StringEquals": {"s3:RequestObjectTag/Project": "X"
        }
      }
    }
  ]
}
```

Mengelola tag objek

Bagian ini menjelaskan bagaimana Anda dapat mengelola tag objek menggunakan AWS SDKs untuk Java dan .NET atau konsol Amazon S3.

Penandaan objek memberi Anda cara untuk mengkategorikan penyimpanan dalam ember tujuan umum. Setiap tag adalah pasangan nilai kunci yang mematuhi aturan berikut:

- Anda dapat mengaitkan hingga 10 tag dengan objek. Tag yang terkait dengan sebuah objek harus memiliki kunci tag unik.
- Kunci tanda dapat terdiri dari hingga 128 karakter Unicode, dan nilai tanda dapat terdiri dari hingga 256 karakter Unicode. Tag objek Amazon S3 diwakili secara internal dalam UTF-16. Perhatikan bahwa dalam UTF-16, karakter menggunakan 1 atau 2 posisi karakter.
- Kunci dan nilai peka terhadap huruf besar dan kecil.

Untuk informasi selengkapnya tentang tag objek, lihat Mengategorikan penyimpanan Anda menggunakan tag. Untuk informasi selengkapnya tentang pembatasan tag, lihat Pembatasan Tag yang Ditentukan Pengguna dalam AWS Manajemen Penagihan dan Biaya Panduan Pengguna.

Menggunakan konsol S3

Untuk menambahkan tag ke objek

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Di bucket list, pilih nama bucket yang berisi objek.
4. Pilih kotak centang di sebelah kiri nama objek yang ingin Anda ubah.
5. Di menu Tindakan, pilih Edit tag.
6. Tinjau objek yang tercantum, lalu pilih Tambahkan tag.

7. Setiap tag objek adalah pasangan nilai kunci. Masukkan Kunci dan Nilai. Untuk menambahkan tag lainnya, pilih Tambahkan tag.

   Anda dapat memasukkan hingga 10 tag untuk objek.

8. Pilih Simpan perubahan.

   Amazon S3 menambahkan tag ke objek tertentu.

Untuk informasi selengkapnya, lihat juga Melihat properti objek di konsol Amazon S3 dan Mengunggah Objek dalam panduan ini.

Menggunakan AWS SDKs

Java

Contoh berikut menunjukkan bagaimana menggunakan untuk mengatur tag untuk objek baru dan mengambil atau mengganti tag untuk objek yang ada. AWS SDK untuk Java Untuk informasi selengkapnya tentang pemberian tag objek, lihat Mengategorikan penyimpanan Anda menggunakan tag. Untuk petunjuk cara membuat dan menguji sampel yang berfungsi, lihat Memulai di Panduan AWS SDK untuk Java Pengembang.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.io.File;
import java.util.ArrayList;
import java.util.List;

public class ManagingObjectTags {

    public static void main(String[] args) {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String keyName = "*** Object key ***";
        String filePath = "*** File path ***";
```

```
        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            // Create an object, add two new tags, and upload the object to Amazon
 S3.
            PutObjectRequest putRequest = new PutObjectRequest(bucketName, keyName,
 new File(filePath));
            List<Tag> tags = new ArrayList<Tag>();
            tags.add(new Tag("Tag 1", "This is tag 1"));
            tags.add(new Tag("Tag 2", "This is tag 2"));
            putRequest.setTagging(new ObjectTagging(tags));
            PutObjectResult putResult = s3Client.putObject(putRequest);

            // Retrieve the object's tags.
            GetObjectTaggingRequest getTaggingRequest = new
 GetObjectTaggingRequest(bucketName, keyName);
            GetObjectTaggingResult getTagsResult =
 s3Client.getObjectTagging(getTaggingRequest);

            // Replace the object's tags with two new tags.
            List<Tag> newTags = new ArrayList<Tag>();
            newTags.add(new Tag("Tag 3", "This is tag 3"));
            newTags.add(new Tag("Tag 4", "This is tag 4"));
            s3Client.setObjectTagging(new SetObjectTaggingRequest(bucketName,
 keyName, new ObjectTagging(newTags)));
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

Contoh berikut menunjukkan bagaimana menggunakan AWS SDK untuk .NET untuk mengatur tag untuk objek baru dan mengambil atau mengganti tag untuk objek yang ada. Untuk informasi selengkapnya tentang pemberian tag objek, lihat Mengategorikan penyimpanan Anda menggunakan tag.

Untuk informasi tentang menyiapkan dan menjalankan contoh kode, lihat Memulai SDK for .NET di AWSAWS SDK for .NET Developer Guide.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    public class ObjectTagsTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string keyName = "*** key name for the new object ***";
        private const string filePath = @"*** file path ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            PutObjectWithTagsTestAsync().Wait();
        }

        static async Task PutObjectWithTagsTestAsync()
        {
            try
            {
                // 1. Put an object with tags.
                var putRequest = new PutObjectRequest
                {
                    BucketName = bucketName,
```

```
                    Key = keyName,
                    FilePath = filePath,
                    TagSet = new List<Tag>{
                        new Tag { Key = "Keyx1", Value = "Value1"},
                        new Tag { Key = "Keyx2", Value = "Value2" }
                    }
                };

                PutObjectResponse response = await
client.PutObjectAsync(putRequest);
                // 2. Retrieve the object's tags.
                GetObjectTaggingRequest getTagsRequest = new GetObjectTaggingRequest
                {
                    BucketName = bucketName,
                    Key = keyName
                };

                GetObjectTaggingResponse objectTags = await
client.GetObjectTaggingAsync(getTagsRequest);
                for (int i = 0; i < objectTags.Tagging.Count; i++)
                    Console.WriteLine("Key: {0}, Value: {1}",
objectTags.Tagging[i].Key, objectTags.Tagging[i].Value);


                // 3. Replace the tagset.

                Tagging newTagSet = new Tagging();
                newTagSet.TagSet = new List<Tag>{
                    new Tag { Key = "Key3", Value = "Value3"},
                    new Tag { Key = "Key4", Value = "Value4" }
                };


                PutObjectTaggingRequest putObjTagsRequest = new
PutObjectTaggingRequest()
                {
                    BucketName = bucketName,
                    Key = keyName,
                    Tagging = newTagSet
                };
                PutObjectTaggingResponse response2 = await
client.PutObjectTaggingAsync(putObjTagsRequest);

                // 4. Retrieve the object's tags.
```

```
                GetObjectTaggingRequest getTagsRequest2 = new
GetObjectTaggingRequest();
                getTagsRequest2.BucketName = bucketName;
                getTagsRequest2.Key = keyName;
                GetObjectTaggingResponse objectTags2 = await
client.GetObjectTaggingAsync(getTagsRequest2);
                for (int i = 0; i < objectTags2.Tagging.Count; i++)
                    Console.WriteLine("Key: {0}, Value: {1}",
objectTags2.Tagging[i].Key, objectTags2.Tagging[i].Value);

            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine(
                        "Error encountered ***. Message:'{0}' when writing an
object"
                        , e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine(
                    "Encountered an error. Message:'{0}' when writing an object"
                    , e.Message);
            }
        }
    }
}
```

Unduh dan unggah objek dengan presigned URLs

Anda dapat menggunakan presigned URLs untuk memberikan akses terbatas waktu ke objek di Amazon S3 tanpa memperbarui kebijakan bucket Anda. URL yang telah ditandatangani dapat dimasukkan di dalam browser atau dapat digunakan oleh program untuk mengunduh objek. Kredensi yang digunakan oleh URL presigned adalah milik AWS pengguna yang membuat URL.

Anda juga dapat menggunakan presigned URLs untuk mengizinkan seseorang mengunggah objek tertentu ke bucket Amazon S3 Anda. Ini memungkinkan unggahan tanpa mengharuskan pihak lain untuk memiliki kredensil atau izin AWS keamanan. Jika sebuah objek dengan kunci yang sama sudah ada di bucket seperti yang ditentukan dalam URL yang telah ditandatangani, Amazon S3 menggantikan objek yang sudah ada dengan objek yang diunggah.

Anda dapat menggunakan URL yang telah ditandatangani beberapa kali, hingga tanggal dan waktu kedaluwarsa.

Saat Anda membuat sebuah URL yang telah ditandatangani, Anda harus memberikan kredensial keamanan Anda, lalu menentukan berikut ini:

- Bucket Amazon S3
- Kunci objek (jika mengunduh objek ini akan ada di bucket Amazon S3 Anda, jika mengunggah ini adalah nama file yang akan diunggah)
- Metode HTTP (`GET`untuk mengunduh objek, `PUT` mengunggah, `HEAD` membaca metadata objek, dll)
- Interval waktu kedaluwarsa

Saat ini, Amazon S3 presigned URLs tidak mendukung penggunaan algoritma checksum integritas data berikut (CRC32,,CRC32C,SHA-1) saat Anda mengunggah objek. SHA-256 Untuk memverifikasi integritas objek Anda setelah mengunggah, Anda dapat memberikan MD5 {i>digest<i} objek saat Anda mengunggahnya dengan URL yang telah ditandatangani. Untuk informasi selengkapnya tentang integritas objek, lihat Memeriksa integritas objek di Amazon S3.

Topik

- Siapa yang dapat membuat URL yang telah ditandatangani
- Waktu kedaluwarsa untuk presigned URLs
- Pembatasan kemampuan URL yang telah ditandatangani
- Berbagi objek dengan presigned URLs
- Mengunggah objek dengan presigned URLs

Siapa yang dapat membuat URL yang telah ditandatangani

Siapa pun yang memiliki kredensial keamanan yang valid dapat membuat sebuah URL yang telah ditandatangani. Tetapi untuk seseorang berhasil mengakses objek, URL yang telah ditandatangani harus dibuat oleh seseorang yang memiliki izin untuk melakukan operasi yang menjadi dasar dari URL yang telah ditandatangani.

Berikut ini adalah jenis kredensial yang dapat Anda gunakan untuk membuat URL yang telah ditandatangani:

- Profil instans IAM–Valid hingga 6 jam.
- AWS Security Token Service–Berlaku hingga maksimum 36 jam saat ditandatangani dengan kredensial keamanan jangka panjang atau durasi kredensial sementara, mana yang berakhir terlebih dulu.
- Pengguna IAM - Berlaku hingga 7 hari saat Anda menggunakan AWS Signature Versi 4.

  Untuk membuat sebuah URL yang telah ditandatangani yang valid hingga 7 hari, pertama-tama delegasikan kredensial pengguna IAM (kunci akses dan kunci rahasia) ke metode yang Anda gunakan untuk membuat URL yang telah ditandatangani.

> **Note**
>
> Jika Anda membuat URL yang telah ditandatangani menggunakan kredensial sementara, URL tersebut akan kedaluwarsa saat kredensial-nya kedaluwarsa. Secara umum, URL yang telah ditetapkan sebelumnya akan kedaluwarsa saat kredensi yang Anda gunakan untuk membuatnya dicabut, dihapus, atau dinonaktifkan. Ini benar terjadi meskipun jika URL dibuat dengan waktu kedaluwarsa yang lebih lama. Untuk masa pakai kredensil keamanan sementara, lihat Membandingkan operasi AWS STS API di Panduan Pengguna IAM.

Waktu kedaluwarsa untuk presigned URLs

URL yang telah ditandatangani tetap valid untuk jangka waktu yang ditentukan saat URL dibuat. Jika Anda membuat URL yang telah ditandatangani dengan konsol Amazon S3, waktu kedaluwarsa dapat diatur antara 1 menit dan 12 jam. Jika Anda menggunakan AWS CLI atau AWS SDKs, waktu kedaluwarsa dapat diatur setinggi 7 hari.

Jika Anda membuat URL yang telah ditetapkan sebelumnya dengan menggunakan token sementara, URL akan kedaluwarsa saat token kedaluwarsa. Secara umum, URL yang telah ditetapkan sebelumnya akan kedaluwarsa saat kredensi yang Anda gunakan untuk membuatnya dicabut, dihapus, atau dinonaktifkan. Ini benar terjadi meskipun jika URL dibuat dengan waktu kedaluwarsa yang lebih lama. Untuk informasi selengkapnya tentang bagaimana kredensial yang Anda gunakan memengaruhi waktu kedaluwarsanya, lihat Siapa yang dapat membuat URL yang telah ditandatangani.

Amazon S3 memeriksa tanggal dan waktu kedaluwarsa URL yang ditandatangani pada saat permintaan HTTP. Misalnya, jika klien mulai mengunduh file besar segera sebelum waktu

kedaluwarsa, pengunduhan berlanjut meskipun waktu kedaluwarsa berlalu selama pengunduhan. Akan tetapi, jika koneksi menurun dan klien mencoba memulai ulang unduhan setelah waktu kedaluwarsa berlalu, pengunduhan gagal.

Pembatasan kemampuan URL yang telah ditandatangani

Kemampuan URL yang telah ditandatangani dibatasi oleh izin pengguna yang membuatnya. Intinya, presigned URLs adalah token pembawa yang memberikan akses kepada mereka yang memilikinya. Oleh karena itu, kami menyarankan agar Anda melindunginya sebagaimana mestinya. Berikut ini adalah beberapa metode yang dapat Anda gunakan untuk membatasi penggunaan URLs presigned Anda.

AWS Tanda Tangan Versi 4 (SiGv4)

Untuk menerapkan perilaku tertentu saat permintaan URL yang telah ditandatangani diautentikasi dengan menggunakan AWS Signature Version 4 (SigV4), Anda dapat menggunakan kunci kondisi dalam kebijakan bucket dan kebijakan titik akses. Contoh, kebijakan bucket berikut menggunakan `s3:signatureAge` kondisi untuk menolak permintaan URL yang telah ditandatangani Amazon S3 pada objek di *`amzn-s3-demo-bucket`* bucket jika tanda tangannya sudah lebih dari 10 menit. Untuk menggunakan contoh ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "Deny a presigned URL request if the signature is more than 10 min
 old",
         "Effect": "Deny",
         "Principal": {"AWS":"*"},
         "Action": "s3:*",
         "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
         "Condition": {
            "NumericGreaterThan": {
               "s3:signatureAge": 600000
            }
         }
      }
   ]
}
```

Untuk informasi selengkapnya tentang kunci kebijakan terkait AWS Tanda Tangan Versi 4, lihat Otentikasi AWS Tanda Tangan Versi 4 di Referensi API Amazon Simple Storage Service.

Pembatasan jalur jaringan

Jika Anda ingin membatasi penggunaan presigned URLs dan semua akses Amazon S3 ke jalur jaringan tertentu, Anda dapat AWS Identity and Access Management menulis kebijakan (IAM). Anda dapat mengatur kebijakan ini pada pengguna utama IAM yang melakukan panggilan, bucket Amazon S3, atau keduanya.

Pembatasan jalur jaringan pada pengguna utama IAM mengharuskan pengguna kredensial tersebut untuk membuat permintaan dari jaringan yang ditentukan. Pembatasan pada bucket atau titik akses mengharuskan semua permintaan ke sumber daya tersebut berasal dari jaringan tertentu. Pembatasan ini juga berlaku di luar skenario URL yang telah ditandatangani sebelumnya.

Kunci kondisi global IAM yang Anda gunakan bergantung pada jenis titik akhir. Jika Anda menggunakan titik akhir publik untuk Amazon S3, gunakan `aws:SourceIp`. Jika Anda menggunakan titik akhir cloud privat virtual (VPC) ke Amazon S3, gunakan `aws:SourceVpc` atau `aws:SourceVpce`.

Pernyataan kebijakan IAM berikut mengharuskan prinsipal untuk mengakses AWS hanya dari rentang jaringan yang ditentukan. Dengan pernyataan kebijakan ini, semua akses harus berasal dari rentang itu. Kebijakan ini juga termasuk kasus seseorang yang menggunakan URL yang telah ditandatangani sebelumnya untuk Amazon S3. Untuk menggunakan contoh ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
{
    "Sid": "NetworkRestrictionForIAMPrincipal",
    "Effect": "Deny",
    "Action": "*",
    "Resource": "*",
    "Condition": {
        "NotIpAddressIfExists": {"aws:SourceIp": "IP-address-range"},
        "BoolIfExists": {"aws:ViaAWSService": "false"}
    }
}
```

Berbagi objek dengan presigned URLs

Semua objek Amazon S3 secara default bersifat privat, hanya pemilik objek yang memiliki izin untuk mengaksesnya. Namun, pemilik objek dapat berbagi objek dengan orang lain dengan membuat

URL yang telah ditandatangani sebelumnya. URL yang telah ditetapkan sebelumnya menggunakan kredensial keamanan untuk memberikan izin terbatas waktu untuk mengunduh objek. URL dapat dimasukkan di dalam browser, atau digunakan oleh program untuk mengunduh objek. Kredensi yang digunakan oleh URL presigned adalah milik AWS pengguna yang membuat URL.

Untuk informasi umum tentang presigned URLs, lihatUnduh dan unggah objek dengan presigned URLs.

Anda dapat membuat URL yang telah ditetapkan sebelumnya untuk berbagi objek tanpa menulis kode apa pun dengan menggunakan konsol Amazon S3 AWS , Explorer for Visual Studio (Windows), atau. AWS Toolkit for Visual Studio Code Anda juga dapat membuat URL presigned secara terprogram dengan menggunakan AWS Command Line Interface (AWS CLI) atau. AWS SDKs

Menggunakan konsol S3

Anda dapat menggunakan konsol Amazon S3 untuk menghasilkan URL yang telah ditandatangani untuk berbagi objek dengan mengikuti langkah-langkah berikut. Saat menggunakan konsol, waktu kedaluwarsa maksimum untuk URL yang telah ditandatangani sebelumnya adalah 12 jam dari waktu pembuatan.

Untuk menghasilkan URL yang telah ditandatangani sebelumnya menggunakan konsol Amazon S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Dalam daftar Bucket tujuan umum, pilih nama bucket tujuan umum yang berisi objek yang Anda inginkan untuk URL yang telah ditetapkan sebelumnya.
4. Dalam daftar Objek, pilih objek yang ingin Anda buatkan URL yang telah ditandatangani.
5. Pada menu Tindakan objek, pilih Bagikan dengan URL yang telah ditandatangani.
6. Tentukan berapa lama Anda menginginkan URL yang telah ditandatangani tersebut valid.
7. Pilih Buat URL yang telah ditandatangani.
8. Ketika konfirmasi muncul, URL secara otomatis disalin ke clipboard Anda. Anda akan melihat tombol untuk menyalin URL yang telah ditandatangani jika Anda perlu menyalinnya lagi.

Menggunakan AWS CLI

AWS CLI Perintah contoh berikut menghasilkan URL yang telah ditetapkan sebelumnya untuk berbagi objek dari bucket Amazon S3. Saat Anda menggunakan AWS CLI, waktu kedaluwarsa

maksimum untuk URL yang telah ditetapkan sebelumnya adalah 7 hari dari waktu pembuatan. Untuk menggunakan contoh ini, ganti *user input placeholders*dengan informasi Anda sendiri.

```
aws s3 presign s3://amzn-s3-demo-bucket/mydoc.txt --expires-in 604800
```

> **Note**
>
> Untuk semua yang Wilayah AWS diluncurkan setelah 20 Maret 2019 Anda perlu menentukan `endpoint-url` dan `Wilayah AWS` dengan permintaan. Untuk daftar semua Wilayah dan titik akhir Amazon S3, lihat Wilayah dan Titik Akhir dalam AWS Referensi Umum.

```
aws s3 presign s3://amzn-s3-demo-bucket/mydoc.txt --expires-in 604800 --region af-south-1 --endpoint-url https://s3.af-south-1.amazonaws.com
```

Untuk informasi selengkapnya, lihat presign dalam AWS CLI Referensi Perintah.

Menggunakan AWS SDKs

Anda dapat membuat URL presigned secara terprogram dengan menggunakan. AWS SDKs

Python

Script Python berikut menghasilkan URL `GET` presigned untuk berbagi objek.

1. Salin isi skrip dan simpan sebagai file "*get-only-url.py*". Untuk menggunakan contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri (seperti nama file Anda).

```
import argparse
import boto3
from botocore.exceptions import ClientError

def generate_presigned_url(s3_client, client_method, method_parameters,
 expires_in):
    """
    Generate a presigned Amazon S3 URL that can be used to perform an action.
```

```
    :param s3_client: A Boto3 Amazon S3 client.
    :param client_method: The name of the client method that the URL performs.
    :param method_parameters: The parameters of the specified client method.
    :param expires_in: The number of seconds the presigned URL is valid for.
    :return: The presigned URL.
    """
    try:
        url = s3_client.generate_presigned_url(
            ClientMethod=client_method,
            Params=method_parameters,
            ExpiresIn=expires_in
        )
    except ClientError:
        print(f"Couldn't get a presigned URL for client method
 '{client_method}'.")
        raise
    return url

def main():
    parser = argparse.ArgumentParser()
    parser.add_argument("bucket", help="The name of the bucket.")
    parser.add_argument(
        "key", help="The key (path and filename) in the S3 bucket.",
    )
    args = parser.parse_args()

    # By default, this will use credentials from ~/.aws/credentials
    s3_client = boto3.client("s3")

    # The presigned URL is specified to expire in 1000 seconds
    url = generate_presigned_url(
        s3_client,
        "get_object",
        {"Bucket": args.bucket, "Key": args.key},
        1000
    )
    print(f"Generated GET presigned URL: {url}")

if __name__ == "__main__":
    main()
```

2. Untuk menghasilkan URL yang GET telah ditetapkan sebelumnya untuk berbagi file, jalankan skrip berikut dengan nama bucket dan jalur objek yang diinginkan.

Perintah berikut menggunakan nilai contoh. Ganti *user input placeholders* dengan informasi Anda sendiri.

```
python get-only-url.py amzn-s3-demo-bucket <object-path>
```

Skrip akan menampilkan URL yang GET telah ditentukan sebelumnya:

```
Generated GET presigned URL: https://amzn-s3-demo-
bucket.s3.amazonaws.com/object.txt?AWS
 AccessKeyId=AKIAIOSFODNN7EXAMPLE&Signature=vjbyNxybdZaMmLa
%2ByT372YEAiv4%3D&Expires=1741978496
```

3. Anda dapat mengunduh file menggunakan URL presigned yang dihasilkan dengan curl:

```
curl -X GET "generated-presigned-url" -o "path/to/save/file"
```

Untuk lebih banyak contoh penggunaan AWS SDKs untuk menghasilkan URL yang telah ditetapkan sebelumnya untuk berbagi objek, lihat Membuat URL yang telah ditetapkan sebelumnya untuk Amazon S3 menggunakan SDK. AWS

> **Note**
>
> Untuk semua yang Wilayah AWS diluncurkan setelah 20 Maret 2019 Anda perlu menentukan `endpoint-url` dan `Wilayah AWS` dengan permintaan. Untuk daftar semua Wilayah dan titik akhir Amazon S3, lihat Wilayah dan Titik Akhir dalam AWS Referensi Umum.

> **Note**
>
> Saat menggunakan AWS SDKs, atribut Tagging harus berupa header dan bukan parameter kueri. Semua atribut lainnya dapat diteruskan sebagai parameter untuk URL presigned.

Menggunakan AWS Toolkit for Visual Studio (Windows)

> **Note**
>
> Saat ini, AWS Toolkit for Visual Studio tidak mendukung Visual Studio untuk Mac.

1. Instal AWS Toolkit for Visual Studio menggunakan petunjuk berikut, Menginstal dan menyiapkan Toolkit for Visual Studio di AWS Toolkit for Visual Studio Panduan Pengguna.
2. Connect untuk AWS menggunakan langkah-langkah berikut, Connecting to AWS di Panduan AWS Toolkit for Visual Studio Pengguna.
3. Di panel sisi kiri berlabel AWS Explorer, klik dua kali bucket yang berisi objek Anda.
4. Klik kanan objek yang ingin Anda buat URL yang telah ditetapkan sebelumnya dan pilih Buat URL Pra-Tanda Tangan... .
5. Di jendela pop-up, atur tanggal kedaluwarsa dan waktu untuk URL yang telah ditetapkan sebelumnya.
6. Object Key, harus diisi terlebih dahulu berdasarkan objek yang Anda pilih.
7. Pilih GET untuk menentukan bahwa URL yang telah ditandatangani sebelumnya ini akan digunakan untuk mengunduh objek.
8. Pilih tombol Hasilkan.
9. Untuk menyalin URL ke clipboard, pilih Salin.
10. Untuk menggunakan URL presigned yang dihasilkan, rekatkan URL ke browser apa pun.

Menggunakan AWS Toolkit for Visual Studio Code

Jika Anda menggunakan Visual Studio Code, Anda dapat membuat URL yang telah ditandatangani sebelumnya untuk membagikan objek tanpa menulis kode apa pun dengan menggunakan AWS Toolkit for Visual Studio Code. Untuk informasi umum, lihat AWS Toolkit for Visual Studio Code dalam Panduan Pengguna AWS Toolkit for Visual Studio Code .

Untuk petunjuk tentang cara menginstal AWS Toolkit for Visual Studio Code, lihat Menginstal AWS Toolkit for Visual Studio Code di Panduan AWS Toolkit for Visual Studio Code Pengguna.

1. Connect untuk AWS menggunakan langkah-langkah berikut, Connecting to AWS Toolkit for Visual Studio Code di Panduan AWS Toolkit for Visual Studio Code Pengguna.
2. Pilih AWS logo di panel kiri di Visual Studio Code.

3. Di bawah EXPLORER, pilih S3.
4. Pilih bucket dan filenya, lalu buka menu konteks (klik kanan).
5. Pilih Hasilkan URL yang telah ditandatangani sebelumnya, lalu atur waktu kedaluwarsa (dalam hitungan menit).
6. Tekan Enter, dan URL yang telah ditandatangani sebelumnya akan disalin ke clipboard Anda.

Mengunggah objek dengan presigned URLs

Anda dapat menggunakan presigned URLs untuk mengizinkan seseorang mengunggah objek ke bucket Amazon S3 Anda. Menggunakan URL yang telah ditetapkan sebelumnya akan memungkinkan unggahan tanpa mengharuskan pihak lain memiliki kredensyal atau izin AWS keamanan. URL yang telah ditandatangani sebelumnya dibatasi oleh izin pengguna yang membuatnya. Artinya, jika Anda menerima URL yang telah ditandatangani sebelumnya untuk mengunggah suatu objek, Anda dapat mengunggah objek hanya jika pembuat URL tersebut memiliki izin yang diperlukan untuk mengunggah objek tersebut.

Saat seseorang menggunakan URL untuk mengunggah sebuah objek, Amazon S3 membuat objek dalam bucket yang ditentukan. Jika objek dengan kunci yang sama yang ditentukan dalam URL yang telah ditandatangani sebelumnya sudah ada di bucket, Amazon S3 akan menggantikan objek yang ada dengan objek yang diunggah. Setelah diunggah, pemilik bucket akan memiliki objek tersebut.

Untuk informasi umum tentang presigned URLs, lihatUnduh dan unggah objek dengan presigned URLs.

Anda dapat membuat URL yang telah ditandatangani sebelumnya untuk mengunggah objek tanpa menulis kode apa pun dengan menggunakan AWS Explorer for Visual Studio. Anda juga dapat membuat URL presigned secara terprogram dengan menggunakan. AWS SDKs

> **Note**
>
> Saat ini, AWS Toolkit for Visual Studio tidak mendukung Visual Studio untuk Mac.

Menggunakan AWS Toolkit for Visual Studio (Windows)

1. Instal AWS Toolkit for Visual Studio menggunakan petunjuk berikut, Menginstal dan menyiapkan Toolkit for Visual Studio di AWS Toolkit for Visual Studio Panduan Pengguna.

2. Connect untuk AWS menggunakan langkah-langkah berikut, Connecting to AWS di Panduan AWS Toolkit for Visual Studio Pengguna.
3. Di panel sisi kiri berlabel AWS Explorer, klik kanan bucket yang ingin Anda unggah objek.
4. Pilih Buat URL Pra-Tanda Tangan... .
5. Di jendela pop-up, atur tanggal kedaluwarsa dan waktu untuk URL yang telah ditetapkan sebelumnya.
6. Untuk Object Key, atur nama file yang akan diunggah. File yang Anda unggah harus sama persis dengan nama ini. Jika objek dengan kunci objek yang sama sudah ada di bucket, Amazon S3 akan mengganti objek yang ada dengan objek yang baru diunggah.
7. Pilih PUT untuk menentukan bahwa URL yang telah ditandatangani sebelumnya ini akan digunakan untuk mengunggah objek.
8. Pilih tombol Hasilkan.
9. Untuk menyalin URL ke clipboard, pilih Salin.
10. Untuk menggunakan URL ini, Anda dapat mengirim permintaan PUT dengan perintah `curl`. Sertakan path lengkap ke file Anda dan URL presigned itu sendiri.

```
curl -X PUT -T "/path/to/file" "presigned URL"
```

Menggunakan AWS SDKs untuk menghasilkan URL yang **PUT** telah ditetapkan sebelumnya untuk mengunggah file

Anda dapat membuat URL presigned yang dapat melakukan tindakan S3 untuk waktu yang terbatas.

> **Note**
>
> Jika Anda menggunakan AWS CLI atau AWS SDKs, waktu kedaluwarsa untuk presigned URLs dapat diatur setinggi 7 hari. Untuk informasi selengkapnya, lihat Waktu kedaluwarsa untuk presigned. URLs

Python

Skrip Python berikut menghasilkan URL yang PUT telah ditetapkan sebelumnya untuk mengunggah objek ke bucket tujuan umum S3.

1. Salin isi skrip dan simpan sebagai file "*put-only-url.py*". Untuk menggunakan contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri (seperti nama file Anda).

```
import argparse
import boto3
from botocore.exceptions import ClientError

def generate_presigned_url(s3_client, client_method, method_parameters,
 expires_in):
    """
    Generate a presigned Amazon S3 URL that can be used to perform an action.

    :param s3_client: A Boto3 Amazon S3 client.
    :param client_method: The name of the client method that the URL performs.
    :param method_parameters: The parameters of the specified client method.
    :param expires_in: The number of seconds the presigned URL is valid for.
    :return: The presigned URL.
    """
    try:
        url = s3_client.generate_presigned_url(
            ClientMethod=client_method,
            Params=method_parameters,
            ExpiresIn=expires_in
        )
    except ClientError:
        print(f"Couldn't get a presigned URL for client method
 '{client_method}'.")
        raise
    return url

def main():
    parser = argparse.ArgumentParser()
    parser.add_argument("bucket", help="The name of the bucket.")
    parser.add_argument(
        "key", help="The key (path and filename) in the S3 bucket.",
    )
    args = parser.parse_args()

    # By default, this will use credentials from ~/.aws/credentials
    s3_client = boto3.client("s3")

    # The presigned URL is specified to expire in 1000 seconds
```

```
    url = generate_presigned_url(
        s3_client,
        "put_object",
        {"Bucket": args.bucket, "Key": args.key},
        1000
    )
    print(f"Generated PUT presigned URL: {url}")

if __name__ == "__main__":
    main()
```

2. Untuk menghasilkan URL yang PUT telah ditetapkan sebelumnya untuk mengunggah file, jalankan skrip berikut dengan nama bucket dan jalur objek yang diinginkan.

   Perintah berikut menggunakan nilai contoh. Ganti *user input placeholders* dengan informasi Anda sendiri.

   ```
   python put-only-url.py amzn-s3-demo-bucket <object-path>
   ```

   Skrip akan menampilkan URL yang PUT telah ditentukan sebelumnya:

   ```
   Generated PUT presigned URL: https://amzn-s3-demo-
   bucket.s3.amazonaws.com/object.txt?AWS
    AccessKeyId=AKIAIOSFODNN7EXAMPLE&Signature=vjbyNxybdZaMmLa
   %2ByT372YEAiv4%3D&Expires=1741978496
   ```

3. Anda sekarang dapat mengunggah file menggunakan URL presigned yang dihasilkan dengan curl:

   ```
   curl -X PUT -T "path/to/your/local/file" "generated-presigned-url"
   ```

Untuk lebih banyak contoh penggunaan AWS SDKs untuk menghasilkan URL yang telah ditetapkan sebelumnya untuk mengunggah objek, lihat Membuat URL yang telah ditetapkan sebelumnya untuk Amazon S3 menggunakan SDK. AWS

Mengubah objek dengan S3 Lambda Objek

Dengan Lambda Objek Amazon S3, Anda dapat menambahkan kode Anda sendiri ke `GET`, `LIST`, dan `HEAD` Amazon S3 dan meminta untuk mengubah dan memproses data saat dikembalikan

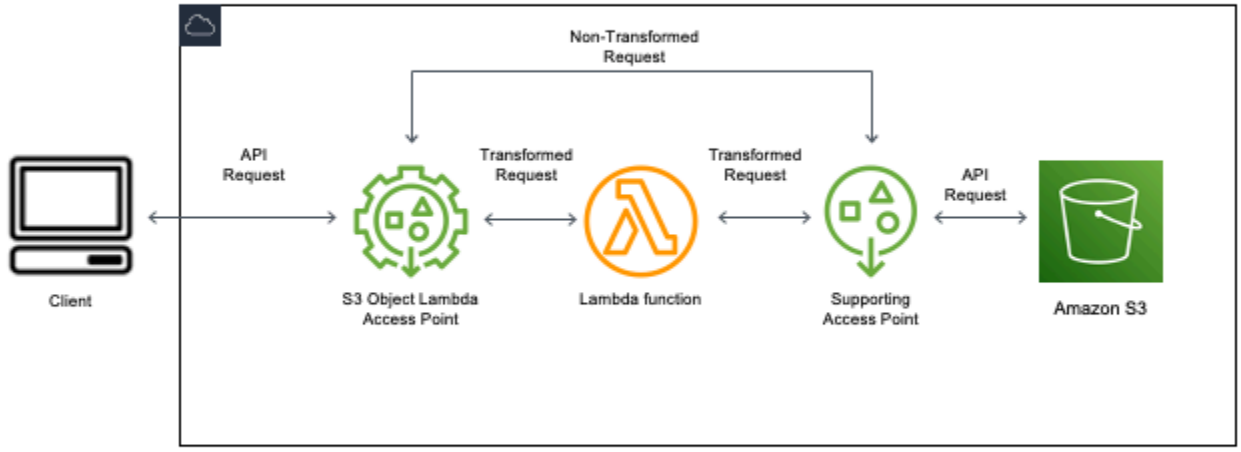
ke aplikasi. Anda dapat menggunakan kode khusus untuk memodifikasi data yang dikembalikan oleh permintaan GET S3 untuk melakukan filter baris, mengubah ukuran dan watermark gambar secara dinamis, menyamarkan data rahasia, dan banyak lagi. Anda juga dapat menggunakan S3 Lambda Objek untuk memodifikasi output permintaan LIST S3 untuk membuat tampilan kustom dari semua objek dalam bucket dan permintaan HEAD S3 untuk memodifikasi metadata objek, seperti nama objek dan ukuran. Anda dapat menggunakan S3 Object Lambda sebagai asal distribusi CloudFront Amazon Anda untuk menyesuaikan data bagi pengguna akhir, seperti mengubah ukuran gambar secara otomatis, mentranskode format lama (seperti dari JPEG ke WebP), atau menghapus metadata. Untuk informasi selengkapnya, lihat posting AWS Blog Menggunakan Amazon S3 Object Lambda dengan Amazon. CloudFront Didukung oleh fungsi AWS Lambda, kode Anda berjalan pada infrastruktur yang sepenuhnya dikelola oleh. AWS Menggunakan S3 Lambda Objek mengurangi kebutuhan untuk membuat dan menyimpan salinan turunan data Anda atau menjalankan proksi, semuanya tanpa perlu mengubah aplikasi Anda.

Bagaimana S3 Lambda Objek bekerja

S3 Object Lambda AWS Lambda menggunakan fungsi untuk secara otomatis memproses output standar GET S3LIST,, atau permintaan. HEAD AWS Lambda adalah layanan komputasi tanpa server yang menjalankan kode yang ditentukan pelanggan tanpa memerlukan pengelolaan sumber daya komputasi yang mendasarinya. Anda dapat melakukan otorisasi dan menjalankan fungsi Lambda kustom Anda sendiri, menyesuaikan transformasi data untuk kasus penggunaan tertentu Anda.

Setelah Anda mengonfigurasi fungsi Lambda, Anda melampirkannya ke S3 Lambda Objek titik akhir layanan, yang dikenal sebagai Titik Akses Objek Lambda. Object Lambda Access Point menggunakan titik akses S3 standar, yang dikenal sebagai titik akses pendukung, untuk mengakses data.

Saat Anda mengirim permintaan ke Titik Akses Lambda Objek, Amazon S3 secara otomatis memanggil fungsi Lambda Anda. Data yang diambil dengan menggunakan permintaan S3 GET, LIST, atau HEAD melalui Titik Akses Lambda Objek mengembalikan hasil berubah kembali ke aplikasi. Semua permintaan lainnya diproses seperti biasa, seperti yang digambarkan dalam diagram berikut.



Topik di bagian ini menjelaskan cara bekerja dengan S3 Lambda Objek.

Topik

- Membuat Titik Akses Objek Lambda
- Menggunakan Titik Akses Lambda Objek Amazon S3
- Pertimbangan keamanan untuk Titik Akses S3 Lambda Objek
- Menulis fungsi Lambda untuk Titik Akses S3 Lambda Objek
- Menggunakan fungsi Lambda yang AWS dibangun
- Praktik terbaik dan pedoman untuk S3 Lambda Objek
- Tutorial S3 Lambda Objek
- Debugging dan pemecahan masalah S3 Object Lambda

Membuat Titik Akses Objek Lambda

Titik Akses Objek Lambda dikaitkan dengan tepat satu titik akses standar, yang Anda tentukan selama pembuatan. Untuk membuat titik akses Lambda Objek, Anda memerlukan sumber daya berikut:

- Titik akses standar S3. Saat Anda bekerja dengan Object Lambda Access Points, titik akses standar ini dikenal sebagai titik akses pendukung dan dilampirkan ke bucket S3 atau Amazon FSx untuk volume OpenZFS. Untuk informasi selengkapnya tentang cara membuat titik akses standar, lihat the section called “Membuat titik akses”.
- Sebuah AWS Lambda fungsi. Anda dapat membuat fungsi Lambda Anda sendiri, atau Anda juga dapat menggunakan fungsi bawaan. Untuk informasi selengkapnya tentang membuat fungsi Lambda, lihat the section called “Menulis fungsi Lambda”. Untuk informasi lebih lanjut tentang fungsi bawaan, lihat Menggunakan fungsi Lambda yang AWS dibangun.
- (Opsional) Kebijakan AWS Identity and Access Management (IAM). Titik akses Amazon S3 mendukung kebijakan sumber daya IAM yang dapat Anda gunakan untuk mengontrol penggunaan titik akses berdasarkan sumber daya, pengguna, atau ketentuan lainnya. Untuk informasi selengkapnya tentang pembuatan kebijakan ini, lihat the section called “Mengonfigurasi kebijakan IAM”.

Bagian berikut menjelaskan cara membuat Titik Akses Lambda Objek dengan menggunakan:

- The AWS Management Console
- AWS Command Line Interface (AWS CLI)
- AWS CloudFormation Template
- The AWS Cloud Development Kit (AWS CDK)

Untuk informasi lebih lanjut tentang cara membuat Titik Akses Lambda Objek menggunakan API REST, lihat `CreateAccessPointForObjectLambda` dalam Referensi API Amazon Simple Storage Service.

Membuat Titik Akses Lambda Object

Gunakan salah satu dari prosedur berikut ini untuk membuat Titik Akses Lambda Object Anda.

Menggunakan konsol S3

Untuk membuat Titik Akses Lambda Objek menggunakan konsol

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah yang ingin Anda alihkan.

3. Di panel navigasi sebelah kiri, pilih Titik Akses Lambda Objek.
4. Pada halaman Titik Akses Lambda Objek, pilih Buat Titik Akses Lambda Objek.
5. Untuk Nama Titik Akses Lambda Objek, masukkan nama yang ingin Anda gunakan untuk titik aksesnya.

   Seperti halnya titik akses standar, ada aturan untuk penamaan Titik Akses Lambda Objek. Untuk informasi selengkapnya, lihat Aturan penamaan untuk titik akses.
6. Untuk Titik Akses Dukungan, masukkan atau telusuri titik akses standar yang ingin Anda gunakan. Titik akses harus Wilayah AWS sama dengan objek yang ingin Anda ubah. Untuk informasi selengkapnya tentang cara membuat titik akses standar, lihat the section called "Membuat titik akses".
7. Di bawah Konfigurasi transformasi, Anda dapat menambahkan fungsi yang mengubah data untuk Titik Akses Lambda Objek Anda. Lakukan salah satu tindakan berikut:

   - Jika Anda sudah memiliki AWS Lambda fungsi di akun Anda, Anda dapat memilihnya di bawah fungsi Invoke Lambda. Di sini Anda dapat memasukkan Nama Sumber Daya Amazon (ARN) dari fungsi Lambda di Anda atau Akun AWS memilih fungsi Lambda dari menu tarik-turun.
   - Jika Anda ingin menggunakan fungsi yang AWS dibangun pilih nama fungsi di bawah fungsi yang AWS dibangun dan pilih Buat fungsi Lambda. Ini akan membawa Anda ke konsol Lambda di mana Anda dapat menerapkan fungsi bawaan ke dalam konsol Anda. Akun AWS Untuk informasi lebih lanjut tentang fungsi bawaan, lihat Menggunakan fungsi Lambda yang AWS dibangun.

   Di bawah S3 APIs, pilih satu atau beberapa operasi API untuk dipanggil. Untuk setiap API yang dipilih, Anda harus menentukan fungsi Lambda yang akan dipanggil.
8. (Opsional) Di bawah Muatan, tambahkan teks JSON yang ingin Anda berikan ke fungsi Lambda Anda sebagai input. Anda dapat mengonfigurasi muatan dengan parameter berbeda untuk Titik Akses Lambda Objek berbeda yang menginvokasi fungsi Lambda yang sama, sehingga memperluas fleksibilitas fungsi Lambda Anda.

   > **Important**
   >
   > Saat menggunakan Titik Akses Lambda Objek, pastikan muatan tidak berisi informasi rahasia apa pun.

9. (Opsional) Untuk Rentang dan nomor bagian, Anda harus mengaktifkan opsi ini jika Anda ingin memproses permintaan `GET` dan `HEAD` dengan header rentang dan nomor bagian. Mengaktifkan opsi ini mengonfirmasi bahwa fungsi Lambda Anda dapat mengenali dan memproses permintaan ini. Untuk informasi lebih lanjut tentang berbagai header dan nomor bagian, lihat Bekerja dengan header Range dan partNumber.
10. (Opsional) Untuk Metrik permintaan, pilih Aktifkan atau Nonaktifkan untuk menambahkan pemantauan Amazon S3 ke Titik Akses Lambda Objek Anda. Metrik permintaan ditagih dengan tarif Amazon CloudWatch standar.
11. (Opsional) Di bawah Kebijakan Titik Akses Lambda, buat pengaturan kebijakan sumber daya. Kebijakan sumber daya memberikan izin untuk Titik Akses Lambda Objek yang ditentukan dan dapat mengontrol penggunaan titik akses berdasarkan sumber daya, pengguna, atau ketentuan lainnya. Untuk informasi selengkapnya tentang kebijakan sumber daya Titik Akses Lambda Object, lihat, Mengonfigurasi kebijakan IAM untuk Titik Akses Lambda Objek.
12. Di bagian bawah Pengaturan Blok Titik Akses untuk Titik Akses Lambda Objek ini, pilih pengaturan blokir akses publik yang ingin Anda terapkan. Semua pengaturan blokir akses publik diaktifkan secara default untuk Titik Akses Lambda Objek baru, dan kami menyarankan Anda membiarkan pengaturan default diaktifkan. Amazon S3 saat ini tidak mendukung perubahan pengaturan blokir akses publik milik Titik Akses Lambda Objek setelah Titik Akses Lambda Objek dibuat.

    Untuk informasi selengkapnya tentang menggunakan Blokir Akses Publik Amazon S3., lihat Mengelola akses publik ke titik akses untuk ember tujuan umum.
13. Pilih Buat Titik Akses Objek Lambda.

Menggunakan AWS CLI

Untuk membuat Object Lambda Access Point dengan menggunakan template AWS CloudFormation

> **Note**
>
> Untuk menggunakan perintah berikut ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

1. Unduh paket penerapan AWS Lambda fungsi `s3objectlambda_deployment_package.zip` di konfigurasi default Objek Lambda S3.

2. Jalankan perintah `put-object` berikut untuk mengunggah paket ke bucket Amazon S3.

```
aws s3api put-object --bucket Amazon S3 bucket name --key
 s3objectlambda_deployment_package.zip --body release/
s3objectlambda_deployment_package.zip
```

3. Unduh AWS CloudFormation template `s3objectlambda_defaultconfig.yaml` di konfigurasi default Objek Lambda S3.

4. Jalankan perintah `deploy` berikut untuk menyebarkan templat ke Akun AWS Anda.

```
aws cloudformation deploy --template-file s3objectlambda_defaultconfig.yaml \
 --stack-name AWS CloudFormation stack name \
 --parameter-overrides ObjectLambdaAccessPointName=Object Lambda Access Point name
 \
  SupportingAccessPointName=Amazon S3 access point S3BucketName=Amazon S3 bucket \
  LambdaFunctionS3BucketName=Amazon S3 bucket containing your Lambda package \
  LambdaFunctionS3Key=Lambda object key LambdaFunctionS3ObjectVersion=Lambda object
 version \
  LambdaFunctionRuntime=Lambda function runtime --capabilities capability_IAM
```

Anda dapat mengonfigurasi AWS CloudFormation template ini untuk memanggil Lambda `GET` untuk`HEAD`,, `LIST` dan operasi API. Untuk informasi selengkapnya tentang mengubah konfigurasi default templat, lihat the section called “Otomatiskan pengaturan Lambda Objek S3 dengan AWS CloudFormation”.

Untuk membuat Object Lambda Access Point dengan menggunakan AWS CLI

> **Note**
>
> Untuk menggunakan perintah berikut ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

Contoh berikut membuat Titik Akses Lambda Objek yang diberi nama *`my-object-lambda-ap`* untuk bucket *`amzn-s3-demo-bucket1`* di akun *`111122223333`*. Contoh ini mengasumsikan bahwa titik akses standar bernama *`example-ap`* telah dibuat. Untuk informasi selengkapnya tentang cara membuat titik akses standar, lihat the section called “Membuat titik akses”.

Contoh ini menggunakan fungsi AWS `decompress` prebuilt. Untuk informasi lebih lanjut tentang fungsi bawaan, lihat the section called “Menggunakan fungsi AWS yang dibangun”.

1. Buat bucket. Dalam contoh ini, kami akan menggunakan *`amzn-s3-demo-bucket1`*. Untuk informasi tentang membuat bucket, lihat the section called “Membuat ember tujuan umum”.
2. Buat titik akses standar, dan lampirkan ke bucket Anda. Dalam contoh ini, kami akan menggunakan *`example-ap`*. Untuk informasi selengkapnya tentang cara membuat titik akses standar, lihat the section called “Membuat titik akses”.
3. Lakukan salah satu hal berikut ini:

   - Buat fungsi Lambda di akun Anda yang ingin Anda gunakan untuk mengubah objek Amazon S3 Anda. Untuk informasi selengkapnya tentang membuat fungsi Lambda, lihat the section called “Menulis fungsi Lambda”. Untuk menggunakan fungsi kustom Anda dengan AWS CLI, lihat Menggunakan Lambda dengan AWS CLI di Panduan AWS Lambda Pengembang.
   - Gunakan fungsi Lambda AWS bawaan. Untuk informasi lebih lanjut tentang fungsi bawaan, lihat Menggunakan fungsi Lambda yang AWS dibangun.
4. Buat file konfigurasi JSON bernama `my-olap-configuration.json`. Dalam konfigurasi ini, sediakan titik akses pendukung dan Amazon Resource Name (ARN) untuk fungsi Lambda yang Anda buat pada langkah sebelumnya atau ARN untuk fungsi bawaan yang Anda gunakan.

   Example

   ```
   {
       "SupportingAccessPoint" : "arn:aws:s3:us-
   east-1:111122223333:accesspoint/example-ap",
       "TransformationConfigurations": [{
           "Actions" : ["GetObject", "HeadObject", "ListObjects", "ListObjectsV2"],
           "ContentTransformation" : {
               "AwsLambda": {
                   "FunctionPayload" : "{\"compressionType\":\"gzip\"}",
                   "FunctionArn" : "arn:aws:lambda:us-east-1:111122223333:function/
   compress"
               }
           }
       }]
   }
   ```
5. Jalankan perintah `create-access-point-for-object-lambda` untuk membuat Titik Akses Lambda Objek Anda.

```
aws s3control create-access-point-for-object-lambda --account-id 111122223333 --
name my-object-lambda-ap --configuration file://my-olap-configuration.json
```

6. (Opsional) Buat file kebijakan JSON bernama `my-olap-policy.json`.

   Menambahkan kebijakan sumber daya Titik Akses Lambda Objek dapat mengontrol penggunaan titik akses berdasarkan sumber daya, pengguna, atau kondisi lainnya. Kebijakan sumber daya ini memberikan izin akun `GetObject` *444455556666* ke Titik Akses Lambda Object yang telah ditentukan.

   Example

```
{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Sid": "Grant account 444455556666 GetObject access",
            "Effect": "Allow",
            "Action": "s3-object-lambda:GetObject",
            "Principal": {
                "AWS": "arn:aws:iam::444455556666:root"
            },
            "Resource": "your-object-lambda-access-point-arn"
        }
    ]
}
```

7. (Opsional) Jalankan perintah `put-access-point-policy-for-object-lambda` untuk mengatur kebijakan sumber daya Anda.

```
aws s3control put-access-point-policy-for-object-lambda --account-id 111122223333
 --name my-object-lambda-ap --policy file://my-olap-policy.json
```

8. (Opsional) Tentukan muatan.

   Payload adalah JSON opsional yang dapat Anda berikan ke AWS Lambda fungsi Anda sebagai input. Anda dapat mengonfigurasi muatan dengan parameter berbeda untuk Titik Akses Lambda Objek berbeda yang menginvokasi fungsi Lambda yang sama, sehingga memperluas fleksibilitas fungsi Lambda Anda.

Konfigurasi titik akses Lambda Objek berikut menunjukkan muatan dengan dua parameter.

```
{
 "SupportingAccessPoint": "AccessPointArn",
 "CloudWatchMetricsEnabled": false,
 "TransformationConfigurations": [{
  "Actions": ["GetObject", "HeadObject", "ListObjects", "ListObjectsV2"],
  "ContentTransformation": {
   "AwsLambda": {
    "FunctionArn": "FunctionArn",
    "FunctionPayload": "{\"res-x\": \"100\",\"res-y\": \"100\"}"
   }
  }
 }]
}
```

Konfigurasi Titik Akses Lambda Objek berikut menunjukkan payload dengan satu parameter, dan dengan `GetObject-Range`, `GetObject-PartNumber`, `HeadObject-Range`, dan `HeadObject-PartNumber` diaktifkan.

```
{
    "SupportingAccessPoint":"AccessPointArn",
    "CloudWatchMetricsEnabled": false,
    "AllowedFeatures": ["GetObject-Range", "GetObject-PartNumber", "HeadObject-
Range", "HeadObject-PartNumber"],
    "TransformationConfigurations": [{
        "Action": ["GetObject", "HeadObject", "ListObjects", "ListObjectsV2"],
        "ContentTransformation": {
            "AwsLambda": {
                "FunctionArn":"FunctionArn",
                "FunctionPayload": "{\"compression-amount\": \"5\"}"
            }
        }
    }]
}
```

> **Important**
>
> Saat menggunakan Titik Akses Lambda Objek, pastikan muatan tidak berisi informasi rahasia apa pun.

Menggunakan AWS CloudFormation konsol dan template

Anda dapat membuat Titik Akses Lambda Objek dengan menggunakan konfigurasi default yang disediakan oleh Amazon S3. Anda dapat mengunduh AWS CloudFormation template dan kode sumber fungsi Lambda dari GitHub repositori dan menyebarkan sumber daya ini untuk menyiapkan Objek Lambda Access Point fungsional.

Untuk informasi tentang memodifikasi konfigurasi default AWS CloudFormation template, lihatthe section called “Otomatiskan pengaturan Lambda Objek S3 dengan AWS CloudFormation”.

Untuk informasi tentang mengonfigurasi Titik Akses Objek Lambda dengan AWS CloudFormation menggunakan tanpa templat, *`AWS::S3ObjectLambda::AccessPoint`*lihat di Panduan Pengguna AWS CloudFormation .

Untuk mengunggah paket deployment fungsi Lambda

1. Unduh paket penerapan AWS Lambda fungsi `s3objectlambda_deployment_package.zip` di konfigurasi default Objek Lambda S3.
2. Unggah paket ke bucket Amazon S3.

Untuk membuat Object Lambda Access Point dengan menggunakan konsol AWS CloudFormation

1. Unduh AWS CloudFormation template `s3objectlambda_defaultconfig.yaml` di konfigurasi default Objek Lambda S3.
2. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di https://console.aws.amazon.com/cloudformation.
3. Lakukan salah satu tindakan berikut:
   - Jika Anda belum pernah menggunakan AWS CloudFormation sebelumnya, di AWS CloudFormation beranda, pilih Buat tumpukan.
   - Jika Anda pernah menggunakan AWS CloudFormation sebelumnya, di panel navigasi kiri, pilih Tumpukan. Pilih Buat tumpukan, lalu pilih Dengan sumber daya baru (standar).
4. Untuk Prasyarat-Siapkan templat, pilih Templat sudah siap.
5. Untuk Tentukan templat, pilih Unggah file templat dan unggah`s3objectlambda_defaultconfig.yaml`.
6. Pilih Berikutnya.
7. Pada halaman Tentukan detail tumpukan, masukkan nama untuk tumpukan.

8. Di bagian Parameter, tentukan parameter berikut, yang ditentukan di dalam templat tumpukan:

   a. Untuk CreateNewSupportingAccessPoint, lakukan salah satu hal berikut:

      - Jika Anda sudah memiliki titik akses pendukung untuk bucket S3 tempat Anda mengunggah templat, pilih false.
      - Jika Anda ingin membuat titik akses baru untuk bucket ini, pilih true.

   b. Untuk EnableCloudWatchMonitoring, pilih true atau false, tergantung apakah Anda ingin mengaktifkan metrik dan alarm CloudWatch permintaan Amazon.

   c. (Opsional) Untuk LambdaFunctionPayload, tambahkan teks JSON yang ingin Anda berikan ke fungsi Lambda Anda sebagai input. Anda dapat mengonfigurasi muatan dengan parameter berbeda untuk Titik Akses Lambda Objek berbeda yang menginvokasi fungsi Lambda yang sama, sehingga memperluas fleksibilitas fungsi Lambda Anda.

      > **Important**
      >
      > Saat menggunakan Titik Akses Lambda Objek, pastikan muatan tidak berisi informasi rahasia apa pun.

   d. Untuk LambdaFunctionRuntime, masukkan runtime pilihan Anda untuk fungsi Lambda. Pilihan yang tersedia adalah `nodejs14.x`, `python3.9`, `java11`.

   e. Untuk LambdaFunctionS3 BucketName, masukkan nama bucket Amazon S3 tempat Anda mengunggah paket penerapan.

   f. Untuk LambdaFunctionS3Key, masukkan kunci objek Amazon S3 tempat Anda mengunggah paket penerapan.

   g. Untuk LambdaFunctionS3 ObjectVersion, masukkan versi objek Amazon S3 tempat Anda mengunggah paket penerapan.

   h. Untuk ObjectLambdaAccessPointName, masukkan nama untuk Object Lambda Access Point Anda.

   i. Untuk S3 BucketName, masukkan nama bucket Amazon S3 yang akan dikaitkan dengan Titik Akses Objek Lambda Anda.

   j. Untuk SupportingAccessPointName, masukkan nama jalur akses pendukung Anda.

> **Note**
>
> Ini adalah titik akses yang terkait dengan bucket Amazon S3 yang Anda pilih pada langkah sebelumnya. Jika Anda tidak memiliki titik akses yang terkait dengan bucket Amazon S3 Anda, Anda dapat mengonfigurasi template untuk membuatnya untuk Anda dengan memilih true for. CreateNewSupportingAccessPoint

9. Pilih Berikutnya.
10. Pada Konfigurasikan halaman opsi stack, pilih Berikutnya.

    Untuk informasi selengkapnya tentang pengaturan opsional di halaman ini, lihat Mengatur AWS CloudFormation opsi tumpukan dalam AWS CloudFormation Panduan Pengguna.
11. Di halaman Tinjau, pilih Buat tumpukan.

Menggunakan AWS Cloud Development Kit (AWS CDK)

Untuk informasi selengkapnya tentang mengonfigurasi Titik Akses Objek Lambda menggunakan AWS CDK, *AWS::S3ObjectLambda*lihat Membangun Pustaka di AWS Cloud Development Kit (AWS CDK) Referensi API.

Otomatiskan pengaturan Lambda Objek S3 dengan templat CloudFormation

Anda dapat menggunakan AWS CloudFormation template untuk membuat Titik Akses Lambda Objek Amazon S3 dengan cepat. CloudFormationTemplate secara otomatis membuat sumber daya yang relevan, mengonfigurasi peran AWS Identity and Access Management (IAM), dan menyiapkan AWS Lambda fungsi yang secara otomatis menangani permintaan melalui Object Lambda Access Point. Dengan CloudFormation template, Anda dapat menerapkan praktik terbaik, meningkatkan postur keamanan Anda, dan mengurangi kesalahan yang disebabkan oleh proses manual.

GitHub Repositori ini berisi CloudFormation template dan kode sumber fungsi Lambda. Untuk petunjuk tentang cara menggunakan templat, lihat the section called “Membuat Titik Akses Objek Lambda”.

Fungsi Lambda yang disediakan dalam templat tidak menjalankan transformasi apa pun. Sebaliknya, ia mengembalikan objek Anda apa adanya dari sumber data yang mendasarinya. Anda dapat mengkloning fungsi dan menambahkan kode transformasi Anda sendiri untuk mengubah dan memproses data saat dikembalikan ke aplikasi. Untuk informasi selengkapnya tentang fungsi Anda, lihat the section called “Memodifikasi fungsi Lambda” dan the section called “Menulis fungsi Lambda”.

Memodifikasi templat

Membuat titik akses pendukung baru

S3 Lambda Objek menggunakan dua titik akses, Titik Akses Lambda Objek dan titik akses S3 standar, yang disebut sebagai titik akses pendukung. Saat Anda membuat permintaan ke Titik Akses Lambda Objek, S3 akan memanggil Lambda atas nama Anda, atau mendelegasikan permintaan tersebut ke titik akses pendukung, bergantung pada konfigurasi Lambda Objek S3. Anda dapat membuat titik akses pendukung baru dengan meneruskan parameter berikut sebagai bagian dari perintah `aws cloudformation deploy` saat mendeploy templat.

```
CreateNewSupportingAccessPoint=true
```

Mengonfigurasi muatan fungsi

Anda dapat mengonfigurasi payload untuk menyediakan data tambahan ke fungsi Lambda dengan meneruskan parameter berikut sebagai bagian dari perintah `aws cloudformation deploy` saat menerapkan templat.

```
LambdaFunctionPayload="format=json"
```

Mengaktifkan pemantauan Amazon CloudWatch

Anda dapat mengaktifkan CloudWatch pemantauan dengan meneruskan parameter berikut sebagai bagian dari `aws cloudformation deploy` perintah saat menerapkan template.

```
EnableCloudWatchMonitoring=true
```

Parameter ini memungkinkan Titik Akses Objek Lambda Anda untuk metrik permintaan Amazon S3 dan membuat CloudWatch dua alarm untuk memantau kesalahan sisi klien dan sisi server.

> **Note**
>
> CloudWatch Penggunaan Amazon akan dikenakan biaya tambahan. Untuk informasi lebih lanjut tentang metrik permintaan Amazon S3, lihat Pemantauan dan pencatatan titik akses. Untuk detail harganya, lihat Harga CloudWatch .

Mengonfigurasi konkurensi yang tersedia

Untuk mengurangi latensi, Anda dapat mengonfigurasi konkurensi yang disediakan untuk fungsi Lambda yang mendukung Titik Akses Lambda Object dengan mengedit templat untuk menyertakan baris berikut di bawah `Resources`.

```
LambdaFunctionVersion:
      Type: AWS::Lambda::Version
      Properties:
        FunctionName: !Ref LambdaFunction
        ProvisionedConcurrencyConfig:
            ProvisionedConcurrentExecutions: Integer
```

> **Note**
>
> Anda akan dikenakan biaya tambahan untuk penyediaan konkurensi. Untuk informasi selengkapnya tentang konkurensi yang disediakan, lihat Mengelola konkurensi Lambda yang disediakan dalam Panduan Pengembang AWS Lambda .
> Untuk detail harganya, lihat Harga AWS Lambda.

Memodifikasi fungsi Lambda

Mengubah nilai header untuk permintaan **`GetObject`**

Secara default, fungsi Lambda meneruskan semua header, kecuali `Content-Length` dan`ETag`, dari permintaan URL yang telah ditetapkan sebelumnya ke `GetObject` klien. Berdasarkan kode transformasi Anda dalam fungsi Lambda, Anda dapat memilih untuk mengirim nilai header baru ke `GetObject` klien.

Anda dapat memperbarui fungsi Lambda Anda untuk mengirim nilai header baru dengan meneruskannya dalam operasi API `WriteGetObjectResponse`.

Misalnya, jika fungsi Lambda Anda menerjemahkan teks dalam objek Amazon S3 ke bahasa yang berbeda, Anda dapat meneruskan nilai baru di header `Content-Language`. Anda dapat melakukan ini dengan memodifikasi fungsi `writeResponse` sebagai berikut:

```
async function writeResponse (s3Client: S3, requestContext: GetObjectContext,
 transformedObject: Buffer,
  headers: Headers): Promise<PromiseResult<{}, AWSError>> {
  const { algorithm, digest } = getChecksum(transformedObject);
```

```
  return s3Client.writeGetObjectResponse({
    RequestRoute: requestContext.outputRoute,
    RequestToken: requestContext.outputToken,
    Body: transformedObject,
    Metadata: {
      'body-checksum-algorithm': algorithm,
      'body-checksum-digest': digest
    },
    ...headers,
    ContentLanguage: 'my-new-language'
  }).promise();
}
```

Untuk daftar lengkap header yang didukung, lihat `WriteGetObjectResponse` dalam Referensi API Amazon Simple Storage Service.

Mengembalikan header metadata

Anda dapat memperbarui fungsi Lambda Anda untuk mengirim nilai header baru dengan meneruskannya dalam permintaan operasi API `WriteGetObjectResponse`.

```
async function writeResponse (s3Client: S3, requestContext: GetObjectContext,
 transformedObject: Buffer,
  headers: Headers): Promise<PromiseResult<{}, AWSError>> {
  const { algorithm, digest } = getChecksum(transformedObject);

  return s3Client.writeGetObjectResponse({
    RequestRoute: requestContext.outputRoute,
    RequestToken: requestContext.outputToken,
    Body: transformedObject,
    Metadata: {
      'body-checksum-algorithm': algorithm,
      'body-checksum-digest': digest,
      'my-new-header': 'my-new-value'
    },
    ...headers
  }).promise();
}
```

Mengembalikan kode status baru

Anda dapat mengembalikan kode status kustom ke klien `GetObject` dengan meneruskannya dalam permintaan operasi API `WriteGetObjectResponse`.

```
async function writeResponse (s3Client: S3, requestContext: GetObjectContext,
 transformedObject: Buffer,
  headers: Headers): Promise<PromiseResult<{}, AWSError>> {
  const { algorithm, digest } = getChecksum(transformedObject);

  return s3Client.writeGetObjectResponse({
    RequestRoute: requestContext.outputRoute,
    RequestToken: requestContext.outputToken,
    Body: transformedObject,
    Metadata: {
      'body-checksum-algorithm': algorithm,
      'body-checksum-digest': digest
    },
    ...headers,
    StatusCode: Integer
  }).promise();
}
```

Untuk daftar lengkap kode status didukung, lihat `WriteGetObjectResponse` dalam Referensi API Amazon Simple Storage Service.

Menerapkan **`Range`** dan **`partNumber`** parameter ke objek sumber

Secara default, Object Lambda Access Point yang dibuat oleh CloudFormation template dapat menangani parameter `Range` dan`partNumber`. Fungsi Lambda menerapkan rentang atau nomor bagian yang diminta ke objek yang diubah. Untuk melakukannya, fungsi harus mengunduh seluruh objek dan menjalankan transformasi. Dalam beberapa kasus, rentang objek Anda yang diubah mungkin dipetakan tepat ke rentang objek sumber Anda. Artinya, meminta rentang byte A-B pada objek sumber dan menjalankan transformasi mungkin menghasilkan hasil yang sama seperti meminta seluruh objek, menjalankan transformasi, dan mengembalikan rentang byte A-B pada objek yang diubah.

Dalam kasus seperti itu, Anda dapat mengubah implementasi fungsi Lambda untuk menerapkan rentang atau nomor bagian langsung ke objek sumber. Pendekatan ini mengurangi latensi fungsi keseluruhan dan memori yang diperlukan. Untuk informasi selengkapnya, lihat the section called “Bekerja dengan header Range dan partNumber”.

Menonaktifkan penanganan **`Range`** dan **`partNumber`**

Secara default, Object Lambda Access Point yang dibuat oleh CloudFormation template dapat menangani parameter `Range` dan`partNumber`. Jika Anda tidak memerlukan perilaku ini, Anda dapat menonaktifkannya dengan menghapus baris berikut dari templat:

```
AllowedFeatures:
  - GetObject-Range
  - GetObject-PartNumber
  - HeadObject-Range
  - HeadObject-PartNumber
```

Mengubah objek yang berukuran besar

Secara default, fungsi Lambda memproses seluruh objek dalam memori sebelum dapat mulai streaming respons ke S3 Lambda Objek. Anda dapat memodifikasi fungsi untuk mengalirkan respons saat melakukan transformasi. Melakukannya akan membantu mengurangi latensi transformasi dan ukuran memori fungsi Lambda. Untuk implementasi contoh tersebut, lihat Contoh konten terkompresi Stream.

Menggunakan Titik Akses Lambda Objek Amazon S3

Membuat permintaan melalui Titik Akses Lambda Objek Amazon S3 berfungsi sama seperti membuat permintaan melalui titik akses lainnya. Untuk informasi selengkapnya tentang cara membuat permintaan melalui titik akses, lihat Menggunakan jalur akses Amazon S3 untuk bucket tujuan umum. Anda dapat membuat permintaan melalui Titik Akses Objek Lambda dengan menggunakan konsol Amazon S3 AWS Command Line Interface ,AWS CLI() AWS SDKs, atau Amazon S3 REST API.

> **Important**
>
> Amazon Resource Names (ARNs) untuk Object Lambda Access Points menggunakan nama layanan. `s3-object-lambda` Dengan demikian, Object Lambda Access Point ARNs dimulai dengan`arn:aws::s3-object-lambda`, bukan`arn:aws::s3`, yang digunakan dengan titik akses lainnya.

Cara untuk menemukan ARN untuk Titik Akses Lambda Objek Anda

Untuk menggunakan Object Lambda Access Point dengan AWS CLI atau AWS SDKs, Anda perlu mengetahui Amazon Resource Name (ARN) dari Object Lambda Access Point. Contoh berikut

menunjukkan cara untuk menemukan ARN untuk Titik Akses Lambda Objek dengan menggunakan konsol Amazon S3 atau. AWS CLI

Menggunakan konsol S3

Untuk menemukan ARN untuk Titik Akses Lambda Objek Anda dengan menggunakan konsol

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi sebelah kiri, pilih Titik Akses Lambda Objek.
3. Pilih tombol opsi di samping Titik Akses Lambda Objek yang ARN-nya ingin Anda salin.
4. Pilih Salin ARN.

Menggunakan AWS CLI

Untuk menemukan ARN untuk Object Lambda Access Point Anda dengan menggunakan AWS CLI

1. Untuk mengambil daftar Titik Akses Lambda Objek yang terkait dengan Akun AWS Anda, jalankan perintah berikut. Sebelum menjalankan perintah, ganti ID akun *111122223333* dengan Akun AWS ID Anda.

```
aws s3control list-access-points-for-object-lambda --account-id 111122223333
```

2. Tinjau output perintah untuk menemukan Titik Akses Lambda Objek ARN yang ingin Anda gunakan. Output dari perintah sebelumnya akan terlihat seperti contoh berikut.

```
{
    "ObjectLambdaAccessPointList": [
        {
            "Name": "my-object-lambda-ap",
            "ObjectLambdaAccessPointArn": "arn:aws:s3-object-lambda:us-
east-1:111122223333:accesspoint/my-object-lambda-ap"
        },
        ...
    ]
}
```

Cara menggunakan alias gaya bucket untuk bucket S3 Anda Titik Akses Lambda Objek

Saat Anda membuat Titik Akses Lambda Objek, Amazon S3 secara otomatis menghasilkan alias unik untuk Titik Akses Lambda Objek Anda. Anda dapat menggunakan alias ini alih-alih nama bucket Amazon S3 atau Amazon Resource Name (ARN) Titik Akses Lamda Objek dalam permintaan untuk operasi bidang data titik akses. Untuk mengetahui daftar operasi ini, lihat Kompatibilitas titik akses.

Nama alias Titik Akses Lambda Objek dibuat dalam namespace yang sama dengan bucket Amazon S3. Nama alias ini dibuat secara otomatis, dan tidak dapat diubah. Untuk Titik Akses Lambda Objek yang ada, alias secara otomatis ditetapkan untuk digunakan. Nama alias Titik Akses Lambda Objek memenuhi semua persyaratan nama bucket Amazon S3 yang valid dan terdiri dari bagian-bagian berikut:

Object Lambda Access Point name prefix`-`*metadata*`--ol-s3`

> **Note**
>
> Sufiks `--ol-s3`dicadangkan untuk nama alias Titik Akses Lambda Objek dan tidak dapat digunakan untuk nama bucket atau Titik Akses Lambda Objek. Untuk informasi selengkapnya tentang penamaan bucket di Amazon S3, lihat Aturan penamaan bucket tujuan umum.

Contoh berikut menunjukkan ARN dan alias Titik Akses Lambda Objek untuk Titik Akses Lambda Objek bernama *my-object-lambda-access-point*:

- ARN–`arn:aws:s3-object-lambda:`*region*`:`*account-id*`:accesspoint/`*my-object-lambda-access-point*
- Alias Titik Akses Lambda Objek–*my-object-lambda-acc-1a4n8yjrb3kda96f67zwrwiiuse1a*`--ol-s3`

Bila Anda menggunakan Titik Akses Lambda Objek, Anda dapat menggunakan nama alias Titik Akses Lambda Objek tanpa memerlukan perubahan kode yang ekstensif.

Ketika Anda menghapus Titik Akses Lambda Objek, nama alias Titik Akses Lambda Objek menjadi tidak aktif dan tidak tersedia.

Cara menemukan alias untuk Titik Akses Lambda Objek Anda

Menggunakan konsol S3

Untuk menemukan alias untuk Titik Akses Lambda Objek Anda dengan menggunakan konsol

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi sebelah kiri, pilih Titik Akses Lambda Objek.
3. Untuk Access Point Lambda Objek yang ingin Anda gunakan, salin nilai Alias Titik Akses Lambda Objek.

Menggunakan AWS CLI

Saat Anda membuat Titik Akses Lambda Objek, Amazon S3 secara otomatis menghasilkan nama alias Titik Akses Lambda Objek, seperti yang ditunjukkan pada perintah contoh berikut. Untuk menjalankan perintah ini, ganti *`user input placeholders`* dengan informasi Anda sendiri. Untuk informasi tentang cara membuat Titik Akses Objek Lambda menggunakan AWS CLI, lihat. Untuk membuat Object Lambda Access Point dengan menggunakan AWS CLI

```
aws s3control create-access-point-for-object-lambda --account-id 111122223333 --
name my-object-lambda-access-point --configuration file://my-olap-configuration.json
{
    "ObjectLambdaAccessPointArn": "arn:aws:s3:region:111122223333:accesspoint/my-
access-point",
    "Alias": {
        "Value": "my-object-lambda-acc-1a4n8yjrb3kda96f67zwrwiiuse1a--ol-s3",
        "Status": "READY"
    }
}
```

Nama alias Titik Akses Lambda Objek yang dihasilkan memiliki dua bidang:

- Kolom `Value` tersebut adalah nilai alias Titik Akses Lambda Objek.
- Bidang `Status` adalah status alias Titik Akses Lambda Objek. Jika statusnya adalah `PROVISIONING`, Amazon S3 menyediakan alias Titik Akses Lambda Objek, dan alias belum siap untuk digunakan. Jika statusnya adalah `READY`, alias Titik Akses Lambda Objek telah berhasil disediakan dan siap digunakan.

Untuk informasi selengkapnya tentang jenis data `ObjectLambdaAccessPointAlias` di API REST, lihat CreateAccessPointForObjectLambdadan ObjectLambdaAccessPointAlias dalam Referensi API Amazon Simple Storage Service.

Cara untuk menggunakan alias Titik Akses Lambda Objek alias

Anda dapat menggunakan alias Titik Akses Lambda Objek, bukan nama bucket Amazon S3 untuk operasi yang tercantum di dalam Kompatibilitas titik akses.

AWS CLI Contoh berikut untuk `get-bucket-location` perintah menggunakan alias access point bucket untuk mengembalikan tempat Wilayah AWS bucket berada. Untuk menjalankan perintah ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3api get-bucket-location --bucket my-object-lambda-
acc-w7i37nq6xuzgax3jw3oqtifiusw2a--ol-s3

{
    "LocationConstraint": "us-west-2"
}
```

Jika alias Titik Akses Lambda Object dalam permintaannya tidak valid, kode kesalahan `InvalidAccessPointAliasError` dikembalikan. Untuk informasi selengkapnya tentang `InvalidAccessPointAliasError`, lihat Daftar Kode Kesalahan dalam Referensi API Amazon Simple Storage Service.

Keterbatasan alias Titik Akses Lambda Objek sama dengan alias titik akses. Untuk informasi lebih lanjut tentang batasan alias titik akses, lihat Keterbatasan alias titik akses.

Pertimbangan keamanan untuk Titik Akses S3 Lambda Objek

Dengan Amazon S3 Object Lambda, Anda dapat melakukan transformasi kustom pada data saat meninggalkan Amazon S3 dengan menggunakan skala dan fleksibilitas sebagai platform komputasi. AWS Lambda S3 dan Lambda tetap aman secara default, namun untuk menjaga keamanan ini, pertimbangan kustom oleh pembuat fungsi Lambda diperlukan. S3 Lambda Objek mengharuskan semua akses dilakukan oleh pengguna utama yang diautentikasi (tidak ada akses anonim) dan melalui HTTPS.

Untuk memitigasi risiko keamanan, kami merekomendasikan hal berikut:

- Cakup peran eksekusi Lambda ke kumpulan izin sekecil mungkin.

- Jika memungkinkan, pastikan fungsi Lambda Anda mengakses Amazon S3 melalui URL yang telah ditentukan sebelumnya.

Mengonfigurasi kebijakan IAM

Kebijakan sumber daya dukungan titik akses S3 AWS Identity and Access Management (IAM) yang memungkinkan Anda mengontrol penggunaan titik akses berdasarkan sumber daya, pengguna, atau kondisi lainnya. Untuk informasi selengkapnya, lihat Mengonfigurasi kebijakan IAM untuk Titik Akses Lambda Objek.

Perilaku enkripsi

Karena Titik Akses Objek Lambda menggunakan Amazon S3 AWS Lambda dan, ada perbedaan dalam perilaku enkripsi. Untuk informasi selengkapnya tentang enkripsi default S3, lihat Mengatur perilaku enkripsi sisi server default untuk bucket Amazon S3.

- Saat Anda menggunakan enkripsi di sisi server S3 dengan Titik Akses Lambda Object, objek didekripsi sebelum dikirimkan ke Lambda. Setelah objek dikirim ke Lambda, objek diproses secara tidak terenkripsi (dalam kasus permintaan `GET` atau `HEAD`).
- Untuk mencegah kunci enkripsi dicatat, S3 menolak permintaan `GET` dan `HEAD` untuk objek yang dienkripsi menggunakan enkripsi di sisi server dengan kunci yang disediakan pelanggan (SSE-C). Namun, fungsi Lambda mungkin masih mengambil objek ini jika memiliki akses ke kunci klien yang disediakan.
- Saat menggunakan enkripsi di sisi klien S3 dengan Titik Akses Lambda Objeks, pastikan Lambda memiliki akses ke kunci enkripsi sehingga dapat mendekripsi dan mengenkripsi kembali objek.

Keamanan titik akses

S3 Lambda Objek menggunakan dua titik akses, Titik Akses Lambda Objek dan titik akses S3 standar, yang disebut sebagai titik akses pendukung. Saat Anda membuat permintaan ke Titik Akses Lambda Objek, S3 akan memanggil Lambda atas nama Anda, atau mendelegasikan permintaan tersebut ke titik akses pendukung, bergantung pada konfigurasi Lambda Objek S3. Ketika Lambda dipanggil untuk permintaan, S3 menghasilkan URL yang telah ditandatangani sebelumnya ke objek Anda atas nama Anda melalui titik akses pendukung. Fungsi Lambda Anda menerima URL ini sebagai input saat fungsi tersebut dipanggil.

Anda dapat mengatur fungsi Lambda Anda untuk menggunakan URL presigned ini untuk mengambil objek asli, alih-alih memanggil S3 secara langsung. Dengan menggunakan model ini, Anda dapat

menerapkan batas keamanan yang lebih baik ke objek Anda. Anda dapat membatasi akses objek langsung melalui bucket S3 atau titik akses S3 ke serangkaian peran IAM atau pengguna terbatas. Pendekatan ini juga melindungi fungsi Lambda Anda agar tidak terkena masalah confused deputy, di mana fungsi yang salah dikonfigurasi dengan izin berbeda dari pemanggil dapat mengizinkan atau menolak akses ke objek, padahal seharusnya tidak demikian.

Akses publik Titik Akses Lambda Objek

S3 Lambda Objek tidak mengizinkan akses anonim atau publik karena Amazon S3 harus mengotorisasi identitas Anda untuk menyelesaikan permintaan S3 Lambda Objek. Saat memanggil permintaan melalui Titik Akses Lambda Objek, Anda harus memiliki izin untuk fungsi Lambda `lambda:InvokeFunction` yang dikonfigurasi. Demikian pula, saat menjalankan operasi API lainnya melalui Titik Akses Lambda Objek, Anda harus memiliki izin `s3:*` yang diperlukan.

Tanpa izin ini, permintaan untuk menginvokasi Lambda atau mendelegasikan ke S3 akan gagal dengan kesalahan HTTP 403 (Forbidden). Semua akses harus dilakukan oleh pengguna utama yang diautentikasi. Jika Anda memerlukan akses publik, Anda dapat menggunakan Lambda @Edge sebagai alternatif yang memungkinkan. Untuk informasi selengkapnya, lihat Menyesuaikan di tepi dengan Lambda @Edge di Panduan Pengembang CloudFront Amazon.

Alamat IP Titik Akses Lambda Objek

Subnet `describe-managed-prefix-lists` mendukung titik akhir gateway cloud privat virtual (VPC), dan terkait dengan tabel perutean titik akhir VPC. Karena Titik Akses Objek Lambda tidak mendukung VPC gateway, rentang IP-nya hilang. Rentang yang hilang adalah milik Amazon S3, namun tidak didukung oleh titik akhir VPC gateway. Untuk informasi selengkapnya`describe-managed-prefix-lists`, lihat DescribeManagedPrefixListsdi Referensi EC2 API Amazon dan rentang alamat AWS IP di Referensi Umum AWS.

Mengonfigurasi kebijakan IAM untuk Titik Akses Lambda Objek

Kebijakan sumber daya dukungan titik akses Amazon S3 AWS Identity and Access Management (IAM) yang dapat Anda gunakan untuk mengontrol penggunaan titik akses menurut sumber daya, pengguna, atau kondisi lainnya. Anda dapat mengontrol akses melalui kebijakan sumber daya opsional pada Titik Akses Lambda Objek, atau kebijakan sumber daya untuk mendukung titik akses. Sebagai step-by-step contoh, lihat Tutorial: Mengubah data untuk aplikasi Anda dengan S3 Object Lambda danTutorial: Mendeteksi dan menyunting data PII dengan Objek S3 Lambda dan Amazon Comprehend.

Empat sumber daya berikut harus memiliki izin yang diberikan untuk bekerja dengan Titik Akses Lambda Objek:

- Identitas IAM, seperti pengguna atau peran. Untuk informasi selengkapnya tentang identitas IAM dan praktik terbaik, lihat Identitas IAM (pengguna, grup pengguna, dan peran) dalam Panduan Pengguna IAM.
- Titik akses standar terhubung ke sumber data yang mendasarinya seperti bucket S3 atau Amazon FSx untuk volume OpenZFS. Saat Anda bekerja dengan Titik Akses Lambda Objek, titik akses standar ini dikenal sebagai titik akses pendukung.
- Titik akses Lambda Object.
- AWS Lambda Fungsinya.

> **Important**
>
> Sebelum menyimpan kebijakan, pastikan untuk menyelesaikan peringatan keamanan, kesalahan, peringatan umum, dan saran dari. AWS Identity and Access Management Access Analyzer Penganalisis Akses IAM menjalankan pemeriksaan kebijakan untuk memvalidasi kebijakan Anda terhadap tata bahasa kebijakan IAM dan praktik terbaik. Pemeriksaan ini menghasilkan temuan dan memberikan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang fungsional dan sesuai dengan praktik terbaik keamanan.
>
> Untuk mempelajari validasi kebijakan menggunakan Penganalisis Akses IAM lebih lanjut, lihat Memvalidasi kebijakan Penganalisis Akses IAM di Panduan Pengguna IAM. Untuk melihat daftar peringatan, kesalahan, dan saran yang ditampilkan oleh Penganalisis Akses IAM, lihat referensi pemeriksaan kebijakan Penganalisis Akses IAM.

Contoh kebijakan berikut mengasumsikan bahwa Anda memiliki sumber daya berikut:

- Bucket Amazon S3 dengan Amazon Resource Name (ARN) berikut:

  `arn:aws:s3:::amzn-s3-demo-bucket1`

- Titik akses Amazon S3 standard pada bucket ini dengan ARN berikut:

  `arn:aws:s3:us-east-1:111122223333:accesspoint/my-access-point`

- Titik akses Lambda Objek dengan ARN berikut:

`arn:aws:s3-object-lambda:us-east-1:111122223333:accesspoint/my-object-lambda-ap`

- AWS Lambda Fungsi dengan ARN berikut:

`arn:aws:lambda:us-east-1:111122223333:function:MyObjectLambdaFunction`

> **Note**
>
> Jika Anda menggunakan fungsi Lambda dari akun, Anda harus menyertakan versi fungsi tertentu dalam pernyataan kebijakan Anda. Dalam contoh ARN berikut, versinya ditandai dengan *1*:
> `arn:aws:lambda:us-east-1:111122223333:function:MyObjectLambdaFunction:1`
> Lambda tidak mendukung penambahan kebijakan IAM ke versi. $LATEST Untuk informasi selengkapnya tentang versi fungsi Lambda, lihat Versi fungsi Lambda dalam Panduan Pengembang AWS Lambda .

Example – Kebijakan bucket yang mendelegasikan kontrol akses ke titik akses standar

Contoh kebijakan bucket S3 berikut mendelegasikan kontrol akses untuk sebuah bucket ke titik akses standar bucket. Kebijakan ini memungkinkan akses penuh ke semua titik akses yang dimiliki oleh akun pemilik bucket. Oleh karena itu, semua akses ke bucket ini dikendalikan oleh kebijakan yang melekat pada titik aksesnya. Pengguna dapat membaca dari bucket hanya melalui titik akses, yang berarti bahwa operasi hanya dapat dipanggil melalui titik akses. Untuk informasi selengkapnya, lihat Mendelegasikan kontrol akses ke titik akses.

```
{
    "Version": "2012-10-17",
    "Statement" : [
    {
        "Effect": "Allow",
        "Principal" : { "AWS":"account-ARN"},
        "Action" : "*",
        "Resource" : [
            "arn:aws:s3:::amzn-s3-demo-bucket1",
            "arn:aws:s3:::amzn-s3-demo-bucket1/*"
        ],
```

```
        "Condition": {
            "StringEquals" : { "s3:DataAccessPointAccount" : "Bucket owner's account
 ID" }
        }
    }]
}
```

Example — Kebijakan IAM yang memberikan pengguna izin yang diperlukan untuk menggunakan Titik Akses Objek Lambda

Kebijakan IAM berikut memberikan izin pengguna ke fungsi Lambda, titik akses standar, dan Titik Akses Lambda Objek.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowLambdaInvocation",
      "Action": [
        "lambda:InvokeFunction"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:lambda:us-
east-1:111122223333:function:MyObjectLambdaFunction:1",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:CalledVia": [
            "s3-object-lambda.amazonaws.com"
          ]
        }
      }
    },
    {
      "Sid": "AllowStandardAccessPointAccess",
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:us-east-1:111122223333:accesspoint/my-access-point/*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:CalledVia": [
```

```
            "s3-object-lambda.amazonaws.com"
          ]
        }
      }
    },
    {
      "Sid": "AllowObjectLambdaAccess",
      "Action": [
        "s3-object-lambda:Get*",
        "s3-object-lambda:List*"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3-object-lambda:us-east-1:111122223333:accesspoint/my-
object-lambda-ap"
    }
  ]
}
```

Aktifkan izin untuk peran eksekusi Lambda

Ketika `GET` permintaan dibuat ke Object Lambda Access Point, fungsi Lambda Anda memerlukan izin untuk mengirim data ke S3 Object Lambda Access Point. Izin ini diberikan dengan mengaktifkan izin `s3-object-lambda:WriteGetObjectResponse` pada peran eksekusi fungsi Lambda Anda. Anda dapat membuat peran eksekusi baru atau memperbarui peran yang sudah ada.

> **Note**
>
> Fungsi Anda memerlukan izin `s3-object-lambda:WriteGetObjectResponse` hanya jika Anda membuat permintaan `GET`.

Untuk membuat peran eksekusi di konsol IAM

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi sebelah kiri, pilih Peran.
3. Pilih Buat peran.
4. Di bawah Kasus penggunaan umum, pilih Lambda.
5. Pilih Berikutnya.

6. Pada halaman Tambahkan izin, cari kebijakan AWS terkelola `AmazonS3ObjectLambdaExecutionRolePolicy`, lalu pilih kotak centang di samping nama kebijakan.

   Kebijakan ini harus berisi Tindakan `s3-object-lambda:WriteGetObjectResponse`.
7. Pilih Berikutnya.
8. Pada halaman Nama, tinjau, dan buat, untuk Nama peran, masukkan **`s3-object-lambda-role`**.
9. (Opsional) Tambahkan deskripsi dan tag untuk peran ini.
10. Pilih Buat peran.
11. Terapkan **`s3-object-lambda-role`** yang baru dibuat sebagai peran eksekusi fungsi Lambda Anda. Ini dapat dilakukan selama atau setelah pembuatan fungsi Lambda di konsol Lambda.

Untuk informasi selengkapnya tentang peran eksekusi, lihat Peran eksekusi Lambda di Panduan Pengembang AWS Lambda .

Menggunakan kunci konteks dengan Titik Akses Lambda Objek

S3 Lambda Objek akan mengevaluasi kunci konteks seperti `s3-object-lambda:TlsVersion` atau `s3-object-lambda:AuthType` yang terkait dengan koneksi atau penandatanganan permintaan. Semua kunci konteks lainnya, seperti `s3:prefix`, dievaluasi oleh Amazon S3.

Dukungan CORS Titik Akses Lambda Objek

Ketika S3 Lambda Objek menerima permintaan dari browser atau permintaannya menyertakan header `Origin`, S3 Lambda Objek selalu menambahkan bidang header `"AllowedOrigins":"*"`.

Untuk informasi selengkapnya, lihat Berbagi sumber daya lintas asal (CORS).

Menulis fungsi Lambda untuk Titik Akses S3 Lambda Objek

Bagian ini merinci cara menulis AWS Lambda fungsi untuk digunakan dengan Amazon S3 Object Lambda Access Points.

Untuk mempelajari tentang end-to-end prosedur lengkap untuk beberapa tugas Lambda Objek S3, lihat berikut ini:

- Tutorial: Mengubah data untuk aplikasi Anda dengan S3 Object Lambda

- Tutorial: Mendeteksi dan menyunting data PII dengan Objek S3 Lambda dan Amazon Comprehend
- Tutorial: Menggunakan S3 Lambda Objek untuk menandai gambar secara dinamis saat gambarnya diambil

Topik

- Bekerja dengan GetObject permintaan di Lambda
- Bekerja dengan permintaan HeadObject di Lambda
- Bekerja dengan permintaan ListObjects di Lambda
- Bekerja dengan permintaan ListObjectsV2 di Lambda
- Format konteks peristiwa dan penggunaan
- Bekerja dengan header Range dan partNumber

Bekerja dengan **`GetObject`** permintaan di Lambda

Bagian ini mengasumsikan bahwa Titik Akses Lambda Objek Anda dikonfigurasi untuk memanggil fungsi Lambda `GetObject`. S3 Lambda Objek mencakup operasi API Amazon S3, `WriteGetObjectResponse`, yang mengaktifkan fungsi Lambda untuk memberikan data yang disesuaikan dan respons header untuk pemanggil `GetObject`.

`WriteGetObjectResponse` memberikan Anda kontrol ekstensif atas kode status, header respons, dan badan respons, berdasarkan kebutuhan pemrosesan Anda. Anda dapat menggunakan `WriteGetObjectResponse` untuk menanggapi dengan seluruh objek yang berubah, bagian dari objek yang berubah, atau tanggapan lain berdasarkan konteks aplikasi Anda. Bagian berikut menampilkan contoh-contoh unik penggunaan operasi API `WriteGetObjectResponse`.

- Contoh 1: Menanggapi dengan kode status HTTP 403 (Forbidden)
- Contoh 2: Merespons dengan citra yang berubah
- Contoh 3: Melakukan streaming konten terkompresi

Contoh 1: Menanggapi dengan kode status HTTP 403 (Forbidden)

Anda dapat menggunakan `WriteGetObjectResponse` untuk menanggapi dengan kode status HTTP 403 (Terlarang) berdasarkan isi dari objek.

Java

```
package com.amazon.s3.objectlambda;

import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.events.S3ObjectLambdaEvent;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.model.WriteGetObjectResponseRequest;

import java.io.ByteArrayInputStream;
import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class Example1 {

    public void handleRequest(S3ObjectLambdaEvent event, Context context) throws
 Exception {
        AmazonS3 s3Client = AmazonS3Client.builder().build();

        // Check to see if the request contains all of the necessary information.
        // If it does not, send a 4XX response and a custom error code and message.
        // Otherwise, retrieve the object from S3 and stream it
        // to the client unchanged.
        var tokenIsNotPresent = !
event.getUserRequest().getHeaders().containsKey("requiredToken");
        if (tokenIsNotPresent) {
            s3Client.writeGetObjectResponse(new WriteGetObjectResponseRequest()
                    .withRequestRoute(event.outputRoute())
                    .withRequestToken(event.outputToken())
                    .withStatusCode(403)
                    .withContentLength(0L).withInputStream(new
 ByteArrayInputStream(new byte[0]))
                    .withErrorCode("MissingRequiredToken")
                    .withErrorMessage("The required token was not present in the
 request."));
            return;
        }

        // Prepare the presigned URL for use and make the request to S3.
        HttpClient httpClient = HttpClient.newBuilder().build();
        var presignedResponse = httpClient.send(
                HttpRequest.newBuilder(new URI(event.inputS3Url())).GET().build(),
```

```
                HttpResponse.BodyHandlers.ofInputStream());

        // Stream the original bytes back to the caller.
        s3Client.writeGetObjectResponse(new WriteGetObjectResponseRequest()
                .withRequestRoute(event.outputRoute())
                .withRequestToken(event.outputToken())
                .withInputStream(presignedResponse.body()));
    }
}
```

Python

```
import boto3
import requests

def handler(event, context):
    s3 = boto3.client('s3')

    """
    Retrieve the operation context object from the event. This object indicates
 where the WriteGetObjectResponse request
    should be delivered and contains a presigned URL in 'inputS3Url' where we can
 download the requested object from.
    The 'userRequest' object has information related to the user who made this
 'GetObject' request to
    S3 Object Lambda.
    """
    get_context = event["getObjectContext"]
    user_request_headers = event["userRequest"]["headers"]

    route = get_context["outputRoute"]
    token = get_context["outputToken"]
    s3_url = get_context["inputS3Url"]

    # Check for the presence of a 'CustomHeader' header and deny or allow based on
 that header.
    is_token_present = "SuperSecretToken" in user_request_headers

    if is_token_present:
        # If the user presented our custom 'SuperSecretToken' header, we send the
 requested object back to the user.
        response = requests.get(s3_url)
```

```
        s3.write_get_object_response(RequestRoute=route, RequestToken=token,
 Body=response.content)
    else:
        # If the token is not present, we send an error back to the user.
        s3.write_get_object_response(RequestRoute=route, RequestToken=token,
 StatusCode=403,
        ErrorCode="NoSuperSecretTokenFound", ErrorMessage="The request was not
 secret enough.")

    # Gracefully exit the Lambda function.
    return { 'status_code': 200 }
```

Node.js

```
const { S3 } = require('aws-sdk');
const axios = require('axios').default;

exports.handler = async (event) => {
    const s3 = new S3();

    // Retrieve the operation context object from the event. This object indicates
 where the WriteGetObjectResponse request
    // should be delivered and contains a presigned URL in 'inputS3Url' where we can
 download the requested object from.
    // The 'userRequest' object has information related to the user who made this
 'GetObject' request to S3 Object Lambda.
    const { userRequest, getObjectContext } = event;
    const { outputRoute, outputToken, inputS3Url } = getObjectContext;

    // Check for the presence of a 'CustomHeader' header and deny or allow based on
 that header.
    const isTokenPresent = Object
        .keys(userRequest.headers)
        .includes("SuperSecretToken");

    if (!isTokenPresent) {
        // If the token is not present, we send an error back to the user. The
 'await' in front of the request
        // indicates that we want to wait for this request to finish sending before
 moving on.
        await s3.writeGetObjectResponse({
            RequestRoute: outputRoute,
            RequestToken: outputToken,
```

```
            StatusCode: 403,
            ErrorCode: "NoSuperSecretTokenFound",
            ErrorMessage: "The request was not secret enough.",
        }).promise();
    } else {
        // If the user presented our custom 'SuperSecretToken' header, we send the
 requested object back to the user.
        // Again, note the presence of 'await'.
        const presignedResponse = await axios.get(inputS3Url);
        await s3.writeGetObjectResponse({
            RequestRoute: outputRoute,
            RequestToken: outputToken,
            Body: presignedResponse.data,
        }).promise();
    }

    // Gracefully exit the Lambda function.
    return { statusCode: 200 };
}
```

Contoh 2: Merespons dengan citra yang berubah

Saat melakukan transformasi citra, Anda mungkin memerlukan semua byte objek sumber sebelum Anda dapat mulai memprosesnya. Dalam kasus ini, permintaan `WriteGetObjectResponse` Anda akan mengembalikan seluruh objek ke aplikasi permintaan dalam satu panggilan.

Java

```
package com.amazon.s3.objectlambda;

import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.events.S3ObjectLambdaEvent;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.model.WriteGetObjectResponseRequest;

import javax.imageio.ImageIO;
import java.awt.image.BufferedImage;
import java.awt.Image;
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.net.URI;
```

```
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class Example2 {

    private static final int HEIGHT = 250;
    private static final int WIDTH = 250;

    public void handleRequest(S3ObjectLambdaEvent event, Context context) throws
 Exception {
        AmazonS3 s3Client = AmazonS3Client.builder().build();
        HttpClient httpClient = HttpClient.newBuilder().build();

        // Prepare the presigned URL for use and make the request to S3.
        var presignedResponse = httpClient.send(
                HttpRequest.newBuilder(new URI(event.inputS3Url())).GET().build(),
                HttpResponse.BodyHandlers.ofInputStream());

        // The entire image is loaded into memory here so that we can resize it.
        // Once the resizing is completed, we write the bytes into the body
        // of the WriteGetObjectResponse request.
        var originalImage = ImageIO.read(presignedResponse.body());
        var resizingImage = originalImage.getScaledInstance(WIDTH, HEIGHT,
 Image.SCALE_DEFAULT);
        var resizedImage = new BufferedImage(WIDTH, HEIGHT,
 BufferedImage.TYPE_INT_RGB);
        resizedImage.createGraphics().drawImage(resizingImage, 0, 0, WIDTH, HEIGHT,
 null);

        var baos = new ByteArrayOutputStream();
        ImageIO.write(resizedImage, "png", baos);

        // Stream the bytes back to the caller.
        s3Client.writeGetObjectResponse(new WriteGetObjectResponseRequest()
                .withRequestRoute(event.outputRoute())
                .withRequestToken(event.outputToken())
                .withInputStream(new ByteArrayInputStream(baos.toByteArray())));
    }
}
```

Python

```
import boto3
import requests
import io
from PIL import Image

def handler(event, context):
    """
    Retrieve the operation context object from the event. This object indicates
 where the WriteGetObjectResponse request
    should be delivered and has a presigned URL in 'inputS3Url' where we can
 download the requested object from.
    The 'userRequest' object has information related to the user who made this
 'GetObject' request to
    S3 Object Lambda.
    """
    get_context = event["getObjectContext"]
    route = get_context["outputRoute"]
    token = get_context["outputToken"]
    s3_url = get_context["inputS3Url"]

    """
    In this case, we're resizing .png images that are stored in S3 and are
 accessible through the presigned URL
    'inputS3Url'.
    """
    image_request = requests.get(s3_url)
    image = Image.open(io.BytesIO(image_request.content))
    image.thumbnail((256,256), Image.ANTIALIAS)

    transformed = io.BytesIO()
    image.save(transformed, "png")

    # Send the resized image back to the client.
    s3 = boto3.client('s3')
    s3.write_get_object_response(Body=transformed.getvalue(), RequestRoute=route,
 RequestToken=token)

    # Gracefully exit the Lambda function.
    return { 'status_code': 200 }
```

Node.js

```
const { S3 } = require('aws-sdk');
const axios = require('axios').default;
const sharp = require('sharp');

exports.handler = async (event) => {
    const s3 = new S3();

    // Retrieve the operation context object from the event. This object indicates
 where the WriteGetObjectResponse request
    // should be delivered and has a presigned URL in 'inputS3Url' where we can
 download the requested object from.
    const { getObjectContext } = event;
    const { outputRoute, outputToken, inputS3Url } = getObjectContext;

    // In this case, we're resizing .png images that are stored in S3 and are
 accessible through the presigned URL
    // 'inputS3Url'.
    const { data } = await axios.get(inputS3Url, { responseType: 'arraybuffer' });

    // Resize the image.
    const resized = await sharp(data)
        .resize({ width: 256, height: 256 })
        .toBuffer();

    // Send the resized image back to the client.
    await s3.writeGetObjectResponse({
        RequestRoute: outputRoute,
        RequestToken: outputToken,
        Body: resized,
    }).promise();

    // Gracefully exit the Lambda function.
    return { statusCode: 200 };
}
```

Contoh 3: Melakukan streaming konten terkompresi

Saat Anda mengompresi objek, data terkompresi dihasilkan secara bertahap. Akibatnya, Anda dapat menggunakan permintaan `WriteGetObjectResponse` Anda untuk mengembalikan data terkompresi segera setelah semuanya siap. Seperti yang ditunjukkan dalam contoh ini, Anda tidak perlu mengetahui panjang transformasi yang telah selesai.

Java

```
package com.amazon.s3.objectlambda;

import com.amazonaws.services.lambda.runtime.events.S3ObjectLambdaEvent;
import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.model.WriteGetObjectResponseRequest;

import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class Example3 {

    public void handleRequest(S3ObjectLambdaEvent event, Context context) throws
 Exception {
        AmazonS3 s3Client = AmazonS3Client.builder().build();
        HttpClient httpClient = HttpClient.newBuilder().build();

        // Request the original object from S3.
        var presignedResponse = httpClient.send(
                HttpRequest.newBuilder(new URI(event.inputS3Url())).GET().build(),
                HttpResponse.BodyHandlers.ofInputStream());

        // Consume the incoming response body from the presigned request,
        // apply our transformation on that data, and emit the transformed bytes
        // into the body of the WriteGetObjectResponse request as soon as they're
 ready.
        // This example compresses the data from S3, but any processing pertinent
        // to your application can be performed here.
        var bodyStream = new GZIPCompressingInputStream(presignedResponse.body());

        // Stream the bytes back to the caller.
        s3Client.writeGetObjectResponse(new WriteGetObjectResponseRequest()
                .withRequestRoute(event.outputRoute())
                .withRequestToken(event.outputToken())
                .withInputStream(bodyStream));
    }
```

```
}
```

Python

```
import boto3
import requests
import zlib
from botocore.config import Config


"""
A helper class to work with content iterators. Takes an interator and compresses the
 bytes that come from it. It
implements 'read' and '__iter__' so that the SDK can stream the response.
"""
class Compress:
    def __init__(self, content_iter):
        self.content = content_iter
        self.compressed_obj = zlib.compressobj()

    def read(self, _size):
        for data in self.__iter__()
            return data

    def __iter__(self):
        while True:
            data = next(self.content)
            chunk = self.compressed_obj.compress(data)
            if not chunk:
                break

            yield chunk

        yield self.compressed_obj.flush()


def handler(event, context):
    """
    Setting the 'payload_signing_enabled' property to False allows us to send a
 streamed response back to the client.
    in this scenario, a streamed response means that the bytes are not buffered into
 memory as we're compressing them,
    but instead are sent straight to the user.
```

```
    """
    my_config = Config(
        region_name='eu-west-1',
        signature_version='s3v4',
        s3={
            "payload_signing_enabled": False
        }
    )
    s3 = boto3.client('s3', config=my_config)

    """
    Retrieve the operation context object from the event. This object indicates
where the WriteGetObjectResponse request
    should be delivered and has a presigned URL in 'inputS3Url' where we can
download the requested object from.
    The 'userRequest' object has information related to the user who made this
'GetObject' request to S3 Object Lambda.
    """
    get_context = event["getObjectContext"]
    route = get_context["outputRoute"]
    token = get_context["outputToken"]
    s3_url = get_context["inputS3Url"]

    # Compress the 'get' request stream.
    with requests.get(s3_url, stream=True) as r:
        compressed = Compress(r.iter_content())

        # Send the stream back to the client.
        s3.write_get_object_response(Body=compressed, RequestRoute=route,
RequestToken=token, ContentType="text/plain",
                                     ContentEncoding="gzip")

    # Gracefully exit the Lambda function.
    return {'status_code': 200}
```

Node.js

```
const { S3 } = require('aws-sdk');
const axios = require('axios').default;
const zlib = require('zlib');

exports.handler = async (event) => {
    const s3 = new S3();
```

```
    // Retrieve the operation context object from the event. This object indicates
 where the WriteGetObjectResponse request
    // should be delivered and has a presigned URL in 'inputS3Url' where we can
 download the requested object from.
    const { getObjectContext } = event;
    const { outputRoute, outputToken, inputS3Url } = getObjectContext;

    // Download the object from S3 and process it as a stream, because it might be a
 huge object and we don't want to
    // buffer it in memory. Note the use of 'await' because we want to wait for
 'writeGetObjectResponse' to finish
    // before we can exit the Lambda function.
    await axios({
        method: 'GET',
        url: inputS3Url,
        responseType: 'stream',
    }).then(
        // Gzip the stream.
        response => response.data.pipe(zlib.createGzip())
    ).then(
        // Finally send the gzip-ed stream back to the client.
        stream => s3.writeGetObjectResponse({
            RequestRoute: outputRoute,
            RequestToken: outputToken,
            Body: stream,
            ContentType: "text/plain",
            ContentEncoding: "gzip",
        }).promise()
    );

    // Gracefully exit the Lambda function.
    return { statusCode: 200 };
}
```

Note

Meskipun S3 Lambda Objek mengizinkan hingga 60 detik untuk mengirim respons lengkap kepada pemanggil melalui permintaan `WriteGetObjectResponse`, jumlah aktual waktu yang tersedia mungkin kurang. Misalnya, batas waktu fungsi Lambda Anda kemungkinan

kurang dari 60 detik. Dalam kasus lain, pemanggil mungkin memiliki batas waktu yang lebih ketat.

Agar penelepon asli menerima respons selain kode status HTTP 500 (Internal Server Error), panggilan `WriteGetObjectResponse` harus diselesaikan. Jika fungsi Lambda kembali, dengan pengecualian atau sebaliknya, sebelum operasi API `WriteGetObjectResponse` dipanggil, pemanggil asli menerima respons 500 (Internal Server Error). Pengecualian yang diberikan selama waktu yang diperlukan untuk menyelesaikan respons akan mengakibatkan respons terpotong kepada pemanggil. Jika fungsi Lambda menerima respons kode status HTTP 200 (OK) dari panggilan API `WriteGetObjectResponse`, lalu pemanggil asli telah mengirim permintaan lengkap. Respon fungsi Lambda, terlepas dari pengecualian diberikan atau tidak, diabaikan oleh S3 Lambda Objek.

Saat memanggil operasi API `WriteGetObjectResponse`, Amazon S3 membutuhkan rute dan meminta token dari konteks peristiwa. Untuk informasi selengkapnya, lihat Format konteks peristiwa dan penggunaan.

Parameter rute dan token permintaan diperlukan untuk menghubungkan respons `WriteGetObjectResult` dengan pemanggil asli. Meskipun selalu tepat untuk mencoba kembali respons 500 (Internal Server Error), karena token permintaan adalah token sekali pakai, upaya selanjutnya untuk menggunakannya mungkin menghasilkan respons kode status HTTP 400 (Bad Request). Meskipun panggilan ke `WriteGetObjectResponse` dengan rute dan permintaan token tidak perlu dibuat dari fungsi Lambda yang dipanggil, itu harus dibuat oleh identitas di akun yang sama. Panggilan itu juga harus diselesaikan sebelum fungsi Lambda selesai dieksekusi.

Bekerja dengan permintaan **`HeadObject`** di Lambda

Bagian ini mengasumsikan bahwa Titik Akses Lambda Objek Anda telah dikonfigurasi untuk memanggil fungsi Lambda `HeadObject`. Lambda akan menerima payload JSON yang berisi kunci bernama `headObjectContext`. Di dalam konteks, ada properti tunggal yang disebut `inputS3Url`, yang merupakan URL yang telah ditandatangani sebelumnya untuk titik akses pendukung untuk `HeadObject`.

URL yang telah ditandatangani sebelumnya akan menyertakan properti berikut jika ditentukan:

- `versionId` (di dalam parameter kueri)
- `requestPayer` (di dalam header `x-amz-request-payer`)
- `expectedBucketOwner` (di dalam header `x-amz-expected-bucket-owner`)

Properti lain tidak akan ditandatangani sebelumnya, dan karenanya tidak akan disertakan. Opsi yang tidak ditandatangani yang dikirim sebagai header dapat ditambahkan secara manual ke permintaan saat memanggil URL yang ditandatangani sebelumnya yang ditemukan di header `userRequest`. Opsi enkripsi di sisi server tidak didukung untuk `HeadObject`.

Untuk parameter URI sintaks permintaan, lihat `HeadObject` di Referensi API Amazon Simple Storage Service.

Contoh berikut ini menunjukkan muatan input Lambda JSON untuk `HeadObject`.

```
{
  "xAmzRequestId": "requestId",
  "**headObjectContext**": {
    "**inputS3Url**": "https://my-s3-ap-111122223333.s3-accesspoint.us-
east-1.amazonaws.com/example?X-Amz-Security-Token=<snip>"
  },
  "configuration": {
        "accessPointArn": "arn:aws:s3-object-lambda:us-
east-1:111122223333:accesspoint/example-object-lambda-ap",
        "supportingAccessPointArn": "arn:aws:s3:us-
east-1:111122223333:accesspoint/example-ap",
        "payload": "{}"
  },
  "userRequest": {
        "url": "https://object-lambda-111122223333.s3-object-lambda.us-
east-1.amazonaws.com/example",
        "headers": {
            "Host": "object-lambda-111122223333.s3-object-lambda.us-
east-1.amazonaws.com",
            "Accept-Encoding": "identity",
            "X-Amz-Content-SHA256": "e3b0c44298fc1example"
        }
   },
   "userIdentity": {
        "type": "AssumedRole",
        "principalId": "principalId",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/example",
        "accountId": "111122223333",
        "accessKeyId": "accessKeyId",
        "sessionContext": {
             "attributes": {
             "mfaAuthenticated": "false",
             "creationDate": "Wed Mar 10 23:41:52 UTC 2021"
```

```
        },
        "sessionIssuer": {
             "type": "Role",
             "principalId": "principalId",
             "arn": "arn:aws:iam::111122223333:role/Admin",
             "accountId": "111122223333",
             "userName": "Admin"
             }
        }
    },
  "protocolVersion": "1.00"
}
```

Fungsi Lambda Anda harus mengembalikan objek JSON yang berisi header dan nilai yang akan dikembalikan untuk panggilan `HeadObject`.

Contoh berikut ini menunjukkan struktur respons Lambda JSON untuk `HeadObject`.

```
{
    "statusCode": <number>; // Required
    "errorCode": <string>;
    "errorMessage": <string>;
    "headers": {
        "Accept-Ranges": <string>,
        "x-amz-archive-status": <string>,
        "x-amz-server-side-encryption-bucket-key-enabled": <boolean>,
        "Cache-Control": <string>,
        "Content-Disposition": <string>,
        "Content-Encoding": <string>,
        "Content-Language": <string>,
        "Content-Length": <number>, // Required
        "Content-Type": <string>,
        "x-amz-delete-marker": <boolean>,
        "ETag": <string>,
        "Expires": <string>,
        "x-amz-expiration": <string>,
        "Last-Modified": <string>,
        "x-amz-missing-meta": <number>,
        "x-amz-object-lock-mode": <string>,
        "x-amz-object-lock-legal-hold": <string>,
        "x-amz-object-lock-retain-until-date": <string>,
        "x-amz-mp-parts-count": <number>,
        "x-amz-replication-status": <string>,
```

```
        "x-amz-request-charged": <string>,
        "x-amz-restore": <string>,
        "x-amz-server-side-encryption": <string>,
        "x-amz-server-side-encryption-customer-algorithm": <string>,
        "x-amz-server-side-encryption-aws-kms-key-id": <string>,
        "x-amz-server-side-encryption-customer-key-MD5": <string>,
        "x-amz-storage-class": <string>,
        "x-amz-tagging-count": <number>,
        "x-amz-version-id": <string>,
        <x-amz-meta-headers>: <string>, // user-defined metadata
        "x-amz-meta-meta1": <string>, // example of the user-defined metadata header,
 it will need the x-amz-meta prefix
        "x-amz-meta-meta2": <string>
        ...
    };
}
```

Contoh berikut menunjukkan cara untuk menggunakan URL yang telah ditentukan sebelumnya untuk mengisi respons Anda dengan mengubah nilai header sesuai kebutuhan sebelum mengembalikan JSON.

Python

```
import requests

def lambda_handler(event, context):
    print(event)

    # Extract the presigned URL from the input.
    s3_url = event["headObjectContext"]["inputS3Url"]

    # Get the head of the object from S3.
    response = requests.head(s3_url)

    # Return the error to S3 Object Lambda (if applicable).
    if (response.status_code >= 400):
        return {
            "statusCode": response.status_code,
            "errorCode": "RequestFailure",
            "errorMessage": "Request to S3 failed"
    }
```

```
    # Store the headers in a dictionary.
    response_headers = dict(response.headers)

    # This obscures Content-Type in a transformation, it is optional to add
    response_headers["Content-Type"] = ""

    # Return the headers to S3 Object Lambda.
    return {
        "statusCode": response.status_code,
        "headers": response_headers
        }
```

Bekerja dengan permintaan **`ListObjects`** di Lambda

Bagian ini mengasumsikan bahwa Titik Akses Lambda Objek Anda telah dikonfigurasi untuk memanggil fungsi Lambda `ListObjects`. Lambda akan menerima payload JSON dengan objek baru bernama `listObjectsContext`. `listObjectsContext`berisi properti tunggal, `inputS3Url`, yang merupakan URL yang telah ditandatangani sebelumnya untuk titik akses pendukung untuk `ListObjects`.

Berbeda dengan `GetObject` dan `HeadObject`, URL yang telah ditandatangani sebelumnya akan menyertakan properti berikut jika ditentukan:

- Semua parameter kueri
- `requestPayer` (di dalam header `x-amz-request-payer`)
- `expectedBucketOwner` (di dalam header `x-amz-expected-bucket-owner`)

Untuk parameter URI sintaks permintaan, lihat `ListObjects` di Referensi API Amazon Simple Storage Service.

> **Important**
>
> Kami menyarankan Anda menggunakan versi yang lebih baru, ListObjectsV2, saat mengembangkan aplikasi. Untuk kompatibilitas mundur, Amazon S3 terus mendukung `ListObjects`.

Contoh berikut ini menunjukkan muatan input Lambda JSON untuk `ListObjects`.

```
{
    "xAmzRequestId": "requestId",
     "**listObjectsContext**": {
     "**inputS3Url**": "https://my-s3-ap-111122223333.s3-accesspoint.us-
east-1.amazonaws.com/?X-Amz-Security-Token=<snip>",
     },
    "configuration": {
        "accessPointArn": "arn:aws:s3-object-lambda:us-
east-1:111122223333:accesspoint/example-object-lambda-ap",
        "supportingAccessPointArn": "arn:aws:s3:us-
east-1:111122223333:accesspoint/example-ap",
        "payload": "{}"
    },
    "userRequest": {
        "url": "https://object-lambda-111122223333.s3-object-lambda.us-
east-1.amazonaws.com/example",
        "headers": {
            "Host": "object-lambda-111122223333.s3-object-lambda.us-
east-1.amazonaws.com",
            "Accept-Encoding": "identity",
            "X-Amz-Content-SHA256": "e3b0c44298fc1example"
        }
    },
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "principalId",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/example",
        "accountId": "111122223333",
        "accessKeyId": "accessKeyId",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "Wed Mar 10 23:41:52 UTC 2021"
            },
            "sessionIssuer": {
                "type": "Role",
                "principalId": "principalId",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            }
        }
    },
```

```
  "protocolVersion": "1.00"
}
```

Fungsi Lambda Anda harus mengembalikan objek JSON yang berisi kode status, daftar hasil XML, atau informasi kesalahan yang akan dikembalikan dari S3 Lambda Objek.

S3 Lambda Objek tidak memproses atau memvalidasi `listResultXml`, melainkan meneruskannya ke pemanggil `ListObjects`. Untuk `listBucketResult`, S3 Lambda Objek mengharapkan properti tertentu memiliki jenis tertentu dan akan memunculkan pengecualian jika tidak dapat menguraikannya. `listResultXml` dan `listBucketResult` tidak dapat diberikan secara bersamaan.

Contoh berikut menunjukkan cara menggunakan URL yang telah ditandatangani untuk memanggil Amazon S3, dan menggunakan hasilnya untuk mengisi respons, termasuk pemeriksaan kesalahan.

Python

```
import requests
import xmltodict

def lambda_handler(event, context):
    # Extract the presigned URL from the input.
    s3_url = event["listObjectsContext"]["inputS3Url"]


    # Get the head of the object from Amazon S3.
    response = requests.get(s3_url)

    # Return the error to S3 Object Lambda (if applicable).
    if (response.status_code >= 400):
        error = xmltodict.parse(response.content)
        return {
            "statusCode": response.status_code,
            "errorCode": error["Error"]["Code"],
            "errorMessage": error["Error"]["Message"]
        }

    # Store the XML result in a dict.
    response_dict = xmltodict.parse(response.content)

    # This obscures StorageClass in a transformation, it is optional to add
    for item in response_dict['ListBucketResult']['Contents']:
```

```
        item['StorageClass'] = ""

    # Convert back to XML.
    listResultXml = xmltodict.unparse(response_dict)

    # Create response with listResultXml.
    response_with_list_result_xml = {
        'statusCode': 200,
        'listResultXml': listResultXml
    }

    # Create response with listBucketResult.
    response_dict['ListBucketResult'] =
 sanitize_response_dict(response_dict['ListBucketResult'])
    response_with_list_bucket_result = {
        'statusCode': 200,
        'listBucketResult': response_dict['ListBucketResult']
    }

    # Return the list to S3 Object Lambda.
    # Can return response_with_list_result_xml or response_with_list_bucket_result
    return response_with_list_result_xml

# Converting the response_dict's key to correct casing
def sanitize_response_dict(response_dict: dict):
    new_response_dict = dict()
    for key, value in response_dict.items():
        new_key = key[0].lower() + key[1:] if key != "ID" else 'id'
        if type(value) == list:
            newlist = []
            for element in value:
                if type(element) == type(dict()):
                    element = sanitize_response_dict(element)
                newlist.append(element)
            value = newlist
        elif type(value) == dict:
            value = sanitize_response_dict(value)
        new_response_dict[new_key] = value
    return new_response_dict
```

Contoh berikut ini menunjukkan struktur respons Lambda JSON untuk `ListObjects`.

```
{
  "statusCode": <number>; // Required
  "errorCode": <string>;
  "errorMessage": <string>;
  "listResultXml": <string>; // This can also be Error XML string in case S3 returned
 error response when calling the pre-signed URL

  "listBucketResult": {  // listBucketResult can be provided instead of listResultXml,
 however they can not both be provided in the JSON response
        "name": <string>,  // Required for 'listBucketResult'
        "prefix": <string>,
        "marker": <string>,
        "nextMarker": <string>,
        "maxKeys": <int>,   // Required for 'listBucketResult'
        "delimiter": <string>,
        "encodingType": <string>
        "isTruncated": <boolean>,  // Required for 'listBucketResult'
        "contents": [  {
            "key": <string>,  // Required for 'content'
            "lastModified": <string>,
            "eTag": <string>,
            "checksumAlgorithm": <string>,   // CRC32,  CRC32C,  SHA1,  SHA256
            "size": <int>,   // Required for 'content'
            "owner": {
                "displayName": <string>,  // Required for 'owner'
                "id": <string>,  // Required for 'owner'
            },
            "storageClass": <string>
            },
        ...
        ],
        "commonPrefixes": [  {
            "prefix": <string>   // Required for 'commonPrefix'
        },
        ...
        ],
    }
}
```

Bekerja dengan permintaan **`ListObjectsV2`** di Lambda

Bagian ini mengasumsikan bahwa Titik Akses Lambda Objek Anda telah dikonfigurasi untuk memanggil fungsi Lambda `ListObjectsV2`. Lambda akan menerima payload JSON dengan

objek baru bernama `listObjectsV2Context`. `listObjectsV2Context`berisi properti tunggal, `inputS3Url`, yang merupakan URL yang telah ditandatangani sebelumnya untuk titik akses pendukung untuk `ListObjectsV2`.

Berbeda dengan `GetObject` dan `HeadObject`, URL yang telah ditandatangani sebelumnya akan menyertakan properti berikut jika ditentukan:

- Semua parameter kueri
- `requestPayer` (di dalam header `x-amz-request-payer`)
- `expectedBucketOwner` (di dalam header `x-amz-expected-bucket-owner`)

Untuk parameter URI sintaks permintaan, lihat `ListObjectsV2` di Referensi API Amazon Simple Storage Service.

Contoh berikut ini menunjukkan muatan input Lambda JSON untuk `ListObjectsV2`.

```
{
    "xAmzRequestId": "requestId",
     "**listObjectsV2Context**": {
     "**inputS3Url**": "https://my-s3-ap-111122223333.s3-accesspoint.us-
east-1.amazonaws.com/?list-type=2&X-Amz-Security-Token=<snip>",
     },
    "configuration": {
        "accessPointArn": "arn:aws:s3-object-lambda:us-
east-1:111122223333:accesspoint/example-object-lambda-ap",
        "supportingAccessPointArn": "arn:aws:s3:us-
east-1:111122223333:accesspoint/example-ap",
        "payload": "{}"
    },
    "userRequest": {
        "url": "https://object-lambda-111122223333.s3-object-lambda.us-
east-1.amazonaws.com/example",
        "headers": {
            "Host": "object-lambda-111122223333.s3-object-lambda.us-
east-1.amazonaws.com",
            "Accept-Encoding": "identity",
            "X-Amz-Content-SHA256": "e3b0c44298fc1example"
        }
    },
    "userIdentity": {
        "type": "AssumedRole",
```

```
        "principalId": "principalId",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/example",
        "accountId": "111122223333",
        "accessKeyId": "accessKeyId",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "Wed Mar 10 23:41:52 UTC 2021"
            },
            "sessionIssuer": {
                "type": "Role",
                "principalId": "principalId",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            }
        }
    },
  "protocolVersion": "1.00"
}
```

Fungsi Lambda Anda harus mengembalikan objek JSON yang berisi kode status, daftar hasil XML, atau informasi kesalahan yang akan dikembalikan dari S3 Lambda Objek.

S3 Lambda Objek tidak memproses atau memvalidasi `listResultXml`, melainkan meneruskannya ke pemanggil `ListObjectsV2`. Untuk `listBucketResult`, S3 Lambda Objek mengharapkan properti tertentu memiliki jenis tertentu dan akan memunculkan pengecualian jika tidak dapat menguraikannya. `listResultXml` dan `listBucketResult` tidak dapat diberikan secara bersamaan.

Contoh berikut menunjukkan cara menggunakan URL yang telah ditandatangani untuk memanggil Amazon S3, dan menggunakan hasilnya untuk mengisi respons, termasuk pemeriksaan kesalahan.

Python

```
import requests
import xmltodict

def lambda_handler(event, context):
    # Extract the presigned URL from the input.
    s3_url = event["listObjectsV2Context"]["inputS3Url"]
```

```
    # Get the head of the object from Amazon S3.
    response = requests.get(s3_url)

    # Return the error to S3 Object Lambda (if applicable).
    if (response.status_code >= 400):
        error = xmltodict.parse(response.content)
        return {
            "statusCode": response.status_code,
            "errorCode": error["Error"]["Code"],
            "errorMessage": error["Error"]["Message"]
        }

    # Store the XML result in a dict.
    response_dict = xmltodict.parse(response.content)

    # This obscures StorageClass in a transformation, it is optional to add
    for item in response_dict['ListBucketResult']['Contents']:
        item['StorageClass'] = ""

    # Convert back to XML.
    listResultXml = xmltodict.unparse(response_dict)

    # Create response with listResultXml.
    response_with_list_result_xml = {
        'statusCode': 200,
        'listResultXml': listResultXml
    }

    # Create response with listBucketResult.
    response_dict['ListBucketResult'] =
 sanitize_response_dict(response_dict['ListBucketResult'])
    response_with_list_bucket_result = {
        'statusCode': 200,
        'listBucketResult': response_dict['ListBucketResult']
    }

    # Return the list to S3 Object Lambda.
    # Can return response_with_list_result_xml or response_with_list_bucket_result
    return response_with_list_result_xml

# Converting the response_dict's key to correct casing
def sanitize_response_dict(response_dict: dict):
    new_response_dict = dict()
```

```
    for key, value in response_dict.items():
        new_key = key[0].lower() + key[1:] if key != "ID" else 'id'
        if type(value) == list:
            newlist = []
            for element in value:
                if type(element) == type(dict()):
                    element = sanitize_response_dict(element)
                newlist.append(element)
            value = newlist
        elif type(value) == dict:
            value = sanitize_response_dict(value)
        new_response_dict[new_key] = value
    return new_response_dict
```

Contoh berikut ini menunjukkan struktur respons Lambda JSON untuk `ListObjectsV2`.

```
{
    "statusCode": <number>; // Required
    "errorCode": <string>;
    "errorMessage": <string>;
    "listResultXml": <string>; // This can also be Error XML string in case S3 returned
 error response when calling the pre-signed URL

    "listBucketResult": {  // listBucketResult can be provided instead of
 listResultXml, however they can not both be provided in the JSON response
        "name": <string>, // Required for 'listBucketResult'
        "prefix": <string>,
        "startAfter": <string>,
        "continuationToken": <string>,
        "nextContinuationToken": <string>,
        "keyCount": <int>, // Required for 'listBucketResult'
        "maxKeys": <int>, // Required for 'listBucketResult'
        "delimiter": <string>,
        "encodingType": <string>
        "isTruncated": <boolean>, // Required for 'listBucketResult'
        "contents": [ {
            "key": <string>, // Required for 'content'
            "lastModified": <string>,
            "eTag": <string>,
            "checksumAlgorithm": <string>, // CRC32, CRC32C, SHA1, SHA256
            "size": <int>, // Required for 'content'
            "owner": {
```

```
                "displayName": <string>, // Required for 'owner'
                "id": <string>, // Required for 'owner'
            },
            "storageClass": <string>
            },
            ...
        ],
        "commonPrefixes": [ {
            "prefix": <string> // Required for 'commonPrefix'
            },
        ...
        ],
    }
}
```

Format konteks peristiwa dan penggunaan

Amazon S3 Object Lambda menyediakan konteks tentang permintaan yang sedang dibuat jika diteruskan ke fungsi Anda. AWS Lambda Berikut adalah contoh permintaannya. Deskripsi bidang disertakan setelah contoh.

```
{
    "xAmzRequestId": "requestId",
    "getObjectContext": {
        "inputS3Url": "https://my-s3-ap-111122223333.s3-accesspoint.us-
east-1.amazonaws.com/example?X-Amz-Security-Token=<snip>",
        "outputRoute": "io-use1-001",
        "outputToken": "OutputToken"
    },
    "configuration": {
        "accessPointArn": "arn:aws:s3-object-lambda:us-
east-1:111122223333:accesspoint/example-object-lambda-ap",
        "supportingAccessPointArn": "arn:aws:s3:us-
east-1:111122223333:accesspoint/example-ap",
        "payload": "{}"
    },
    "userRequest": {
        "url": "https://object-lambda-111122223333.s3-object-lambda.us-
east-1.amazonaws.com/example",
        "headers": {
            "Host": "object-lambda-111122223333.s3-object-lambda.us-
east-1.amazonaws.com",
            "Accept-Encoding": "identity",
```

```
            "X-Amz-Content-SHA256": "e3b0c44298fc1example"
        }
    },
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "principalId",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/example",
        "accountId": "111122223333",
        "accessKeyId": "accessKeyId",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "Wed Mar 10 23:41:52 UTC 2021"
            },
            "sessionIssuer": {
                "type": "Role",
                "principalId": "principalId",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            }
        }
    },
    "protocolVersion": "1.00"
}
```

Kolom permintaan disertakan dalam permintaan berikut:

- `xAmzRequestId`–ID permintaan Amazon S3 untuk permintaan ini. Kami merekomendasikan agar Anda mencatat nilai ini untuk membantu dengan debugging.
- `getObjectContext`–Detail input dan output untuk koneksi ke Amazon S3 dan S3 Lambda Objek.
  - `inputS3Url`–URL yang telah ditandatangani sebelumnya, yang dapat digunakan untuk mengambil objek asli dari Amazon S3. URL ditandatangani dengan menggunakan identitas pemanggil asli, dan izin pengguna tersebut akan berlaku saat URL tersebut digunakan. Jika ada header ditandatangani di URL, fungsi Lambda harus menyertakan header ini dalam panggilan ke Amazon S3, kecuali untuk header `Host`.
  - `outputRoute`–Token perutean yang ditambahkan ke URL S3 Lambda Objek ketika fungsi Lambda memanggil `WriteGetObjectResponse`.
  - `outputToken`–Token buram yang digunakan oleh S3 Lambda Objek untuk mencocokkan panggilan `WriteGetObjectResponse` dengan pemanggil asli.

- `configuration`–Informasi konfigurasi tentang Titik Akses Lambda Objek.
  - `accessPointArn`–Amazon Resource Name (ARN) dari Titik Akses Lambda Objek yang menerima permintaan ini.
  - `supportingAccessPointArn`–ARN dari titik akses pendukung yang ditentukan dalam konfigurasi Titik Akses Lambda Objek.
  - `payload`–Data kustom yang diterapkan pada konfigurasi Titik Akses Lambda Objek. S3 Lambda Objek memperlakukan data ini sebagai string buram, sehingga mungkin perlu diterjemahkan sebelum digunakan.
- `userRequest`–Informasi tentang panggilan asli untuk S3 Lambda Objek.
  - `url`–URL permintaan yang didekodekan seperti yang diterima oleh S3 Lambda Objek, tidak termasuk parameter kueri terkait otorisasi.
  - `headers`–Peta string ke string yang berisi header HTTP serta nilai dari panggilan aslinya, tidak termasuk header terkait otorisasi. Jika header yang sama muncul beberapa kali, nilai dari setiap instans header yang sama digabungkan ke dalam daftar yang dibatasi koma. Kasus header asli dipertahankan dalam peta ini.
- `userIdentity`–Detail tentang identitas yang melakukan panggilan ke S3 Lambda Objek. Untuk informasi lebih lanjut, lihat Peristiwa Pencatatan Data untuk Pelacakan dalam AWS CloudTrail Panduan Pengguna.
  - `type`–Jenis identitas.
  - `accountId`— Akun AWS yang menjadi milik identitas.
  - `userName`–Nama ramah dari identitas yang melakukan panggilan.
  - `principalId`–Pengidentifikasi unik untuk identitas yang melakukan panggilan.
  - `arn`–ARN pengguna utama yang melakukan panggilan. Bagian terakhir ARN berisi pengguna atau peran yang melakukan panggilan.
  - `sessionContext`–Jika permintaan dibuat dengan kredensial keamanan sementara, elemen ini memberikan informasi tentang sesi yang dibuat untuk kredensial tersebut.
  - `invokedBy`— Nama Layanan AWS yang membuat permintaan, seperti Amazon EC2 Auto Scaling atau. AWS Elastic Beanstalk
  - `sessionIssuer`–Jika permintaan dibuat dengan kredensial keamanan sementara, elemen ini memberikan informasi tentang bagaimana kredensial tersebut diperoleh.
- `protocolVersion`–ID Versi dari konteks yang disediakan. Format bidang ini adalah `{Major Version}.{Minor Version}`. Nomor versi minor selalu berupa angka dua digit. Penghapusan atau perubahan apa pun pada semantik suatu bidang memerlukan perubahan versi utama.

dan memerlukan keikutsertaan aktif. Amazon S3 dapat menambahkan kolom baru kapan saja, sehingga Anda mungkin mengalami peningkatan versi kecil. Karena sifat peluncuran perangkat lunak, Anda mungkin melihat beberapa versi minor digunakan sekaligus.

Bekerja dengan header Range dan partNumber

Saat bekerja dengan objek besar di Lambda Objek Amazon S3, Anda dapat menggunakan header HTTP `Range` untuk mengunduh rentang byte tertentu dari sebuah objek. Untuk mengambil rentang byte berbeda dari dalam objek yang sama, Anda dapat menggunakan koneksi bersamaan ke Amazon S3. Anda juga dapat menentukan parameter `partNumber` (integer antara 1 dan 10.000), yang melakukan permintaan berkisar untuk bagian objek tertentu.

Karena ada beberapa cara yang mungkin Anda inginkan untuk menangani permintaan yang menyertakan parameter `Range` atau `partNumber`, S3 Lambda Objek tidak menerapkan parameter ini ke objek yang ditransformasi. Sebagai gantinya, AWS Lambda fungsi Anda harus mengimplementasikan fungsi ini sesuai kebutuhan untuk aplikasi Anda.

Untuk menggunakan parameter `Range` dan `partNumber` dengan S3 Lambda Objek, Anda melakukan hal berikut ini:

- Aktifkan parameter ini di konfigurasi Titik Akses Lambda Objek.
- Tulis fungsi Lambda yang dapat menangani permintaan yang menyertakan parameter ini.

Langkah-langkah berikut menjelaskan cara untuk mencapai hal ini.

Langkah 1: Mengonfigurasi Titik Akses Lambda Objek Anda

Secara default, Titik Akses Objek Lambda merespons dengan kesalahan kode status HTTP 501 (Tidak Diimplementasikan) terhadap permintaan `GetObject` atau `HeadObject` yang berisi parameter `Range` atau `partNumber`, baik di header atau parameter kueri.

Untuk mengaktifkan Titik Akses Lambda Objek untuk menerima permintaan tersebut, Anda harus menyertakan `GetObject-Range`, `GetObject-PartNumber`, `HeadObject-Range`, atau `HeadObject-PartNumber` dalam bagian `AllowedFeatures` konfigurasi Titik Akses Lambda Objek Anda. Untuk informasi selengkapnya tentang memperbarui konfigurasi Titik Akses Lambda Objek, lihat Membuat Titik Akses Objek Lambda.

Langkah 2: Menerapkan penanganan **`Range`** atau **`partNumber`** di dalam fungsi Lambda Anda

Saat Titik Akses Lambda Object Anda menginvokasi fungsi Lambda Anda dengan rentang permintaan `GetObject` atau `HeadObject`, parameter `Range` atau `partNumber` disertakan dalam konteks peristiwa. Lokasi parameter dalam konteks peristiwa tergantung pada parameter mana yang digunakan dan bagaimana itu dimasukkan dalam permintaan asli ke Titik Akses Lambda Object, seperti yang dijelaskan dalam tabel berikut.

Parameter	Lokasi konteks peristiwa
`Range` (header)	`userRequest.headers.Range`
`Range` (parameter kueri)	`userRequest.url` (parameter kueri `Range`)
`partNumber`	`userRequest.url` (parameter kueri `partNumber`)

> **Important**
>
> URL presigned yang disediakan untuk Titik Akses Lambda Objek Anda tidak berisi parameter `Range` atau `partNumber` dari permintaan asli. Lihat opsi berikut tentang cara menangani parameter ini dalam AWS Lambda fungsi Anda.

Setelah Anda mengekstrak nilai `Range` atau `partNumber`, Anda dapat mengambil salah satu pendekatan berikut ini, berdasarkan kebutuhan aplikasi Anda:

A. Petakan **`Range`** yang diminta atau **`partNumber`** ke objek yang diubah (disarankan).

Cara yang paling dapat diandalkan untuk menangani permintaan `Range` atau `partNumber` adalah dengan melakukan hal berikut ini:

- Mengambil objek lengkap dari Amazon S3.
- Mengubah objek tersebut.
- Terapkan parameter `Range` atau `partNumber` yang diminta ke objek yang diubah.

Untuk melakukan ini, gunakan URL yang telah ditandatangani sebelumnya untuk mengambil seluruh objek dari Amazon S3, dan kemudian memproses objek tersebut sesuai kebutuhan. Untuk

contoh fungsi Lambda yang memproses `Range` parameter dengan cara ini, lihat sampel ini di repositori AWS Sampel GitHub .

B. Petakan **`Range`** yang diminta ke URL yang telah ditandatangani sebelumnya.

Dalam beberapa kasus, fungsi Lambda Anda dapat memetakan permintaan `Range` langsung ke URL yang telah ditentukan sebelumnya untuk mengambil hanya sebagian objek dari Amazon S3. Pendekatan ini hanya sesuai jika transformasi Anda memenuhi kedua kriteria berikut:

1. Fungsi transformasi Anda dapat diterapkan ke sebagian rentang objek.
2. Menerapkan parameter `Range` sebelum atau sesudah fungsi transformasi menghasilkan objek yang ditransformasikan yang sama.

Misalnya, fungsi transformasi yang mengubah semua karakter dalam objek berkode ASCII menjadi huruf besar memenuhi kedua kriteria sebelumnya. Transformasi dapat diterapkan pada bagian dari suatu objek, dan menerapkan parameter `Range` sebelum transformasi mencapai hasil yang sama seperti menerapkannya setelah transformasi.

Sebaliknya, fungsi yang membalikkan karakter dalam objek yang dikodekan ASCII tidak memenuhi kriteria ini. Fungsi seperti itu memenuhi kriteria 1, karena dapat diterapkan pada rentang objek sebagian. Namun, itu tidak memenuhi kriteria 2, karena menerapkan parameter `Range` sebelum transformasi mencapai hasil yang berbeda daripada menerapkan parameter setelah transformasi.

Pertimbangkan permintaan untuk menerapkan fungsi ke tiga karakter pertama dari suatu objek dengan isinya `abcdefg`. Menerapkan parameter `Range` sebelum transformasi hanya mengambil `abc` dan kemudian membalikkan data, mengembalikan `cba`. Tetapi jika parameter diterapkan setelah transformasi, fungsi mengambil seluruh objek, membalikkannya, lalu menerapkan parameter `Range`, mengembalikan `gfe`. Karena hasil ini berbeda, fungsi ini tidak boleh menerapkan parameter `Range` saat mengambil objek dari Amazon S3. Sebaliknya, itu harus mengambil seluruh objek, melakukan transformasi, dan hanya kemudian menerapkan parameter `Range`.

> ⚠ Warning
>
> Dalam banyak kasus, menerapkan parameter `Range` ke URL yang telah ditandatangani sebelumnya akan menghasilkan perilaku tak terduga oleh fungsi Lambda atau klien yang meminta. Kecuali Anda yakin bahwa aplikasi Anda akan berfungsi dengan baik ketika hanya mengambil sebagian objek dari Amazon S3, kami menyarankan Anda mengambil dan mengubah objek penuh seperti yang dijelaskan sebelumnya dalam pendekatan A.

Jika aplikasi Anda memenuhi kriteria yang dijelaskan sebelumnya dalam pendekatan B, Anda dapat menyederhanakan AWS Lambda fungsi Anda dengan mengambil hanya rentang objek yang diminta dan kemudian menjalankan transformasi Anda pada rentang tersebut.

Contoh kode Java berikut mendemonstrasikan cara untuk melakukan hal berikut ini:

- Ambil header `Range` dari permintaan `GetObject`.
- Tambahkan header `Range` ke URL yang telah ditandatangani sebelumnya, yang dapat digunakan oleh Lambda untuk mengambil rentang yang diminta dari Amazon S3.

```
private HttpRequest.Builder applyRangeHeader(ObjectLambdaEvent event,
 HttpRequest.Builder presignedRequest) {
    var header = event.getUserRequest().getHeaders().entrySet().stream()
            .filter(e -> e.getKey().toLowerCase(Locale.ROOT).equals("range"))
            .findFirst();

    // Add check in the query string itself.
    header.ifPresent(entry -> presignedRequest.header(entry.getKey(),
 entry.getValue()));
    return presignedRequest;
}
```

Menggunakan fungsi Lambda yang AWS dibangun

AWS menyediakan beberapa AWS Lambda fungsi bawaan yang dapat Anda gunakan dengan Amazon S3 Object Lambda untuk mendeteksi dan menyunting informasi identitas pribadi (PII) dan mendekompresi objek S3. Fungsi Lambda ini tersedia di AWS Serverless Application Repository. Anda dapat memilih fungsi-fungsi ini melalui AWS Management Console ketika Anda membuat Titik Akses Lambda Objek Anda.

Untuk informasi selengkapnya tentang cara menerapkan aplikasi tanpa server dari aplikasi AWS Serverless Application Repository, lihat Menerapkan Aplikasi di Panduan Pengembang.AWS Serverless Application Repository

> **Note**
>
> Contoh berikut hanya dapat digunakan dengan permintaan `GetObject`.

Contoh 1: Kontrol akses PII

Fungsi Lambda ini menggunakan Amazon Comprehend, layanan pemrosesan bahasa alami (NLP) yang menggunakan machine learning untuk menemukan wawasan dan hubungan dalam teks. Fungsi ini secara otomatis mendeteksi informasi pengenal pribadi (PII), seperti nama, alamat, tanggal, nomor kartu kredit, dan nomor jaminan sosial dalam dokumen di bucket Amazon S3 Anda. Jika Anda memiliki dokumen di bucket Anda yang menyertakan PII, Anda dapat mengonfigurasi fungsi Kontrol Akses PII untuk mendeteksi jenis entitas PII ini dan membatasi akses ke pengguna yang tidak sah.

Untuk memulai, terapkan fungsi Lambda berikut di akun Anda dan tambahkan Amazon Resource Name (ARN) untuk fungsi tersebut ke konfigurasi Titik Akses Lambda Objek Anda.

Berikut adalah contoh ARN untuk fungsi ini:

```
arn:aws:serverlessrepo:us-east-1:111122223333:applications/
ComprehendPiiAccessControlS3ObjectLambda
```

Anda dapat menambahkan atau melihat fungsi ini pada AWS Management Console dengan menggunakan AWS Serverless Application Repository link berikut: ComprehendPiiAccessControlS3 ObjectLambda.

Untuk melihat fungsi ini aktif GitHub, lihat Amazon Comprehend S3 Object Lambda.

Contoh 2: Penyamaran PII

Fungsi Lambda ini menggunakan Amazon Comprehend, layanan pemrosesan bahasa alami (NLP) yang menggunakan machine learning untuk menemukan wawasan dan hubungan dalam teks. Fungsi ini secara otomatis menyunting informasi pengenal pribadi (PII), seperti nama, alamat, tanggal, nomor kartu kredit, dan nomor jaminan sosial dari dokumen di bucket Amazon S3 Anda.

Jika Anda memiliki dokumen di bucket Anda yang menyertakan informasi seperti nomor kartu kredit atau informasi rekening bank, Anda dapat mengonfigurasi fungsi Lambda PII Redaction S3 Object untuk mendeteksi PII dan kemudian mengembalikan salinan dokumen ini di mana jenis entitas PII disunting.

Untuk memulai, deploy fungsi Lambda berikut ini di akun Anda, dan tambahkan ARN untuk fungsi tersebut ke konfigurasi Titik Akses Lambda Objek Anda.

Berikut adalah contoh ARN untuk fungsi ini:

```
arn:aws:serverlessrepo:us-east-1:111122223333::applications/
ComprehendPiiRedactionS3ObjectLambda
```

Anda dapat menambahkan atau melihat fungsi ini pada AWS Management Console dengan menggunakan AWS Serverless Application Repository link berikut: ComprehendPiiRedactionS3 ObjectLambda.

Untuk melihat fungsi ini aktif GitHub, lihat Amazon Comprehend S3 Object Lambda.

Untuk mempelajari tentang end-to-end prosedur lengkap untuk beberapa tugas Lambda Objek S3 dalam redaksi PII, lihat. Tutorial: Mendeteksi dan menyunting data PII dengan Objek S3 Lambda dan Amazon Comprehend

Contoh 3: Dekompresi

Fungsi Lambda `S3ObjectLambdaDecompression` dapat mendekompresi objek yang disimpan di Amazon S3 dalam salah satu dari enam format file terkompresi: `bzip2`, `gzip`, `snappy`, `zlib`, `zstandard`, dan `ZIP`.

Untuk memulai, deploy fungsi Lambda berikut ini di akun Anda, dan tambahkan ARN untuk fungsi tersebut ke konfigurasi Titik Akses Lambda Objek Anda.

Berikut adalah contoh ARN untuk fungsi ini:

```
arn:aws:serverlessrepo:us-east-1:111122223333::applications/S3ObjectLambdaDecompression
```

Anda dapat menambahkan atau melihat fungsi ini pada AWS Management Console dengan menggunakan AWS Serverless Application Repository link berikut: S3 ObjectLambdaDecompression.

Untuk melihat fungsi ini aktif GitHub, lihat Dekompresi Lambda Objek S3.

Praktik terbaik dan pedoman untuk S3 Lambda Objek

Saat menggunakan S3 Lambda Objek, ikuti praktik terbaik dan pedoman ini untuk mengoptimalkan operasi dan performanya.

Topik

- Bekerja dengan S3 Lambda Objek

- Layanan AWS digunakan sehubungan dengan S3 Object Lambda
- Header Range dan partNumber
- Mengubah expiry-date
- Bekerja dengan AWS CLI dan AWS SDKs

Bekerja dengan S3 Lambda Objek

S3 Lambda Objek hanya mendukung pemrosesan permintaan `GET`, `LIST`, dan `HEAD`. Permintaan lain apa pun tidak memanggil AWS Lambda dan sebagai gantinya mengembalikan respons API standar yang tidak diubah. Anda dapat membuat maksimum 1.000 Titik Akses Objek Lambda Akun AWS per Wilayah. AWS Lambda Fungsi yang Anda gunakan harus sama Akun AWS dan Region sebagai Object Lambda Access Point.

S3 Lambda Objek memungkinkan hingga 60 detik untuk mengalirkan respons lengkap ke pemanggilnya. Fungsi Anda juga tunduk pada kuota AWS Lambda default. Untuk informasi selengkapnya, lihat Kuota Lambda dalam Panduan Pengembang AWS Lambda .

Saat S3 Lambda Objek menginvokasi fungsi Lambda yang Anda tentukan, Anda bertanggung jawab untuk memastikan bahwa data apa pun yang ditimpa atau dihapus dari Amazon S3 oleh fungsi atau aplikasi Lambda yang Anda tentukan memang tepat dan benar.

Anda hanya dapat menggunakan S3 Lambda Objek untuk melakukan operasi pada objek. Anda tidak dapat menggunakan Lambda Objek S3 untuk melakukan operasi Amazon S3 lainnya, seperti memodifikasi atau menghapus bucket. Untuk mengetahui daftar lengkap pengoperasian S3 yang mendukung titik akses, lihat Kompatibilitas titik akses dengan operasi S3.

Selain daftar ini, Titik Akses Lambda Objek tidak mendukung `POST Object`, `CopyObject` (sebagai sumber), dan operasi API `SelectObjectContent`.

Layanan AWS digunakan sehubungan dengan S3 Object Lambda

S3 Object Lambda menghubungkan Amazon S3 AWS Lambda, dan secara opsional, pilihan Layanan AWS lain yang Anda pilih untuk mengirimkan objek yang relevan dengan aplikasi yang meminta. Semua yang Layanan AWS digunakan dengan S3 Object Lambda diatur oleh Perjanjian Tingkat Layanan masing-masing (). SLAs Misalnya, jika ada yang Layanan AWS tidak memenuhi Komitmen Layanannya, Anda berhak menerima Kredit Layanan, sebagaimana didokumentasikan dalam SLA layanan.

Header **`Range`** dan **`partNumber`**

Saat bekerja dengan objek besar, Anda dapat menggunakan header HTTP `Range` untuk mengunduh rentang byte tertentu dari suatu objek. Saat Anda menggunakan header `Range`, permintaan Anda hanya mengambil bagian tertentu dari objek. Anda juga dapat menggunakan header `partNumber` untuk melakukan permintaan berkisar untuk bagian tertentu dari objek.

Untuk informasi selengkapnya, lihat Bekerja dengan header Range dan partNumber.

Mengubah **`expiry-date`**

Anda dapat membuka atau mengunduh objek yang diubah dari Titik Akses Objek Lambda Anda di file. AWS Management Console Benda-benda ini harus berupa non-kedaluwarsa. Jika fungsi Lambda Anda mengubah `expiry-date` dari objek Anda, Anda mungkin melihat objek kedaluwarsa yang tidak dapat dibuka atau diunduh. Perilaku ini hanya berlaku untuk objek yang dipulihkan S3 Glacier Fleksibel Retrieval dan S3 Glacier Deep Archive.

Bekerja dengan AWS CLI dan AWS SDKs

AWS Command Line Interface (AWS CLI) Subperintah S3 (`cp`,`mv`, dan`sync`) dan penggunaan AWS SDK untuk Java `TransferManager` kelas tidak didukung untuk digunakan dengan S3 Object Lambda.

Tutorial S3 Lambda Objek

Tutorial berikut menyajikan end-to-end prosedur lengkap untuk beberapa tugas Lambda Objek S3.

Dengan S3 Object Lambda, Anda dapat menambahkan kode Anda sendiri untuk memproses data yang diambil dari S3 sebelum mengembalikannya ke aplikasi. Masing-masing tutorial berikut akan memodifikasi data karena diambil dari Amazon S3, tanpa mengubah objek yang ada atau mempertahankan beberapa salinan data. Tutorial pertama akan membahas cara menambahkan AWS Lambda fungsi ke permintaan S3 GET untuk memodifikasi objek yang diambil dari S3. Tutorial kedua menunjukkan cara menggunakan fungsi Lambda bawaan yang didukung oleh Amazon Comprehend untuk melindungi informasi identitas pribadi (PII) yang diambil dari S3 sebelum mengembalikannya ke aplikasi. Tutorial ketiga menggunakan S3 Object Lambda untuk menambahkan tanda air ke gambar karena diambil dari Amazon S3.

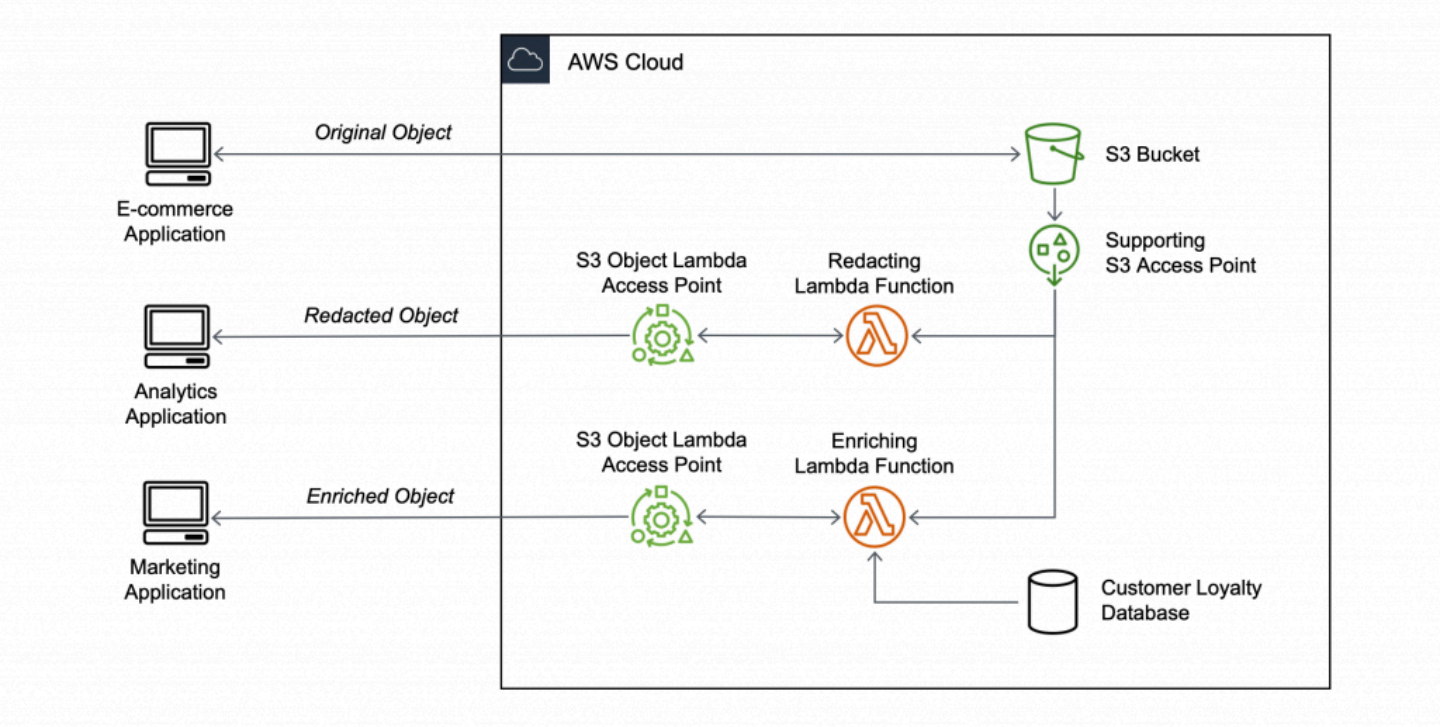
- Tutorial: Mengubah data untuk aplikasi Anda dengan S3 Object Lambda
- Tutorial: Mendeteksi dan menyunting data PII dengan Objek S3 Lambda dan Amazon Comprehend

- Tutorial: Menggunakan S3 Lambda Objek untuk menandai gambar secara dinamis saat gambarnya diambil

Tutorial: Mengubah data untuk aplikasi Anda dengan S3 Object Lambda

Saat Anda menyimpan data di Amazon S3, Anda dapat dengan mudah membagikannya untuk digunakan oleh beberapa aplikasi. Namun, setiap aplikasi mungkin memiliki persyaratan format data yang unik, dan mungkin memerlukan modifikasi atau pemrosesan data Anda untuk kasus penggunaan tertentu. Misalnya, kumpulan data yang dibuat oleh aplikasi e-niaga mungkin termasuk informasi identitas pribadi (PII). Ketika data yang sama diproses untuk analitik, PII ini tidak diperlukan dan harus disunting. Namun, jika dataset yang sama digunakan untuk kampanye pemasaran, Anda mungkin perlu memperkaya data dengan detail tambahan, seperti informasi dari database loyalitas pelanggan.

Dengan S3 Object Lambda, Anda dapat menambahkan kode Anda sendiri untuk memproses data yang diambil dari S3 sebelum mengembalikannya ke aplikasi. Secara khusus, Anda dapat mengonfigurasi AWS Lambda fungsi dan melampirkannya ke Titik Akses Lambda Objek S3. Saat aplikasi mengirimkan permintaan GET S3 standar melalui Titik Akses Lambda Objek S3, fungsi Lambda yang ditentukan dipanggil untuk memproses data apa pun yang diambil dari sumber data yang mendasarinya melalui jalur akses S3 pendukung. Kemudian, Titik Akses Lambda Objek S3 mengembalikan hasil yang diubah kembali ke aplikasi. Anda dapat membuat dan menjalankan fungsi Lambda kustom Anda sendiri, menyesuaikan transformasi data Lambda Objek S3 dengan kasus penggunaan spesifik Anda, semua tanpa perubahan yang diperlukan untuk aplikasi Anda.



Tujuan

Dalam tutorial ini, Anda mempelajari cara menambahkan kode kustom ke permintaan GET S3 standar untuk memodifikasi objek yang diminta diambil dari S3 sehingga objek sesuai dengan kebutuhan klien atau aplikasi yang meminta. Secara khusus, Anda mempelajari cara mengubah semua teks dalam objek asli yang disimpan dalam bucket S3 menjadi huruf besar melalui S3 Object Lambda.

> **Note**
>
> Tutorial ini menggunakan kode Python untuk mengubah data, misalnya menggunakan lainnya AWS SDKs lihat Transform data untuk aplikasi Anda dengan S3 Object Lambda di SDK Code Examples Library. AWS

Topik

- Prasyarat
- Langkah 1: Buat ember S3
- Langkah 2: Unggah file ke bucket S3
- Langkah 3: Buat titik akses S3

- Langkah 4: Buat fungsi Lambda
- Langkah 5: Konfigurasikan kebijakan IAM untuk peran eksekusi fungsi Lambda Anda
- Langkah 6: Buat Titik Akses Lambda Objek S3
- Langkah 7: Lihat data yang diubah
- Langkah 8: Membersihkan
- Langkah selanjutnya

Prasyarat

Sebelum Anda memulai tutorial ini, Anda harus memiliki Akun AWS yang dapat Anda masuki sebagai pengguna AWS Identity and Access Management (IAM) dengan izin yang benar. Anda juga harus menginstal Python versi 3.8 atau yang lebih baru.

Sublangkah

- Buat pengguna IAM dengan izin di Akun AWS (konsol) Anda
- Instal Python 3.8 atau yang lebih baru di komputer lokal Anda

Buat pengguna IAM dengan izin di Akun AWS (konsol) Anda

Anda dapat membuat pengguna IAM untuk tutorial. Untuk menyelesaikan tutorial ini, pengguna IAM Anda harus melampirkan kebijakan IAM berikut untuk mengakses AWS sumber daya yang relevan dan melakukan tindakan tertentu. Untuk informasi selengkapnya tentang cara membuat pengguna IAM, lihat Membuat pengguna IAM (konsol) di Panduan Pengguna IAM.

Pengguna IAM Anda memerlukan kebijakan berikut:

- AmazonS3 FullAccess - Memberikan izin untuk semua tindakan Amazon S3, termasuk izin untuk membuat dan menggunakan Titik Akses Lambda Objek.
- AWSLambda_FullAccess— Memberikan izin untuk semua tindakan Lambda.
- IAMFullAkses - Memberikan izin untuk semua tindakan IAM.
- IAMAccessAnalyzerReadOnlyAccess— Memberikan izin untuk membaca semua informasi akses yang disediakan oleh IAM Access Analyzer.
- CloudWatchLogsFullAccess— Memberikan akses penuh ke CloudWatch Log.

> **Note**
>
> Untuk kesederhanaan, tutorial ini membuat dan menggunakan pengguna IAM. Setelah menyelesaikan tutorial ini, ingatlah untukHapus pengguna IAM. Untuk penggunaan produksi, kami menyarankan Anda mengikuti praktik terbaik Keamanan di IAM dalam Panduan Pengguna IAM. Praktik terbaik mengharuskan pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses AWS dengan kredensi sementara. Praktik terbaik lainnya adalah meminta beban kerja untuk menggunakan kredensil sementara dengan peran IAM untuk mengakses. AWS Untuk mempelajari cara menggunakan AWS IAM Identity Center untuk membuat pengguna dengan kredensial sementara, lihat Memulai di AWS IAM Identity Center Panduan Pengguna.
> Tutorial ini juga menggunakan kebijakan AWS terkelola akses penuh. Untuk penggunaan produksi, sebaiknya Anda hanya memberikan izin minimum yang diperlukan untuk kasus penggunaan Anda, sesuai dengan praktik terbaik keamanan.

Instal Python 3.8 atau yang lebih baru di komputer lokal Anda

Gunakan prosedur berikut untuk menginstal Python 3.8 atau yang lebih baru pada mesin lokal Anda. Untuk petunjuk penginstalan lainnya, lihat halaman Mengunduh Python di Panduan Pemula Python.

1. Buka terminal atau shell lokal Anda dan jalankan perintah berikut untuk menentukan apakah Python sudah diinstal, dan jika demikian, versi mana yang diinstal.

   ```
   python --version
   ```

2. Jika Anda tidak memiliki Python 3.8 atau yang lebih baru, unduh penginstal resmi Python 3.8 atau yang lebih baru yang cocok untuk mesin lokal Anda.
3. Jalankan penginstal dengan mengklik dua kali file yang diunduh, dan ikuti langkah-langkah untuk menyelesaikan instalasi.

   Untuk pengguna Windows, pilih Tambahkan Python 3.X ke PATH di wizard instalasi sebelum memilih Instal Sekarang.
4. Mulai ulang terminal Anda dengan menutup dan membukanya kembali.
5. Jalankan perintah berikut untuk memverifikasi bahwa Python 3.8 atau yang lebih baru diinstal dengan benar.

   Untuk pengguna macOS, jalankan perintah ini:

```
python3 --version
```

Untuk pengguna Windows, jalankan perintah ini:

```
python --version
```

6. Jalankan perintah berikut untuk memverifikasi bahwa manajer paket pip3 diinstal. Jika Anda melihat nomor versi pip dan python 3.8 atau yang lebih baru dalam respons perintah, itu berarti manajer paket pip3 berhasil diinstal.

```
pip --version
```

Langkah 1: Buat ember S3

Buat bucket untuk menyimpan data asli yang Anda rencanakan untuk diubah.

> **Note**
>
> Titik akses dapat dilampirkan untuk sumber data lain, seperti Amazon FSx untuk volume OpenZFS, namun tutorial ini menggunakan titik akses pendukung yang terpasang pada bucket S3.

Untuk membuat bucket

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Pilih Buat bucket.

   Halaman Buat bucket terbuka.
4. Untuk nama Bucket, masukkan nama (misalnya,**`tutorial-bucket`**) untuk bucket Anda.

   Untuk informasi selengkapnya tentang penamaan bucket di Amazon S3, lihat. Aturan penamaan bucket tujuan umum
5. Untuk Wilayah, pilih Wilayah AWS tempat Anda ingin ember berada.

Untuk informasi selengkapnya tentang Wilayah bucket, lihatIkhtisar ember tujuan umum.

6. Untuk pengaturan Blokir Akses Publik untuk bucket ini, pertahankan pengaturan default (Blokir semua akses publik diaktifkan).

   Kami menyarankan agar Anda tetap mengaktifkan semua pengaturan Blokir Akses Publik kecuali Anda perlu menonaktifkan satu atau beberapa pengaturan tersebut untuk kasus penggunaan Anda. Untuk informasi lebih lanjut tentang pemblokiran akses publik, lihat Melakukan blok akses publik ke penyimpanan Amazon S3 Anda.
7. Untuk pengaturan yang tersisa, pertahankan default.

   (Opsional) Jika Anda ingin mengonfigurasi pengaturan bucket tambahan untuk kasus penggunaan spesifik Anda, lihatMembuat ember tujuan umum.
8. Pilih Buat bucket.

Langkah 2: Unggah file ke bucket S3

Unggah file teks ke bucket S3. File teks ini berisi data asli yang akan Anda ubah menjadi huruf besar nanti dalam tutorial ini.

Misalnya, Anda dapat mengunggah `tutorial.txt` file yang berisi teks berikut:

```
Amazon S3 Object Lambda Tutorial:
You can add your own code to process data retrieved from S3 before
returning it to an application.
```

Untuk mengunggah file ke ember

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Dalam daftar Bucket, pilih nama bucket yang Anda buat di Langkah 1 (misalnya,**`tutorial-bucket`**) untuk mengunggah file Anda.
4. Di tab Objek untuk bucket Anda, pilih Unggah.
5. Pada halaman Unggah, di bawah File dan folder, pilih Tambahkan file.
6. Pilih file yang akan diunggah, lalu pilih Buka. Misalnya, Anda dapat mengunggah contoh `tutorial.txt` file yang disebutkan sebelumnya.

7. Pilih Unggah.

Langkah 3: Buat titik akses S3

Untuk menggunakan Titik Akses Lambda Objek S3 untuk mengakses dan mengubah data asli, Anda harus membuat titik akses S3 dan mengaitkannya dengan bucket S3 yang Anda buat di Langkah 1. Titik akses harus Wilayah AWS sama dengan objek yang ingin Anda ubah.

Kemudian dalam tutorial ini, Anda akan menggunakan titik akses ini sebagai titik akses pendukung untuk Object Lambda Access Point Anda.

Untuk membuat titik akses

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Pada panel navigasi di kiri, pilih Titik Akses.
3. Pada halaman Titik Akses, pilih Buat titik akses.
4. Di bidang Nama titik akses, masukkan nama (misalnya,**`tutorial-access-point`**) untuk titik akses.

   Untuk informasi lebih lanjut tentang penamaan titik akses, lihat Aturan penamaan untuk titik akses.
5. Di bidang Sumber data, masukkan nama bucket yang Anda buat di Langkah 1 (misalnya,**`tutorial-bucket`**). S3 melampirkan titik akses ke bucket ini.

   (Opsional) Anda dapat memilih Browse S3 untuk menelusuri dan mencari bucket di akun Anda. Jika Anda memilih Browse S3, pilih bucket yang diinginkan, lalu pilih Pilih jalur untuk mengisi bidang nama Bucket dengan nama bucket tersebut.
6. Untuk asal Jaringan, pilih Internet.

   Untuk informasi lebih lanjut tentang asal jaringan untuk titik akses, lihat Membuat titik akses terbatas pada cloud privat virtual.
7. Secara default, semua pengaturan Blokir Akses Publik diaktifkan untuk titik akses Anda. Kami menyarankan agar Anda tetap mengaktifkan Blokir semua akses publik.

   Untuk informasi selengkapnya, lihat Mengelola akses publik ke titik akses untuk ember tujuan umum.
8. Untuk semua pengaturan titik akses lainnya, pertahankan pengaturan default.

(Opsional) Anda dapat mengubah pengaturan titik akses untuk mendukung kasus penggunaan Anda. Untuk tutorial ini, kami sarankan untuk menjaga pengaturan default.

(Opsional) Jika Anda perlu mengelola akses ke titik akses Anda, Anda dapat menentukan kebijakan titik akses. Untuk informasi selengkapnya, lihat Contoh kebijakan untuk titik akses.

9. Pilih Buat titik akses.

Langkah 4: Buat fungsi Lambda

Untuk mengubah data asli, buat fungsi Lambda untuk digunakan dengan Titik Akses Lambda Objek S3 Anda.

Sublangkah

- Tulis kode fungsi Lambda dan buat paket penyebaran dengan lingkungan virtual
- Buat fungsi Lambda dengan peran eksekusi (konsol)
- Terapkan kode fungsi Lambda Anda dengan arsip file.zip dan konfigurasikan fungsi Lambda (konsol)

Tulis kode fungsi Lambda dan buat paket penyebaran dengan lingkungan virtual

1. Pada mesin lokal Anda, buat folder dengan nama folder `object-lambda` untuk lingkungan virtual untuk digunakan nanti dalam tutorial ini.
2. Di `object-lambda` folder, buat file dengan fungsi Lambda yang mengubah semua teks di objek asli menjadi huruf besar. Misalnya, Anda dapat menggunakan fungsi berikut yang ditulis dengan Python. Simpan fungsi ini dalam file bernama`transform.py`.

```
import boto3
import requests
from botocore.config import Config

# This function capitalizes all text in the original object
def lambda_handler(event, context):
    object_context = event["getObjectContext"]
    # Get the presigned URL to fetch the requested original object
    # from S3
    s3_url = object_context["inputS3Url"]
    # Extract the route and request token from the input context
    request_route = object_context["outputRoute"]
```

```
    request_token = object_context["outputToken"]

    # Get the original S3 object using the presigned URL
    response = requests.get(s3_url)
    original_object = response.content.decode("utf-8")

    # Transform all text in the original object to uppercase
    # You can replace it with your custom code based on your use case
    transformed_object = original_object.upper()

    # Write object back to S3 Object Lambda
    s3 = boto3.client('s3', config=Config(signature_version='s3v4'))
    # The WriteGetObjectResponse API sends the transformed data
    # back to S3 Object Lambda and then to the user
    s3.write_get_object_response(
        Body=transformed_object,
        RequestRoute=request_route,
        RequestToken=request_token)

    # Exit the Lambda function: return the status code
    return {'status_code': 200}
```

> **Note**
>
> Contoh sebelumnya fungsi Lambda memuat seluruh objek yang diminta ke dalam memori sebelum mengubahnya dan mengembalikannya ke klien. Atau, Anda dapat melakukan streaming objek dari S3 untuk menghindari memuat seluruh objek ke dalam memori. Pendekatan ini dapat berguna saat bekerja dengan benda besar. Untuk informasi selengkapnya tentang respons streaming dengan Titik Akses Objek Lambda, lihat contoh streaming di. Bekerja dengan `GetObject` permintaan di Lambda

Saat Anda menulis fungsi Lambda untuk digunakan dengan Titik Akses Lambda Objek S3, fungsi ini didasarkan pada konteks peristiwa masukan yang disediakan S3 Object Lambda ke fungsi Lambda. Konteks acara memberikan informasi tentang permintaan yang dibuat dalam acara yang diteruskan dari S3 Object Lambda ke Lambda. Ini berisi parameter yang Anda gunakan untuk membuat fungsi Lambda.

Bidang yang digunakan untuk membuat fungsi Lambda sebelumnya adalah sebagai berikut:

Bidang `getObjectContext` berarti rincian input dan output untuk koneksi ke Amazon S3 dan S3 Object Lambda. Ini memiliki bidang-bidang berikut:

- `inputS3Url`— URL presigned yang dapat digunakan fungsi Lambda untuk mengunduh objek asli dari titik akses pendukung. Dengan menggunakan URL yang telah ditetapkan sebelumnya, fungsi Lambda tidak perlu memiliki izin baca Amazon S3 untuk mengambil objek asli dan hanya dapat mengakses objek yang diproses oleh setiap pemanggilan.
- `outputRoute`— Token routing yang ditambahkan ke URL Lambda Objek S3 saat fungsi Lambda memanggil `WriteGetObjectResponse` untuk mengirim kembali objek yang diubah.
- `outputToken`— Token yang digunakan oleh S3 Object Lambda untuk `WriteGetObjectResponse` mencocokkan panggilan dengan penelepon asli saat mengirim kembali objek yang diubah.

Untuk informasi selengkapnya tentang semua bidang dalam konteks acara, lihat Format konteks peristiwa dan penggunaan danMenulis fungsi Lambda untuk Titik Akses S3 Lambda Objek.

3. Di terminal lokal Anda, masukkan perintah berikut untuk menginstal `virtualenv` paket:

```
python -m pip install virtualenv
```

4. Di terminal lokal Anda, buka `object-lambda` folder yang Anda buat sebelumnya, lalu masukkan perintah berikut untuk membuat dan menginisialisasi lingkungan virtual yang disebut`venv`.

```
python -m virtualenv venv
```

5. Untuk mengaktifkan lingkungan virtual, masukkan perintah berikut untuk menjalankan `activate` file dari folder lingkungan:

   Untuk pengguna macOS, jalankan perintah ini:

```
source venv/bin/activate
```

   Untuk pengguna Windows, jalankan perintah ini:

```
.\venv\Scripts\activate
```

Sekarang, prompt perintah Anda berubah menjadi show (venv), menunjukkan bahwa lingkungan virtual aktif.

6. Untuk menginstal pustaka yang diperlukan, jalankan perintah berikut baris demi baris di lingkungan `venv` virtual.

   Perintah ini menginstal versi terbaru dari dependensi fungsi Lambda `lambda_handler` Anda. Dependensi ini adalah AWS SDK untuk Python (Boto3) dan modul request.

   ```
   pip3 install boto3
   ```

   ```
   pip3 install requests
   ```

7. Untuk menonaktifkan lingkungan virtual, jalankan perintah berikut:

   ```
   deactivate
   ```

8. Untuk membuat paket penyebaran dengan pustaka yang diinstal sebagai `.zip` file bernama `lambda.zip` di root `object-lambda` direktori, jalankan perintah berikut baris demi baris di terminal lokal Anda.

   > **Tip**
   >
   > Perintah berikut mungkin perlu disesuaikan untuk bekerja di lingkungan khusus Anda. Misalnya, pustaka mungkin muncul di `site-packages` ataudist-packages, dan folder pertama mungkin `lib` ataulib64. Juga, `python` folder mungkin diberi nama dengan versi Python yang berbeda. Untuk menemukan paket tertentu, gunakan `pip show` perintah.

   Untuk pengguna macOS, jalankan perintah ini:

   ```
   cd venv/lib/python3.8/site-packages
   ```

   ```
   zip -r ../../../../lambda.zip .
   ```

   Untuk pengguna Windows, jalankan perintah ini:

```
cd .\venv\Lib\site-packages\
```

```
powershell Compress-Archive * ../../../lambda.zip
```

Perintah terakhir menyimpan paket deployment ke akar direktori `object-lambda`.

9. Tambahkan file kode fungsi `transform.py` ke root paket penyebaran Anda.

   Untuk pengguna macOS, jalankan perintah ini:

```
cd ../../../../
```

```
zip -g lambda.zip transform.py
```

   Untuk pengguna Windows, jalankan perintah ini:

```
cd ..\..\..\
```

```
powershell Compress-Archive -update transform.py lambda.zip
```

   Setelah menyelesaikan langkah ini, struktur direktori Anda adalah sebagai berikut:

```
lambda.zip$
  # transform.py
  # __pycache__
  | boto3/
  # certifi/
  # pip/
  # requests/
  ...
```

Buat fungsi Lambda dengan peran eksekusi (konsol)

1. Masuk ke AWS Management Console dan buka AWS Lambda konsol di https://console.aws.amazon.com/lambda/.

2. Di panel navigasi kiri, pilih Fungsi.

3. Pilih Buat fungsi.
4. Pilih Penulis dari scratch.
5. Di bagian Informasi dasar, lakukan hal berikut:
   a. Untuk Nama fungsi, masukkan **`tutorial-object-lambda-function`**.
   b. Untuk Runtime, pilih Python 3.8 atau versi yang lebih baru.
6. Perluas bagian Ubah peran eksekusi default. Di bawah Peran eksekusi, pilih Buat peran baru dengan izin Lambda dasar.

   Pada Langkah 5 nanti dalam tutorial ini, Anda melampirkan AmazonS3 ke peran eksekusi ObjectLambdaExecutionRolePolicy fungsi Lambda ini.
7. Pertahankan pengaturan yang tersisa disetel ke default.
8. Pilih Buat fungsi.

Terapkan kode fungsi Lambda Anda dengan arsip file.zip dan konfigurasikan fungsi Lambda (konsol)

1. Di AWS Lambda konsol di https://console.aws.amazon.com/lambda/, pilih Fungsi di panel navigasi kiri.
2. Pilih fungsi Lambda yang Anda buat sebelumnya (misalnya,**`tutorial-object-lambda-function`**).
3. Pada halaman detail fungsi Lambda, pilih tab Kode. Di bagian Sumber Kode, pilih Unggah dari dan kemudian file.zip.
4. Pilih Unggah untuk memilih `.zip` file lokal Anda.
5. Pilih `lambda.zip` file yang Anda buat sebelumnya, lalu pilih Buka.
6. Pilih Simpan.
7. Di bagian Pengaturan waktu proses, pilih Edit.
8. Pada halaman Edit pengaturan runtime, konfirmasikan bahwa Runtime diatur ke Python 3.8 atau versi yang lebih baru.
9. Untuk memberi tahu runtime Lambda metode handler mana dalam kode fungsi Lambda Anda yang akan dipanggil, masukkan untuk Handler. **`transform.lambda_handler`**

   Ketika Anda mengkonfigurasi fungsi dalam Python, nilai pengaturan handler adalah nama file dan nama modul handler, dipisahkan oleh titik. Misalnya, `transform.lambda_handler` memanggil `lambda_handler` metode yang didefinisikan dalam `transform.py` file.

10. Pilih Simpan.
11. (Opsional) Pada halaman detail fungsi Lambda Anda, pilih tab Konfigurasi. Di panel navigasi kiri, pilih Konfigurasi umum, lalu pilih Edit. Di bidang Timeout, masukkan **1** min **0** sec. Pertahankan pengaturan yang tersisa disetel ke default, dan pilih Simpan.

    Timeout adalah jumlah waktu yang memungkinkan Lambda menjalankan fungsi untuk pemanggilan sebelum menghentikannya. Default-nya adalah 3 detik. Durasi maksimum untuk fungsi Lambda yang digunakan oleh S3 Object Lambda adalah 60 detik. Harga didasarkan pada jumlah memori yang dikonfigurasi dan jumlah waktu kode Anda berjalan.

Langkah 5: Konfigurasikan kebijakan IAM untuk peran eksekusi fungsi Lambda Anda

Untuk mengaktifkan fungsi Lambda Anda untuk menyediakan data yang disesuaikan dan header respons kepada `GetObject` pemanggil, peran eksekusi fungsi Lambda Anda harus memiliki izin IAM untuk memanggil API. `WriteGetObjectResponse`

Untuk melampirkan kebijakan IAM ke peran fungsi Lambda Anda

1. Di AWS Lambda konsol di https://console.aws.amazon.com/lambda/, pilih Fungsi di panel navigasi kiri.
2. Pilih fungsi yang Anda buat di Langkah 4 (misalnya,**`tutorial-object-lambda-function`**).
3. Pada halaman detail fungsi Lambda Anda, pilih tab Konfigurasi, lalu pilih Izin di panel navigasi kiri.
4. Di bawah Peran eksekusi, pilih tautan nama Peran. Konsol IAM terbuka.
5. Pada halaman Ringkasan konsol IAM untuk peran eksekusi fungsi Lambda Anda, pilih tab Izin. Kemudian, dari menu Tambahkan Izin, pilih Lampirkan kebijakan.
6. Pada halaman Lampirkan Izin, masukkan **`AmazonS3ObjectLambdaExecutionRolePolicy`** di kotak pencarian untuk memfilter daftar kebijakan. Pilih kotak centang di sebelah nama kebijakan AmazonS3 ObjectLambdaExecutionRolePolicy.
7. Pilih Lampirkan kebijakan.

Langkah 6: Buat Titik Akses Lambda Objek S3

Titik Akses Lambda Objek S3 memberikan fleksibilitas untuk memanggil fungsi Lambda langsung dari permintaan S3 GET sehingga fungsi tersebut dapat memproses data yang diambil dari titik akses S3.

Saat membuat dan mengonfigurasi Titik Akses Lambda Objek S3, Anda harus menentukan fungsi Lambda untuk memanggil dan memberikan konteks acara dalam format JSON sebagai parameter khusus untuk digunakan Lambda.

Untuk membuat Objek S3 Lambda Access Point

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi sebelah kiri, pilih Titik Akses Lambda Objek.
3. Pada halaman Titik Akses Lambda Objek, pilih Buat Titik Akses Lambda Objek.
4. Untuk nama Object Lambda Access Point, masukkan nama yang ingin Anda gunakan untuk Object Lambda Access Point (misalnya,). **`tutorial-object-lambda-accesspoint`**
5. Untuk Mendukung Titik Akses, masukkan atau telusuri ke titik akses standar yang Anda buat di Langkah 3 (misalnya,**`tutorial-access-point`**), lalu pilih Pilih Titik Akses pendukung.
6. Untuk S3 APIs, untuk mengambil objek dari bucket S3 agar fungsi Lambda dapat diproses, pilih. GetObject
7. Untuk fungsi Invoke Lambda, Anda dapat memilih salah satu dari dua opsi berikut untuk tutorial ini.
   - Pilih Pilih dari fungsi di akun Anda, lalu pilih fungsi Lambda yang Anda buat di Langkah 4 (misalnya,**`tutorial-object-lambda-function`**) dari daftar dropdown fungsi Lambda.
   - Pilih Masukkan ARN, lalu masukkan Nama Sumber Daya Amazon (ARN) dari fungsi Lambda yang Anda buat di Langkah 4.
8. Untuk versi fungsi Lambda, pilih $LATEST (versi terbaru dari fungsi Lambda yang Anda buat di Langkah 4).
9. (Opsional) Jika Anda memerlukan fungsi Lambda Anda untuk mengenali dan memproses permintaan GET dengan header rentang dan nomor bagian, pilih Fungsi Lambda mendukung permintaan menggunakan rentang dan fungsi Lambda mendukung permintaan menggunakan nomor bagian. Jika tidak, kosongkan dua kotak centang ini.

   Untuk informasi selengkapnya tentang cara menggunakan nomor rentang atau bagian dengan S3 Object Lambda, lihat. Bekerja dengan header Range dan partNumber
10. (Opsional) Di bawah Payload - opsional, tambahkan teks JSON untuk menyediakan fungsi Lambda Anda dengan informasi tambahan.

    Payload adalah teks JSON opsional yang dapat Anda berikan ke fungsi Lambda Anda sebagai masukan untuk semua pemanggilan yang berasal dari Titik Akses Lambda Objek S3 tertentu.

Untuk menyesuaikan perilaku untuk beberapa Titik Akses Lambda Objek yang menjalankan fungsi Lambda yang sama, Anda dapat mengonfigurasi muatan dengan parameter yang berbeda, sehingga memperluas fleksibilitas fungsi Lambda Anda.

Untuk informasi selengkapnya tentang payload, lihatFormat konteks peristiwa dan penggunaan.

11. (Opsional) Untuk metrik Permintaan - opsional, pilih Nonaktifkan atau Aktifkan untuk menambahkan pemantauan Amazon S3 ke Titik Akses Lambda Objek Anda. Metrik permintaan ditagih dengan tarif Amazon CloudWatch standar. Untuk informasi selengkapnya, lihat harga CloudWatch.
12. Di bawah kebijakan Titik Akses Objek Lambda - opsional, pertahankan pengaturan default.

    (Opsional) Anda dapat menetapkan kebijakan sumber daya. Kebijakan resource ini memberikan izin `GetObject` API untuk menggunakan Object Lambda Access Point yang ditentukan.
13. Pertahankan pengaturan yang tersisa disetel ke default, dan pilih Buat Objek Lambda Access Point.

Langkah 7: Lihat data yang diubah

Sekarang, S3 Object Lambda siap untuk mengubah data Anda untuk kasus penggunaan Anda. Dalam tutorial ini, S3 Object Lambda mengubah semua teks dalam objek Anda menjadi huruf besar.

Sublangkah

- Lihat data yang diubah di Titik Akses Lambda Objek S3 Anda
- Jalankan skrip Python untuk mencetak data asli dan diubah

Lihat data yang diubah di Titik Akses Lambda Objek S3 Anda

Saat Anda meminta untuk mengambil file melalui Titik Akses Lambda Objek S3, Anda membuat panggilan `GetObject` API ke S3 Object Lambda. S3 Object Lambda memanggil fungsi Lambda untuk mengubah data Anda, dan kemudian mengembalikan data yang diubah sebagai respons terhadap panggilan API S3 standar. `GetObject`

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi sebelah kiri, pilih Titik Akses Lambda Objek.
3. Pada halaman Titik Akses Objek Lambda, pilih Titik Akses Lambda Objek S3 yang Anda buat di Langkah 6 (misalnya,). **`tutorial-object-lambda-accesspoint`**

4. Pada tab Objek pada Titik Akses Lambda Objek S3 Anda, pilih file yang memiliki nama yang sama (misalnya`tutorial.txt`,) dengan file yang Anda unggah ke bucket S3 di Langkah 2.

   File ini harus berisi semua data yang diubah.

5. Untuk melihat data yang diubah, pilih Buka atau Unduh.

Jalankan skrip Python untuk mencetak data asli dan diubah

Anda dapat menggunakan S3 Object Lambda dengan aplikasi yang ada. Untuk melakukannya, perbarui konfigurasi aplikasi Anda untuk menggunakan ARN Titik Akses Lambda Objek S3 baru yang Anda buat di Langkah 6 untuk mengambil data dari S3.

Contoh skrip Python berikut mencetak data asli dari bucket S3 dan data yang diubah dari S3 Object Lambda Access Point.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi sebelah kiri, pilih Titik Akses Lambda Objek.
3. Pada halaman Titik Akses Objek Lambda, pilih tombol radio di sebelah kiri Titik Akses Lambda Objek S3 yang Anda buat di Langkah 6 (misalnya,). **`tutorial-object-lambda-accesspoint`**
4. Pilih Salin ARN.
5. Simpan ARN untuk digunakan nanti.
6. Tulis skrip Python di mesin lokal Anda untuk mencetak data asli (misalnya,`tutorial.txt`) dari Bucket S3 Anda dan data yang diubah (misalnya,`tutorial.txt`) dari Titik Akses Lambda Objek S3 Anda). Anda dapat menggunakan contoh script berikut.

```
import boto3
from botocore.config import Config

s3 = boto3.client('s3', config=Config(signature_version='s3v4'))

def getObject(bucket, key):
    objectBody = s3.get_object(Bucket = bucket, Key = key)
    print(objectBody["Body"].read().decode("utf-8"))
    print("\n")

print('Original object from the S3 bucket:')
```

```
# Replace the two input parameters of getObject() below with
# the S3 bucket name that you created in Step 1 and
# the name of the file that you uploaded to the S3 bucket in Step 2
getObject("tutorial-bucket",
          "tutorial.txt")

print('Object transformed by S3 Object Lambda:')
# Replace the two input parameters of getObject() below with
# the ARN of your S3 Object Lambda Access Point that you saved earlier and
# the name of the file with the transformed data (which in this case is
# the same as the name of the file that you uploaded to the S3 bucket
# in Step 2)
getObject("arn:aws:s3-object-lambda:us-west-2:111122223333:accesspoint/tutorial-
object-lambda-accesspoint",
          "tutorial.txt")
```

7. Simpan skrip Python Anda dengan nama khusus (misalnya,`tutorial_print.py`) di folder (misalnya,`object-lambda`) yang Anda buat di Langkah 4 di mesin lokal Anda.
8. Di terminal lokal Anda, jalankan perintah berikut dari root direktori (misalnya,`object-lambda`) yang Anda buat di Langkah 4.

   ```
   python3 tutorial_print.py
   ```

   Anda akan melihat data asli dan data yang diubah (semua teks sebagai huruf besar) melalui terminal. Misalnya, Anda akan melihat sesuatu seperti teks berikut.

   ```
   Original object from the S3 bucket:
   Amazon S3 Object Lambda Tutorial:
   You can add your own code to process data retrieved from S3 before
   returning it to an application.

   Object transformed by S3 Object Lambda:
   AMAZON S3 OBJECT LAMBDA TUTORIAL:
   YOU CAN ADD YOUR OWN CODE TO PROCESS DATA RETRIEVED FROM S3 BEFORE
   RETURNING IT TO AN APPLICATION.
   ```

Langkah 8: Membersihkan

Jika Anda mengubah data Anda melalui S3 Object Lambda hanya sebagai latihan pembelajaran, hapus AWS sumber daya yang Anda alokasikan sehingga Anda tidak lagi menambah biaya.

Sublangkah

- Hapus Titik Akses Objek Lambda
- Hapus titik akses S3
- Hapus peran eksekusi untuk fungsi Lambda Anda
- Hapus fungsi Lambda
- Hapus grup CloudWatch log
- Hapus file asli di bucket sumber S3
- Hapus bucket sumber S3
- Hapus pengguna IAM

Hapus Titik Akses Objek Lambda

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi sebelah kiri, pilih Titik Akses Lambda Objek.
3. Pada halaman Titik Akses Objek Lambda, pilih tombol radio di sebelah kiri Titik Akses Lambda Objek S3 yang Anda buat di Langkah 6 (misalnya,). **`tutorial-object-lambda-accesspoint`**
4. Pilih Hapus.
5. Konfirmasikan bahwa Anda ingin menghapus Titik Akses Objek Lambda Anda dengan memasukkan namanya di bidang teks yang muncul, lalu pilih Hapus.

Hapus titik akses S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Pada panel navigasi di kiri, pilih Titik Akses.
3. Arahkan ke titik akses yang Anda buat di Langkah 3 (misalnya,**`tutorial-access-point`**), dan pilih tombol radio di sebelah nama titik akses.
4. Pilih Hapus.
5. Konfirmasikan bahwa Anda ingin menghapus jalur akses Anda dengan memasukkan namanya di bidang teks yang muncul, lalu pilih Hapus.

Hapus peran eksekusi untuk fungsi Lambda Anda

1. Masuk ke AWS Management Console dan buka AWS Lambda konsol di https://console.aws.amazon.com/lambda/.
2. Di panel navigasi kiri, pilih Fungsi.
3. Pilih fungsi yang Anda buat di Langkah 4 (misalnya,**`tutorial-object-lambda-function`**).
4. Pada halaman detail fungsi Lambda Anda, pilih tab Konfigurasi, lalu pilih Izin di panel navigasi kiri.
5. Di bawah Peran eksekusi, pilih tautan nama Peran. Konsol IAM terbuka.
6. Pada halaman Ringkasan konsol IAM tentang peran eksekusi fungsi Lambda Anda, pilih Hapus peran.
7. Di kotak dialog Hapus peran, pilih Ya, hapus.

Hapus fungsi Lambda

1. Di AWS Lambda konsol di https://console.aws.amazon.com/lambda/, pilih Fungsi di panel navigasi kiri.
2. Pilih kotak centang di sebelah kiri nama fungsi yang Anda buat di Langkah 4 (misalnya,**`tutorial-object-lambda-function`**).
3. Pilih Tindakan, lalu pilih Hapus.
4. Di kotak dialog Hapus fungsi, pilih Hapus.

Hapus grup CloudWatch log

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Pada panel navigasi kiri, pilih Grup log.
3. Temukan grup log yang namanya diakhiri dengan fungsi Lambda yang Anda buat di Langkah 4 (misalnya,**`tutorial-object-lambda-function`**).
4. Pilih kotak centang di sebelah kiri nama grup log.
5. Pilih Tindakan, lalu pilih Hapus grup log.
6. Di kotak dialog Hapus grup log, pilih Hapus.

Hapus file asli di bucket sumber S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Dalam daftar nama Bucket, pilih nama bucket tempat Anda mengunggah file asli di Langkah 2 (misalnya,**`tutorial-bucket`**).
4. Pilih kotak centang di sebelah kiri nama objek yang ingin Anda hapus (misalnya,`tutorial.txt`).
5. Pilih Hapus.
6. Pada halaman Hapus objek, di objek Hapus secara permanen? bagian, konfirmasikan bahwa Anda ingin menghapus objek ini dengan memasukkan **`permanently delete`** di kotak teks.
7. Pilih Hapus objek.

Hapus bucket sumber S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Dalam daftar Bucket, pilih tombol radio di sebelah nama bucket yang Anda buat di Langkah 1 (misalnya,**`tutorial-bucket`**).
4. Pilih Hapus.
5. Di halaman Hapus bucket, konfirmasikan bahwa Anda ingin menghapus bucket dengan memasukkan nama bucket ke dalam bidang teks, lalu pilih Hapus bucket.

Hapus pengguna IAM

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi kiri, pilih Pengguna, lalu pilih kotak centang di sebelah nama pengguna yang ingin Anda hapus.
3. Pilih Hapus di bagian atas halaman.
4. Di Hapus***`user name`***? kotak dialog, masukkan nama pengguna di bidang input teks untuk mengonfirmasi penghapusan pengguna. Pilih Hapus.

Langkah selanjutnya

Setelah menyelesaikan tutorial ini, Anda dapat menyesuaikan fungsi Lambda untuk kasus penggunaan Anda untuk memodifikasi data yang dikembalikan oleh permintaan GET S3 standar.

Berikut ini adalah daftar kasus penggunaan umum untuk S3 Object Lambda:

- Menyembunyi data sensitif untuk keamanan dan kepatuhan.

  Untuk informasi selengkapnya, lihat Tutorial: Mendeteksi dan menyunting data PII dengan Objek S3 Lambda dan Amazon Comprehend.
- Memfilter baris data tertentu untuk memberikan informasi spesifik.
- Menambah data dengan informasi dari layanan atau database lain.
- Mengonversi seluruh format data, seperti mengonversi XHTML ke JSON untuk kompatibilitas aplikasi.
- Mengompresi atau mendekompresi file saat sedang diunduh.
- Mengubah ukuran dan menandai gambar.

  Untuk informasi lebih lanjut, lihat Tutorial: Menggunakan S3 Object Lambda untuk menandai gambar secara dinamis saat diambil.
- Menerapkan aturan otorisasi khusus untuk mengakses data.

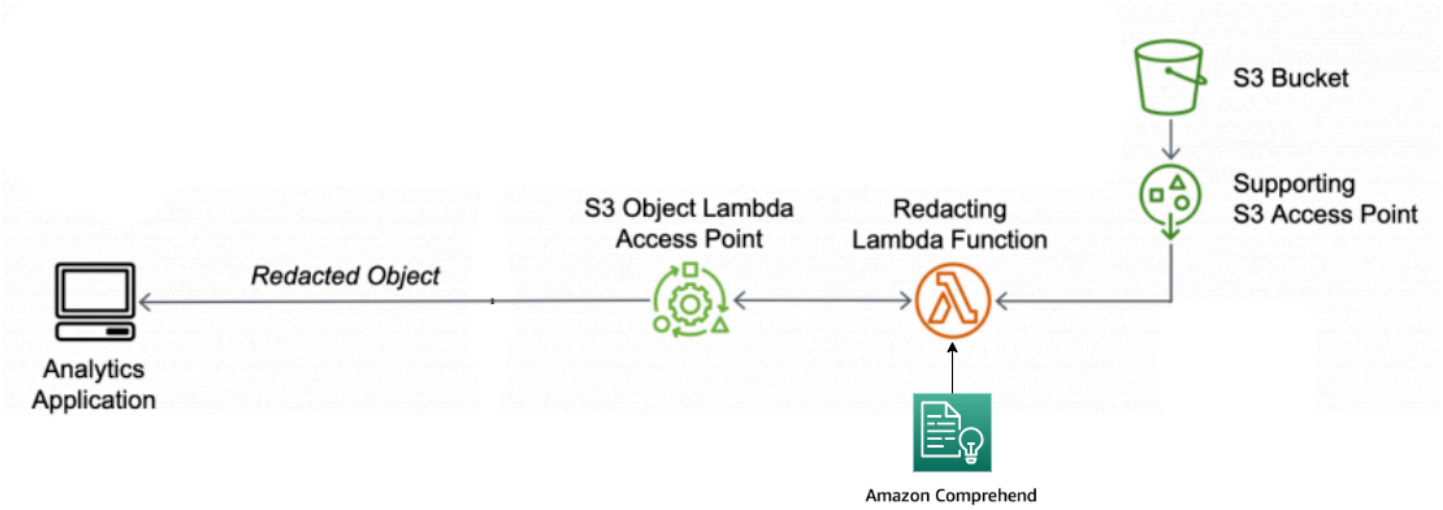
Untuk informasi selengkapnya tentang S3 Lambda Objek, lihat Mengubah objek dengan S3 Lambda Objek.

Tutorial: Mendeteksi dan menyunting data PII dengan Objek S3 Lambda dan Amazon Comprehend

Saat Anda menggunakan Amazon S3 untuk kumpulan data bersama untuk beberapa aplikasi dan pengguna untuk mengakses, penting untuk membatasi informasi istimewa, seperti informasi identitas pribadi (PII), hanya untuk entitas yang berwenang. Misalnya, ketika aplikasi pemasaran menggunakan beberapa data yang mengandung PII, mungkin perlu terlebih dahulu menutupi data PII untuk memenuhi persyaratan privasi data. Juga, ketika aplikasi analitik menggunakan kumpulan data inventaris pesanan produksi, mungkin perlu terlebih dahulu menyunting informasi kartu kredit pelanggan untuk mencegah kebocoran data yang tidak diinginkan.

Dengan S3 Object Lambda dan fungsi AWS Lambda bawaan yang didukung oleh Amazon Comprehend, Anda dapat melindungi data PII yang diambil dari S3 sebelum mengembalikannya

ke aplikasi. Secara khusus, Anda dapat menggunakan fungsi Lambda bawaan sebagai fungsi penyuntingan dan melampirkannya ke Titik Akses Lambda Objek S3. Ketika aplikasi (misalnya, aplikasi analitik) mengirimkan permintaan GET S3 standar, permintaan ini dibuat melalui Titik Akses Lambda Objek S3 memanggil fungsi Lambda penyuntingan bawaan untuk mendeteksi dan menyunting data PII yang diambil dari sumber data yang mendasarinya melalui jalur akses S3 pendukung. Kemudian, Titik Akses Lambda Objek S3 mengembalikan hasil yang disunting kembali ke aplikasi.



Dalam prosesnya, fungsi Lambda bawaan menggunakan Amazon Comprehend, layanan pemrosesan bahasa alami (NLP), untuk menangkap variasi dalam bagaimana PII direpresentasikan, terlepas dari bagaimana PII ada dalam teks (seperti numerik atau sebagai kombinasi kata dan angka). Amazon Comprehend bahkan dapat menggunakan konteks dalam teks untuk mengetahui apakah angka 4 digit adalah PIN, empat angka terakhir dari nomor Jaminan Sosial (SSN), atau satu tahun. Amazon Comprehend memproses file teks apa pun dalam format UTF-8 dan dapat melindungi PII dalam skala besar tanpa memengaruhi akurasi. Untuk informasi selengkapnya, lihat Apa itu Amazon Comprehend? di Panduan Pengembang Amazon Comprehend.

Tujuan

Dalam tutorial ini, Anda belajar cara menggunakan S3 Object Lambda dengan fungsi Lambda prebuilt. `ComprehendPiiRedactionS3ObjectLambda` Fungsi ini menggunakan Amazon Comprehend untuk mendeteksi entitas PII. Kemudian menyunting entitas ini dengan menggantinya dengan tanda bintang. Dengan menyunting PII, Anda menyembunyikan data sensitif, yang dapat membantu keamanan dan kepatuhan.

Anda juga mempelajari cara menggunakan dan mengonfigurasi AWS Lambda fungsi bawaan AWS Serverless Application Repositoryuntuk bekerja sama dengan S3 Object Lambda untuk penyebaran yang mudah.

Topik

- Prasyarat: Buat pengguna IAM dengan izin
- Langkah 1: Buat ember S3
- Langkah 2: Unggah file ke bucket S3
- Langkah 3: Buat titik akses S3
- Langkah 4: Konfigurasikan dan terapkan fungsi Lambda bawaan
- Langkah 5: Buat Titik Akses Lambda Objek S3
- Langkah 6: Gunakan S3 Object Lambda Access Point untuk mengambil file yang disunting
- Langkah 7: Bersihkan
- Langkah selanjutnya

Prasyarat: Buat pengguna IAM dengan izin

Sebelum memulai tutorial ini, Anda harus memiliki AWS akun yang dapat Anda masuki sebagai AWS Identity and Access Management pengguna (pengguna IAM) dengan izin yang benar.

Anda dapat membuat pengguna IAM untuk tutorial. Untuk menyelesaikan tutorial ini, pengguna IAM Anda harus melampirkan kebijakan IAM berikut untuk mengakses AWS sumber daya yang relevan dan melakukan tindakan tertentu.

> **Note**
>
> Untuk kesederhanaan, tutorial ini membuat dan menggunakan pengguna IAM. Setelah menyelesaikan tutorial ini, ingatlah untukHapus pengguna IAM. Untuk penggunaan produksi, kami menyarankan Anda mengikuti praktik terbaik Keamanan di IAM dalam Panduan Pengguna IAM. Praktik terbaik mengharuskan pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses AWS dengan kredensi sementara. Praktik terbaik lainnya adalah meminta beban kerja untuk menggunakan kredensil sementara dengan peran IAM untuk mengakses. AWS Untuk mempelajari cara menggunakan AWS IAM Identity Center untuk membuat pengguna dengan kredensial sementara, lihat Memulai di AWS IAM Identity Center Panduan Pengguna.

> Tutorial ini juga menggunakan kebijakan akses penuh. Untuk penggunaan produksi, sebaiknya Anda hanya memberikan izin minimum yang diperlukan untuk kasus penggunaan Anda, sesuai dengan praktik terbaik keamanan.

Pengguna IAM Anda memerlukan kebijakan AWS terkelola berikut:

- AmazonS3 FullAccess - Memberikan izin untuk semua tindakan Amazon S3, termasuk izin untuk membuat dan menggunakan Titik Akses Lambda Objek.
- AWSLambda_FullAccess— Memberikan izin untuk semua tindakan Lambda.
- AWSCloudFormationFullAccess— Memberikan izin untuk semua AWS CloudFormation tindakan.
- IAMFullAkses - Memberikan izin untuk semua tindakan IAM.
- IAMAccessAnalyzerReadOnlyAccess— Memberikan izin untuk membaca semua informasi akses yang disediakan oleh IAM Access Analyzer.

Anda dapat langsung melampirkan kebijakan yang ada saat membuat pengguna IAM. Untuk informasi selengkapnya tentang cara membuat pengguna IAM, lihat Membuat pengguna IAM (konsol) di Panduan Pengguna IAM.

Selain itu, pengguna IAM Anda memerlukan kebijakan yang dikelola pelanggan. Untuk memberikan izin pengguna IAM ke semua AWS Serverless Application Repository sumber daya dan tindakan, Anda harus membuat kebijakan IAM dan melampirkan kebijakan tersebut ke pengguna IAM.

Untuk membuat dan melampirkan kebijakan IAM ke pengguna IAM Anda

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi di sebelah kiri, pilih Kebijakan.
3. Pilih Buat kebijakan.
4. Pada tab Editor visual, untuk Layanan, pilih Pilih layanan. Kemudian, pilih Serverless Application Repository.
5. Untuk Tindakan, di bawah tindakan Manual, pilih Semua tindakan Repositori Aplikasi Tanpa Server (serverlessrepo: *) untuk tutorial ini.

Sebagai praktik keamanan terbaik, Anda harus mengizinkan izin hanya untuk tindakan dan sumber daya yang dibutuhkan pengguna, berdasarkan kasus penggunaan Anda. Untuk informasi selengkapnya, lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.

6. Untuk Sumber Daya, pilih Semua sumber daya untuk tutorial ini.

   Sebagai praktik terbaik, Anda harus menentukan izin hanya untuk sumber daya tertentu di akun tertentu. Atau, Anda dapat memberikan hak istimewa paling sedikit menggunakan kunci kondisi. Untuk informasi selengkapnya, lihat Berikan hak istimewa paling sedikit di Panduan Pengguna IAM.

7. Pilih Berikutnya: Tanda.
8. Pilih Berikutnya: Tinjau.
9. Pada halaman Kebijakan ulasan, masukkan Nama (misalnya,**`tutorial-serverless-application-repository`**) dan Deskripsi (opsional) untuk kebijakan yang Anda buat. Tinjau ringkasan kebijakan untuk memastikan bahwa Anda telah memberikan izin yang dimaksud, lalu pilih Buat kebijakan untuk menyimpan kebijakan baru Anda.
10. Pada panel navigasi kiri, pilih Pengguna. Kemudian, pilih pengguna IAM untuk tutorial ini.
11. Pada halaman Ringkasan pengguna yang dipilih, pilih tab Izin, lalu pilih Tambahkan izin.
12. Di bawah Izin hibah, pilih Lampirkan kebijakan yang ada secara langsung.
13. Pilih kotak centang di samping kebijakan yang baru saja Anda buat (misalnya,**`tutorial-serverless-application-repository`**) lalu pilih Berikutnya: Tinjau.
14. Di bawah ringkasan Izin, tinjau ringkasan untuk memastikan bahwa Anda melampirkan kebijakan yang dimaksud. Kemudian, pilih Tambahkan izin.

Langkah 1: Buat ember S3

Buat bucket untuk menyimpan data asli yang Anda rencanakan untuk diubah.

> **Note**
>
> Titik akses dapat dilampirkan untuk sumber data lain, seperti Amazon FSx untuk volume OpenZFS, namun tutorial ini menggunakan titik akses pendukung yang terpasang pada bucket S3.

Untuk membuat bucket

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Pilih Buat bucket.

   Halaman Buat bucket terbuka.
4. Untuk nama Bucket, masukkan nama (misalnya,**`tutorial-bucket`**) untuk bucket Anda.

   Untuk informasi selengkapnya tentang penamaan bucket di Amazon S3, lihat. Aturan penamaan bucket tujuan umum
5. Untuk Wilayah, pilih Wilayah AWS tempat Anda ingin bucket berada.

   Untuk informasi selengkapnya tentang Wilayah bucket, lihatIkhtisar ember tujuan umum.
6. Untuk pengaturan Blokir Akses Publik untuk bucket ini, pertahankan pengaturan default (Blokir semua akses publik diaktifkan).

   Kami menyarankan agar Anda tetap mengaktifkan semua pengaturan Blokir Akses Publik kecuali Anda perlu menonaktifkan satu atau beberapa pengaturan tersebut untuk kasus penggunaan Anda. Untuk informasi lebih lanjut tentang pemblokiran akses publik, lihat Melakukan blok akses publik ke penyimpanan Amazon S3 Anda.
7. Untuk pengaturan yang tersisa, pertahankan default.

   (Opsional) Jika Anda ingin mengonfigurasi pengaturan bucket tambahan untuk kasus penggunaan spesifik Anda, lihatMembuat ember tujuan umum.
8. Pilih Buat bucket.

Langkah 2: Unggah file ke bucket S3

Unggah file teks yang berisi data PII yang dikenal dari berbagai jenis, seperti nama, informasi perbankan, nomor telepon, dan SSNs, ke bucket S3 sebagai data asli yang akan Anda edit PII nanti dalam tutorial ini.

Misalnya, Anda dapat mengunggah `tutorial.txt` file berikut. Ini adalah contoh file input dari Amazon Comprehend.

```
Hello Zhang Wei, I am John. Your AnyCompany Financial Services,
```

```
LLC credit card account 1111-0000-1111-0008 has a minimum payment
of $24.53 that is due by July 31st. Based on your autopay settings,
we will withdraw your payment on the due date from your
bank account number XXXXXX1111 with the routing number XXXXX0000.

Your latest statement was mailed to 100 Main Street, Any City,
WA 98121.
After your payment is received, you will receive a confirmation
text message at 206-555-0100.
If you have questions about your bill, AnyCompany Customer Service
is available by phone at 206-555-0199 or
email at support@anycompany.com.
```

Untuk mengunggah file ke ember

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Dalam daftar Bucket, pilih nama bucket yang Anda buat di Langkah 1 (misalnya,**`tutorial-bucket`**) untuk mengunggah file Anda.
4. Di tab Objek untuk bucket Anda, pilih Unggah.
5. Pada halaman Unggah, di bawah File dan folder, pilih Tambahkan file.
6. Pilih file yang akan diunggah, lalu pilih Buka. Misalnya, Anda dapat mengunggah contoh `tutorial.txt` file yang disebutkan sebelumnya.
7. Pilih Unggah.

Langkah 3: Buat titik akses S3

Untuk menggunakan Titik Akses Lambda Objek S3 untuk mengakses dan mengubah data asli, Anda harus membuat titik akses S3 dan mengaitkannya dengan bucket S3 yang Anda buat di Langkah 1. Titik akses harus Wilayah AWS sama dengan objek yang ingin Anda ubah.

Kemudian dalam tutorial ini, Anda akan menggunakan titik akses ini sebagai titik akses pendukung untuk Object Lambda Access Point Anda.

Untuk membuat titik akses

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

2. Pada panel navigasi di kiri, pilih Titik Akses.
3. Pada halaman Titik Akses, pilih Buat titik akses.
4. Di bidang Nama titik akses, masukkan nama (misalnya,**`tutorial-pii-access-point`**) untuk titik akses.

   Untuk informasi lebih lanjut tentang penamaan titik akses, lihat Aturan penamaan untuk titik akses.
5. Di bidang Sumber data, masukkan nama bucket yang Anda buat di Langkah 1 (misalnya,**`tutorial-bucket`**). S3 melampirkan titik akses ke bucket ini.

   (Opsional) Anda dapat memilih Browse S3 untuk menelusuri dan mencari bucket di akun Anda. Jika Anda memilih Browse S3, pilih bucket yang diinginkan, lalu pilih Pilih jalur untuk mengisi bidang nama Bucket dengan nama bucket tersebut.
6. Untuk asal Jaringan, pilih Internet.

   Untuk informasi lebih lanjut tentang asal jaringan untuk titik akses, lihat Membuat titik akses terbatas pada cloud privat virtual.
7. Secara default, semua pengaturan akses publik blok diaktifkan untuk titik akses Anda. Kami menyarankan agar Anda tetap mengaktifkan Blokir semua akses publik. Untuk informasi selengkapnya, lihat Mengelola akses publik ke titik akses untuk ember tujuan umum.
8. Untuk semua pengaturan titik akses lainnya, pertahankan pengaturan default.

   (Opsional) Anda dapat mengubah pengaturan titik akses untuk mendukung kasus penggunaan Anda. Untuk tutorial ini, kami sarankan untuk menjaga pengaturan default.

   (Opsional) Jika Anda perlu mengelola akses ke titik akses Anda, Anda dapat menentukan kebijakan titik akses. Untuk informasi selengkapnya, lihat Contoh kebijakan untuk titik akses.
9. Pilih Buat titik akses.

Langkah 4: Konfigurasikan dan terapkan fungsi Lambda bawaan

Untuk menyunting data PII, konfigurasikan dan terapkan AWS Lambda fungsi bawaan `ComprehendPiiRedactionS3ObjectLambda` untuk digunakan dengan Titik Akses Lambda Objek S3 Anda.

Untuk mengkonfigurasi dan menyebarkan fungsi Lambda

1. Masuk ke AWS Management Console dan lihat `ComprehendPiiRedactionS3ObjectLambda`fungsi di AWS Serverless Application Repository.
2. Untuk pengaturan Aplikasi, di bawah Nama aplikasi, pertahankan nilai default (`ComprehendPiiRedactionS3ObjectLambda`) untuk tutorial ini.

   (Opsional) Anda dapat memasukkan nama yang ingin Anda berikan ke aplikasi ini. Anda mungkin ingin melakukan ini jika Anda berencana untuk mengonfigurasi beberapa fungsi Lambda untuk kebutuhan akses yang berbeda untuk kumpulan data bersama yang sama.
3. Untuk MaskCharacter, pertahankan nilai default (**`*`**). Karakter topeng menggantikan setiap karakter dalam entitas PII yang disunting.
4. Untuk MaskMode, pertahankan nilai default (MASK). MaskModeNilai menentukan apakah entitas PII disunting dengan `MASK` karakter atau nilai. `PII_ENTITY_TYPE`
5. Untuk menyunting tipe data yang ditentukan, untuk PiiEntityTypes, pertahankan nilai default ALL. PiiEntityTypesNilai menentukan jenis entitas PII yang akan dipertimbangkan untuk redaksi.

   Untuk informasi selengkapnya tentang daftar jenis entitas PII yang didukung, lihat Mendeteksi Informasi Identifikasi Pribadi (PII) di Panduan Pengembang Amazon Comprehend.
6. Pertahankan pengaturan yang tersisa disetel ke default.

   (Opsional) Jika Anda ingin mengonfigurasi pengaturan tambahan untuk kasus penggunaan spesifik Anda, lihat bagian File Readme di sisi kiri halaman.
7. Pilih kotak centang di sebelah Saya mengakui bahwa aplikasi ini membuat peran IAM khusus.
8. Pilih Deploy.
9. Pada halaman aplikasi baru, di bawah Resources, pilih ID Logis dari fungsi Lambda yang Anda gunakan untuk meninjau fungsi pada halaman fungsi Lambda.

Langkah 5: Buat Titik Akses Lambda Objek S3

Titik Akses Lambda Objek S3 memberikan fleksibilitas untuk menjalankan fungsi Lambda langsung dari permintaan S3 GET sehingga fungsi tersebut dapat menyunting data PII yang diambil dari titik akses S3. Saat membuat dan mengonfigurasi Titik Akses Lambda Objek S3, Anda harus menentukan fungsi Lambda penyuntingan untuk memanggil dan menyediakan konteks acara dalam format JSON sebagai parameter khusus untuk digunakan Lambda.

Konteks acara memberikan informasi tentang permintaan yang dibuat dalam acara yang diteruskan dari S3 Object Lambda ke Lambda. Untuk informasi selengkapnya tentang semua bidang dalam konteks acara, lihatFormat konteks peristiwa dan penggunaan.

Untuk membuat Objek S3 Lambda Access Point

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi sebelah kiri, pilih Titik Akses Lambda Objek.
3. Pada halaman Titik Akses Lambda Objek, pilih Buat Titik Akses Lambda Objek.
4. Untuk nama Object Lambda Access Point, masukkan nama yang ingin Anda gunakan untuk Object Lambda Access Point (misalnya,). **`tutorial-pii-object-lambda-accesspoint`**
5. Untuk Mendukung Titik Akses, masukkan atau telusuri ke titik akses standar yang Anda buat di Langkah 3 (misalnya,**`tutorial-pii-access-point`**), lalu pilih Pilih Titik Akses pendukung.
6. Untuk S3 APIs, untuk mengambil objek dari bucket S3 agar fungsi Lambda dapat diproses, pilih. GetObject
7. Untuk fungsi Invoke Lambda, Anda dapat memilih salah satu dari dua opsi berikut untuk tutorial ini.
   - Pilih Pilih dari fungsi di akun Anda dan pilih fungsi Lambda yang Anda gunakan di Langkah 4 (misalnya,**`serverlessrepo-ComprehendPiiRedactionS3ObjectLambda`**) dari daftar dropdown fungsi Lambda.
   - Pilih Masukkan ARN, lalu masukkan Nama Sumber Daya Amazon (ARN) dari fungsi Lambda yang Anda buat di Langkah 4.
8. Untuk versi fungsi Lambda, pilih $LATEST (versi terbaru dari fungsi Lambda yang Anda gunakan di Langkah 4).
9. (Opsional) Jika Anda memerlukan fungsi Lambda Anda untuk mengenali dan memproses permintaan GET dengan header rentang dan nomor bagian, pilih Fungsi Lambda mendukung permintaan menggunakan rentang dan fungsi Lambda mendukung permintaan menggunakan nomor bagian. Jika tidak, kosongkan dua kotak centang ini.

   Untuk informasi selengkapnya tentang cara menggunakan nomor rentang atau bagian dengan S3 Object Lambda, lihat. Bekerja dengan header Range dan partNumber
10. (Opsional) Di bawah Payload - opsional, tambahkan teks JSON untuk menyediakan fungsi Lambda Anda dengan informasi tambahan.

Payload adalah teks JSON opsional yang dapat Anda berikan ke fungsi Lambda Anda sebagai masukan untuk semua pemanggilan yang berasal dari Titik Akses Lambda Objek S3 tertentu. Untuk menyesuaikan perilaku untuk beberapa Titik Akses Lambda Objek yang menjalankan fungsi Lambda yang sama, Anda dapat mengonfigurasi muatan dengan parameter yang berbeda, sehingga memperluas fleksibilitas fungsi Lambda Anda.

Untuk informasi selengkapnya tentang payload, lihatFormat konteks peristiwa dan penggunaan.

11. (Opsional) Untuk metrik Permintaan - opsional, pilih Nonaktifkan atau Aktifkan untuk menambahkan pemantauan Amazon S3 ke Titik Akses Lambda Objek Anda. Metrik permintaan ditagih dengan tarif Amazon CloudWatch standar. Untuk informasi selengkapnya, lihat harga CloudWatch.
12. Di bawah kebijakan Titik Akses Objek Lambda - opsional, pertahankan pengaturan default.

    (Opsional) Anda dapat menetapkan kebijakan sumber daya. Kebijakan resource ini memberikan izin `GetObject` API untuk menggunakan Object Lambda Access Point yang ditentukan.
13. Pertahankan pengaturan yang tersisa disetel ke default, dan pilih Buat Objek Lambda Access Point.

Langkah 6: Gunakan S3 Object Lambda Access Point untuk mengambil file yang disunting

Sekarang, S3 Object Lambda siap untuk menyunting data PII dari file asli Anda.

Untuk menggunakan S3 Object Lambda Access Point untuk mengambil file yang disunting

Saat Anda meminta untuk mengambil file melalui Titik Akses Lambda Objek S3, Anda membuat panggilan `GetObject` API ke S3 Object Lambda. S3 Object Lambda memanggil fungsi Lambda untuk menyunting data PII Anda dan mengembalikan data yang diubah sebagai respons terhadap panggilan API S3 standar. `GetObject`

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi sebelah kiri, pilih Titik Akses Lambda Objek.
3. Pada halaman Titik Akses Objek Lambda, pilih Titik Akses Lambda Objek S3 yang Anda buat di Langkah 5 (misalnya,). **`tutorial-pii-object-lambda-accesspoint`**
4. Pada tab Objek pada Titik Akses Lambda Objek S3 Anda, pilih file yang memiliki nama yang sama (misalnya`tutorial.txt`,) dengan file yang Anda unggah ke bucket S3 di Langkah 2.

File ini harus berisi semua data yang diubah.

5. Untuk melihat data yang diubah, pilih Buka atau Unduh.

   Anda harus dapat melihat file yang disunting, seperti yang ditunjukkan pada contoh berikut.

```
Hello *********. Your AnyCompany Financial Services,
LLC credit card account ******************** has a minimum payment
of $24.53 that is due by *********. Based on your autopay settings,
we will withdraw your payment on the due date from your
bank account ********** with the routing number *********.

Your latest statement was mailed to ********************************.
After your payment is received, you will receive a confirmation
text message at ************.
If you have questions about your bill, AnyCompany Customer Service
is available by phone at ************ or
email at **********************.
```

Langkah 7: Bersihkan

Jika Anda menyunting data Anda melalui S3 Object Lambda hanya sebagai latihan pembelajaran, hapus AWS sumber daya yang Anda alokasikan sehingga Anda tidak lagi dikenakan biaya.

Sublangkah

- Hapus Titik Akses Objek Lambda
- Hapus titik akses S3
- Hapus fungsi Lambda
- Hapus grup CloudWatch log
- Hapus file asli di bucket sumber S3
- Hapus bucket sumber S3
- Hapus peran IAM untuk fungsi Lambda Anda
- Menghapus kebijakan terkelola pelanggan untuk pengguna IAM Anda
- Hapus pengguna IAM

Hapus Titik Akses Objek Lambda

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi sebelah kiri, pilih Titik Akses Lambda Objek.
3. Pada halaman Titik Akses Objek Lambda, pilih tombol opsi di sebelah kiri Titik Akses Lambda Objek S3 yang Anda buat di Langkah 5 (misalnya,). **`tutorial-pii-object-lambda-accesspoint`**
4. Pilih Hapus.
5. Konfirmasikan bahwa Anda ingin menghapus Titik Akses Objek Lambda Anda dengan memasukkan namanya di bidang teks yang muncul, lalu pilih Hapus.

Hapus titik akses S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Pada panel navigasi di kiri, pilih Titik Akses.
3. Arahkan ke titik akses yang Anda buat di Langkah 3 (misalnya,**`tutorial-pii-access-point`**), dan pilih tombol opsi di sebelah nama titik akses.
4. Pilih Hapus.
5. Konfirmasikan bahwa Anda ingin menghapus jalur akses Anda dengan memasukkan namanya di bidang teks yang muncul, lalu pilih Hapus.

Hapus fungsi Lambda

1. Di AWS Lambda konsol di https://console.aws.amazon.com/lambda/, pilih Fungsi di panel navigasi kiri.
2. Pilih fungsi yang Anda buat di Langkah 4 (misalnya,**`serverlessrepo-ComprehendPiiRedactionS3ObjectLambda`**).
3. Pilih Tindakan, lalu pilih Hapus.
4. Di kotak dialog Hapus fungsi, pilih Hapus.

Hapus grup CloudWatch log

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.

2. Pada panel navigasi kiri, pilih Grup log.
3. Temukan grup log yang namanya diakhiri dengan fungsi Lambda yang Anda buat di Langkah 4 (misalnya,**`serverlessrepo-ComprehendPiiRedactionS3ObjectLambda`**).
4. Pilih Tindakan, lalu pilih Hapus grup log.
5. Di kotak dialog Hapus grup log, pilih Hapus.

Hapus file asli di bucket sumber S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Dalam daftar nama Bucket, pilih nama bucket tempat Anda mengunggah file asli di Langkah 2 (misalnya,**`tutorial-bucket`**).
4. Pilih kotak centang di sebelah kiri nama objek yang ingin Anda hapus (misalnya,`tutorial.txt`).
5. Pilih Hapus.
6. Pada halaman Hapus objek, di objek Hapus secara permanen? bagian, konfirmasikan bahwa Anda ingin menghapus objek ini dengan memasukkan **`permanently delete`** di kotak teks.
7. Pilih Hapus objek.

Hapus bucket sumber S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Dalam daftar Bucket, pilih tombol opsi di sebelah nama bucket yang Anda buat di Langkah 1 (misalnya,**`tutorial-bucket`**).
4. Pilih Hapus.
5. Di halaman Hapus bucket, konfirmasikan bahwa Anda ingin menghapus bucket dengan memasukkan nama bucket ke dalam bidang teks, lalu pilih Hapus bucket.

Hapus peran IAM untuk fungsi Lambda Anda

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi kiri, pilih Peran, lalu pilih kotak centang di samping nama peran yang ingin Anda hapus. Nama peran dimulai dengan nama fungsi Lambda yang Anda gunakan di Langkah 4 (misalnya,). **`serverlessrepo-ComprehendPiiRedactionS3ObjectLambda`**
3. Pilih Hapus.
4. Di kotak dialog Hapus, masukkan nama peran di bidang input teks untuk mengonfirmasi penghapusan. Lalu, pilih Hapus.

Menghapus kebijakan terkelola pelanggan untuk pengguna IAM Anda

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi di sebelah kiri, pilih Kebijakan.
3. Pada halaman Kebijakan, masukkan nama kebijakan terkelola pelanggan yang Anda buat di Prasyarat (misalnya,**`tutorial-serverless-application-repository`**) di kotak pencarian untuk memfilter daftar kebijakan. Pilih tombol opsi di sebelah nama kebijakan yang ingin Anda hapus.
4. Pilih Tindakan, lalu pilih Hapus.
5. Konfirmasikan bahwa Anda ingin menghapus kebijakan ini dengan memasukkan namanya di bidang teks yang muncul, lalu pilih Hapus.

Hapus pengguna IAM

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi kiri, pilih Pengguna, lalu pilih kotak centang di sebelah nama pengguna yang ingin Anda hapus.
3. Pilih Hapus di bagian atas halaman.
4. Di Hapus***`user name`***? kotak dialog, masukkan nama pengguna di bidang input teks untuk mengonfirmasi penghapusan pengguna. Pilih Hapus.

Langkah selanjutnya

Setelah menyelesaikan tutorial ini, Anda dapat menjelajahi lebih lanjut kasus penggunaan terkait berikut:

- Anda dapat membuat beberapa Titik Akses Lambda Objek S3 dan mengaktifkannya dengan fungsi Lambda bawaan yang dikonfigurasi secara berbeda untuk menyunting jenis PII tertentu tergantung pada kebutuhan bisnis aksesori data.

  Setiap jenis pengguna mengasumsikan peran IAM dan hanya memiliki akses ke satu Titik Akses Lambda Objek S3 (dikelola melalui kebijakan IAM). Kemudian, Anda melampirkan setiap fungsi `ComprehendPiiRedactionS3ObjectLambda` Lambda yang dikonfigurasi untuk kasus penggunaan redaksi yang berbeda ke Titik Akses Lambda Objek S3 yang berbeda. Untuk setiap Titik Akses Lambda Objek S3, Anda dapat memiliki jalur akses S3 pendukung untuk membaca data dari bucket S3 yang menyimpan kumpulan data bersama.

  Untuk informasi selengkapnya tentang cara membuat kebijakan bucket S3 yang memungkinkan pengguna membaca dari bucket hanya melalui jalur akses S3, lihat. Mengonfigurasi kebijakan IAM untuk menggunakan titik akses

  Untuk informasi selengkapnya tentang cara memberikan izin pengguna untuk mengakses fungsi Lambda, titik akses S3, dan Titik Akses Lambda Objek S3, lihat. Mengonfigurasi kebijakan IAM untuk Titik Akses Lambda Objek
- Anda dapat membangun fungsi Lambda Anda sendiri dan menggunakan S3 Object Lambda dengan fungsi Lambda yang disesuaikan untuk memenuhi kebutuhan data spesifik Anda.

  Misalnya, untuk menjelajahi berbagai nilai data, Anda dapat menggunakan Lambda Objek S3 dan fungsi Lambda Anda sendiri yang menggunakan fitur Amazon Comprehend tambahan, seperti pengenalan entitas, pengenalan frasa kunci, analisis sentimen, dan klasifikasi dokumen, untuk memproses data. Anda juga dapat menggunakan S3 Object Lambda bersama dengan Amazon Comprehend Medical, layanan NLP yang memenuhi syarat HIPAA, untuk menganalisis dan mengekstrak data dengan cara yang sadar konteks.

  Untuk informasi selengkapnya tentang cara mengubah data dengan S3 Object Lambda dan fungsi Lambda Anda sendiri, lihat. Tutorial: Mengubah data untuk aplikasi Anda dengan S3 Object Lambda

Debugging dan pemecahan masalah S3 Object Lambda

Permintaan ke titik akses Lambda Objek Amazon S3 mungkin menghasilkan respons kesalahan baru ketika terjadi kesalahan dengan invokasi atau eksekusi fungsi Lambda. Kesalahan ini mengikuti format yang sama dengan kesalahan standar Amazon S3. Untuk informasi tentang kesalahan S3 Lambda Objek, lihat Daftar Kode Kesalahan S3 Lambda Objek DALAM Referensi API Amazon Simple Storage Service.

Untuk informasi selengkapnya tentang fungsi Lambda umum debugging, lihat Pemantauan dan pemecahan masalah aplikasi DALAM Panduan Pengembang AWS Lambda .

Untuk informasi tentang kesalahan Amazon S3 standar, lihat Respons kesalahan dalam Referensi API Layanan Amazon Simple Storage Service.

Anda dapat mengaktifkan metrik permintaan di Amazon CloudWatch untuk Titik Akses Objek Lambda Anda. Metrik ini membantu Anda untuk memantau performa operasional titik akses Anda. Anda dapat mengaktifkan metrik permintaan selama, atau setelah pembuatan Titik Akses Lambda Objek. Untuk informasi selengkapnya, lihat Metrik permintaan Lambda Objek S3 di CloudWatch.

Untuk mendapatkan pencatatan log yang lebih terperinci tentang permintaan yang dibuat ke Titik Akses Lambda Objek Anda, Anda dapat mengaktifkan AWS CloudTrail peristiwa data. Untuk informasi lebih lanjut, lihat Peristiwa Pencatatan Data untuk Pelacakan dalam AWS CloudTrail Panduan Pengguna.

Untuk tutorial S3 Lambda Objek, lihat hal berikut ini:

- Tutorial: Mengubah data untuk aplikasi Anda dengan S3 Object Lambda
- Tutorial: Mendeteksi dan menyunting data PII dengan Objek S3 Lambda dan Amazon Comprehend
- Tutorial: Menggunakan S3 Lambda Objek untuk menandai gambar secara dinamis saat gambarnya diambil

Untuk informasi lebih lanjut tentang titik akses standar, lihat Mengelola akses ke kumpulan data bersama dengan titik akses.

Untuk informasi tentang bekerja dengan bucket, lihat Ikhtisar ember tujuan umum. Untuk informasi tentang bekerja dengan objek, lihat Gambaran umum objek Amazon S3.

Melakukan operasi objek secara massal dengan Operasi Batch

Anda dapat menggunakan Operasi Batch S3 untuk melakukan operasi batch berskala besar pada objek Amazon S3. Operasi Batch S3 dapat melakukan satu operasi untuk sejumlah objek Amazon S3 yang telah Anda tentukan. Satu pekerjaan dapat melakukan operasi tertentu pada miliaran objek yang berisi data berukuran exabyte. Amazon S3 melacak kemajuan, mengirim pemberitahuan, dan menyimpan laporan penyelesaian terperinci dari semua tindakan, menyediakan pengalaman yang sepenuhnya terkelola, dapat diaudit, dan nirserver. Anda dapat menggunakan Operasi Batch S3 melalui konsol Amazon S3 AWS CLI, AWS SDKs, atau Amazon S3 REST API.

Gunakan Operasi Batch S3 untuk menyalin objek dan mengatur tag objek atau daftar kontrol akses (ACLs). Anda juga dapat memulai pemulihan objek dari S3 Glacier Flexible Retrieval atau menginvokasi fungsi AWS Lambda untuk melakukan tindakan khusus menggunakan objek Anda. Anda dapat melakukan operasi ini pada daftar objek kustom, atau menggunakan laporan inventaris Amazon S3 untuk membuat daftar objek dengan mudah. Operasi Batch Amazon S3 menggunakan operasi API Amazon S3 yang sama yang sudah Anda gunakan dengan Amazon S3.

> **Note**
>
> Untuk informasi selengkapnya tentang menggunakan kelas penyimpanan Amazon S3 Express One Zone dengan bucket direktori, lihat S3 Express One Zone dan Bekerja dengan bucket direktori. Untuk informasi selengkapnya tentang penggunaan Operasi Batch dengan S3 Express One Zone dan bucket direktori, lihat Menggunakan Operasi Batch dengan bucket direktori.

Dasar-dasar Operasi Batch S3

Anda dapat menggunakan Operasi Batch S3 untuk melakukan operasi batch berskala besar pada objek Amazon S3. Operasi Batch S3 dapat menjalankan operasi tunggal atau tindakan pada sejumlah objek Amazon S3 yang telah Anda tentukan.

Terminologi

Bagian ini menggunakan istilah manifes, pekerjaan, operasi, dan tugas, yang didefinisikan sebagai berikut:

Manifes

Manifes adalah objek Amazon S3 yang berisi kunci objek yang harus ditindaklanjuti Amazon S3. Jika Anda ingin membuat pekerjaan Operasi Batch, Anda harus menyediakan manifes. Manifes buatan pengguna Anda harus berisi nama bucket, kunci objek, dan secara opsional, versi objek untuk setiap objek. Jika Anda menyediakan manifes buatan pengguna, manifes tersebut harus dalam bentuk laporan Inventaris Amazon S3 atau file CSV.

Anda juga dapat meminta Amazon S3 menghasilkan manifes secara otomatis berdasarkan kriteria filter objek yang Anda tentukan saat membuat pekerjaan. Opsi ini tersedia untuk pekerjaan Replikasi Batch S3 yang Anda buat di konsol Amazon S3, atau untuk jenis pekerjaan apa pun yang Anda buat dengan menggunakan AWS Command Line Interface AWS CLI() AWS SDKs, atau Amazon S3 REST API.

Pekerjaan

Pekerjaan adalah unit kerja dasar untuk Operasi Batch S3. Pekerjaan berisi semua informasi yang diperlukan untuk menjalankan operasi tertentu pada objek yang tercantum dalam manifes. Setelah Anda memberikan informasi ini dan meminta agar pekerjaan tersebut dimulai, pekerjaan akan menjalankan operasi untuk setiap objek dalam manifes.

Operasi

Operasi adalah jenis tindakan API, seperti menyalin objek yang akan dijalankan pekerjaan Operasi Batch. Setiap pekerjaan melakukan satu jenis operasi di semua objek yang telah ditentukan dalam manifes.

Tugas

Tugas adalah unit pelaksanaan pekerjaan. Tugas mewakili satu panggilan ke operasi Amazon S3 atau AWS Lambda API untuk melakukan operasi pekerjaan pada satu objek. Selama masa aktif pekerjaan, Operasi Batch S3 membuat satu tugas untuk setiap objek yang ditentukan dalam manifes.

Cara kerja pekerjaan Operasi Batch S3

Pekerjaan adalah unit kerja dasar untuk Operasi Batch S3. Pekerjaan berisi semua informasi yang diperlukan untuk menjalankan operasi tertentu pada sejumlah objek. Untuk membuat pekerjaan, Anda memberikan daftar objek kepada Operasi Batch S3 dan menentukan tindakan yang harus dilakukan pada objek tersebut.

Untuk informasi tentang operasi yang didukung Operasi Batch S3, lihat Operasi yang didukung oleh Operasi Batch S3.

Pekerjaan batch melakukan operasi tertentu pada setiap objek yang disertakan dalam manifesnya. Manifes mencantumkan objek yang harus diproses pekerjaan batch dan disimpan sebagai objek dalam bucket. Anda dapat menggunakan laporan Membuat katalog dan menganalisis data Anda dengan S3 Inventory berformat nilai yang dipisahkan koma (CSV) sebagai manifes, yang memudahkan pembuatan daftar besar objek yang berada di bucket. Anda juga dapat menentukan manifes dalam format CSV sederhana yang memungkinkan Anda melakukan operasi batch pada daftar objek yang telah disesuaikan yang dimuat dalam satu bucket.

Setelah Anda membuat pekerjaan, Amazon S3 memproses daftar objek dalam manifes dan menjalankan operasi yang telah ditentukan terhadap setiap objek. Saat pekerjaan berjalan, Anda dapat memantau kemajuannya secara terprogram atau melalui konsol Amazon S3. Anda juga dapat mengonfigurasi pekerjaan untuk menghasilkan laporan penyelesaian setelah selesai. Laporan penyelesaian menguraikan hasil dari setiap tugas yang dilakukan oleh pekerjaan. Untuk informasi selengkapnya tentang cara memantau pekerjaan, lihat Mengelola tugas Operasi Batch S3.

Ada biaya yang terkait dengan Operasi Batch S3. Anda ditagih untuk membuat pekerjaan Operasi Batch, termasuk pekerjaan yang dibatalkan sebelum selesai. Untuk informasi selengkapnya, lihat Harga Amazon S3.

S3Pekerjaan Operasi Batch secara default dapat memproses hingga 4 miliar objek untuk semua operasi. Secara khusus Copy, Object Tagging, Object Lock, memanggil AWS Lambda fungsi, dan pekerjaan Batch Replication dapat mendukung hingga 20 miliar objek. Ada batasan 6 pekerjaan Replikasi Batch aktif per AWS akun. Untuk mulai membuat pekerjaan Operasi Batch, lihat Membuat pekerjaan Operasi Batch S3.

Tutorial Operasi Batch S3

Tutorial berikut menyajikan end-to-end prosedur lengkap untuk beberapa tugas Operasi Batch.

- Tutorial: Video transcoding batch dengan Operasi Batch S3

Memberikan izin untuk Batch Operations

Sebelum membuat dan menjalankan pekerjaan Operasi Batch S3, Anda harus memberikan izin yang diperlukan. Untuk membuat pekerjaan Operasi Batch Amazon S3, izin pengguna `s3:CreateJob` diperlukan. Entitas yang sama yang membuat pekerjaan juga harus memiliki `iam:PassRole` izin

untuk meneruskan peran AWS Identity and Access Management (IAM) yang ditentukan untuk pekerjaan tersebut ke Operasi Batch.

Untuk informasi umum tentang menentukan sumber daya IAM, lihat Kebijakan JSON IAM, Elemen sumber daya di Panduan Pengguna IAM. Bagian berikut memberikan informasi tentang membuat peran IAM dan melampirkan kebijakan.

Topik

- Membuat peran IAM Operasi Batch S3
- Melampirkan kebijakan izin

Membuat peran IAM Operasi Batch S3

Amazon S3 harus memiliki izin untuk menjalankan Operasi Batch S3 atas nama Anda. Anda memberikan izin ini melalui peran AWS Identity and Access Management (IAM). Bagian ini memberikan contoh kebijakan kepercayaan dan izin yang Anda gunakan saat membuat peran IAM. Untuk informasi selengkapnya, lihat peran IAM dalam Panduan Pengguna IAM. Sebagai contoh, lihat Mengontrol izin untuk Operasi Batch menggunakan tag pekerjaan dan Menyalin objek menggunakan Operasi Batch S3.

Dalam kebijakan IAM, Anda juga dapat menggunakan kunci persyaratan untuk memfilter izin akses yang akan digunakan untuk pekerjaan Operasi Batch S3. Untuk informasi selengkapnya dan daftar lengkap kunci kondisi khusus Amazon S3, lihat Kunci tindakan, sumber daya, dan kondisi untuk Amazon S3 di Referensi Otorisasi Layanan.

Untuk informasi selengkapnya tentang izin operasi S3 API menurut jenis sumber daya S3, lihat. Izin yang diperlukan untuk operasi API Amazon S3

Kebijakan kepercayaan

Untuk mengizinkan pengguna utama layanan Operasi Batch S3 agar dapat menjalankan peran IAM, lampirkan kebijakan kepercayaan berikut ke peran tersebut.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"batchoperations.s3.amazonaws.com"
```

```
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

Melampirkan kebijakan izin

Tergantung pada jenis operasinya, Anda dapat melampirkan salah satu kebijakan berikut.

Sebelum Anda mengonfigurasi izin, perhatikan hal berikut ini:

- Apa pun operasinya, Amazon S3 tetap memerlukan izin untuk membaca objek manifes Anda dari bucket S3 dan menulis laporan ke bucket Anda secara opsional. Oleh karena itu, semua kebijakan berikut mencakup izin ini.
- Untuk manifes laporan inventaris Amazon S3, Operasi Batch S3 memerlukan izin untuk membaca objek manifest.json dan semua file data CSV terkait.
- Izin khusus versi seperti `s3:GetObjectVersion` hanya diperlukan saat Anda menentukan ID versi objek.
- Jika Anda menjalankan Operasi Batch S3 pada objek terenkripsi, peran IAM juga harus memiliki akses ke AWS KMS kunci yang digunakan untuk mengenkripsi mereka.
- Jika Anda mengirimkan manifes laporan inventaris yang dienkripsi AWS KMS, kebijakan IAM Anda harus menyertakan izin `"kms:Decrypt"` dan `"kms:GenerateDataKey"` untuk objek manifest.json serta semua file data CSV terkait.
- Jika pekerjaan Operasi Batch menghasilkan manifes dalam bucket yang mengaktifkan daftar kontrol akses (ACLs) dan berada di posisi lain Akun AWS, Anda harus memberikan `s3:PutObjectAcl` izin dalam kebijakan IAM untuk peran IAM yang dikonfigurasi untuk pekerjaan batch. Jika Anda tidak menyertakan izin ini, pekerjaan batch gagal dengan kesalahan`Error occurred when preparing manifest: Failed to write manifest`.

Salin objek: PutObject

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:PutObject",
```

```
                "s3:PutObjectAcl",
                "s3:PutObjectTagging"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
        },
        {
            "Action": [
                "s3:GetObject",
                "s3:GetObjectAcl",
                "s3:GetObjectTagging",
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-source-bucket",
                "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-completion-report-bucket/*"
            ]
        }
    ]
}
```

Ganti penandaan objek: PutObjectTagging

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutObjectTagging",
        "s3:PutObjectVersionTagging"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
      ]
    },
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutObject"
      ],
      "Resource":[
        "arn:aws:s3:::amzn-s3-demo-completion-report-bucket/*"
      ]
    }
  ]
}
```

Hapus penandaan objek: DeleteObjectTagging

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
```

```
              "s3:DeleteObjectTagging",
              "s3:DeleteObjectVersionTagging"
          ],
          "Resource": [
              "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
          ]
      },
      {
          "Effect": "Allow",
          "Action": [
              "s3:GetObject",
              "s3:GetObjectVersion"
          ],
          "Resource": [
              "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
          ]
      },
      {
          "Effect": "Allow",
          "Action": [
              "s3:PutObject"
          ],
          "Resource": [
              "arn:aws:s3:::amzn-s3-demo-completion-report-bucket/*"
          ]
      }
  ]
}
```

Ganti daftar kontrol akses: PutObjectAcl

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutObjectAcl",
        "s3:PutObjectVersionAcl"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
    },
    {
```

```
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
      ]
    },
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutObject"
      ],
      "Resource":[
        "arn:aws:s3:::amzn-s3-demo-completion-report-bucket/*"
      ]
    }
  ]
}
```

Kembalikan objek: RestoreObject

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
          "s3:RestoreObject"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
      ]
    },
```

```
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutObject"
      ],
      "Resource":[
        "arn:aws:s3:::amzn-s3-demo-completion-report-bucket/*"
      ]
    }
  ]
}
```

Terapkan retensi Object Lock: PutObjectRetention

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:GetBucketObjectLockConfiguration",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-destination-bucket"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObjectRetention",
                "s3:BypassGovernanceRetention"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
            ]
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-completion-report-bucket/*"
            ]
        }
    ]
}
```

Terapkan penahanan hukum Object Lock: PutObjectLegalHold

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:GetBucketObjectLockConfiguration",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-destination-bucket"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "s3:PutObjectLegalHold",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
            ]
        },
        {
```

```
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-completion-report-bucket/*"
            ]
        }
    ]
}
```

Replikasi objek yang ada: InitiateReplication dengan manifes yang dihasilkan S3

Gunakan kebijakan ini jika Anda menggunakan dan menyimpan manifes yang dihasilkan S3. Untuk informasi selengkapnya tentang menggunakan Operasi Batch untuk mereplikasi objek yang ada, lihatMereplikasi objek yang ada dengan Batch Replication.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Action":[
            "s3:InitiateReplication"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
         ]
      },
      {
         "Action":[
            "s3:GetReplicationConfiguration",
            "s3:PutInventoryConfiguration"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket"
         ]
      },
      {
         "Action":[
            "s3:GetObject",
            "s3:GetObjectVersion"
```

```
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject"
         ],
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-completion-report-bucket/*",
            "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
         ]
      }
   ]
}
```

Replikasi objek yang ada: InitiateReplication dengan manifes pengguna

Gunakan kebijakan ini jika Anda menggunakan manifes yang disediakan pengguna. Untuk informasi selengkapnya tentang menggunakan Operasi Batch untuk mereplikasi objek yang ada, lihatMereplikasi objek yang ada dengan Batch Replication.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Action":[
            "s3:InitiateReplication"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
         ]
      },
      {
         "Action":[
            "s3:GetObject",
            "s3:GetObjectVersion"
         ],
         "Effect":"Allow",
```

```
            "Resource":[
               "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
            ]
         },
         {
            "Effect":"Allow",
            "Action":[
               "s3:PutObject"
            ],
            "Resource":[
               "arn:aws:s3:::amzn-s3-demo-completion-report-bucket/*"
            ]
         }
      ]
}
```

Membuat pekerjaan Operasi Batch S3

Dengan Operasi Batch Amazon S3, Anda dapat melakukan operasi batch berskala besar pada daftar objek Amazon S3 tertentu. Bagian ini menjelaskan informasi yang Anda perlukan untuk membuat pekerjaan Operasi Batch S3 dan hasil permintaan `CreateJob`. Ini juga menyediakan instruksi untuk membuat pekerjaan Operasi Batch dengan menggunakan konsol Amazon S3, AWS Command Line Interface (AWS CLI), dan. AWS SDK untuk Java

Saat membuat pekerjaan Operasi Batch S3, Anda dapat meminta laporan penyelesaian untuk semua tugas atau hanya untuk tugas yang gagal. Selama setidaknya ada satu tugas yang berhasil diinvokasi, Operasi Batch S3 akan menghasilkan laporan untuk pekerjaan yang telah selesai, gagal, atau dibatalkan. Untuk informasi selengkapnya, lihat Contoh: Laporan penyelesaian Operasi Batch S3.

Topik

- Elemen permintaan pekerjaan Operasi Batch
- Menentukan manifes

Elemen permintaan pekerjaan Operasi Batch

Untuk membuat pekerjaan Operasi Batch S3, Anda harus memberikan informasi berikut:

Operasi

Tentukan operasi yang akan dijalankan Operasi Batch S3 terhadap objek dalam manifes. Setiap jenis operasi menerima parameter khusus untuk operasi tersebut. Dengan Operasi Batch, Anda dapat melakukan operasi secara massal, dengan hasil yang sama seperti jika Anda melakukan operasi itu one-by-one pada setiap objek.

Manifes

Manifes adalah daftar semua objek yang harus dijalankan Operasi Batch S3. Anda dapat menggunakan metode berikut untuk menentukan manifes yang akan digunakan dalam pekerjaan Operasi Batch:

- Buat daftar objek berformat CSV yang disesuaikan secara manual.
- Pilih laporan Membuat katalog dan menganalisis data Anda dengan S3 Inventory berformat CSV yang ada.
- Atur Operasi Batch agar menghasilkan manifes secara otomatis berdasarkan kriteria filter objek yang Anda tentukan saat membuat pekerjaan. Opsi ini tersedia untuk pekerjaan replikasi batch yang Anda buat di konsol Amazon S3, atau untuk jenis pekerjaan apa pun yang Anda buat dengan menggunakan AWS CLI AWS SDKs,, atau Amazon S3 REST API.

> **Note**
>
> - Terlepas dari cara Anda menentukan manifes, daftar itu harus disimpan di dalam sebuah bucket yang bersifat umum. Operasi Batch tidak dapat mengimpor manifes yang ada dari, atau menyimpan manifes yang dihasilkan ke bucket direktori. Objek yang dijelaskan dalam manifes tetap dapat disimpan dalam direktori bucket. Untuk informasi selengkapnya, lihat Bucket direktori.
> - Jika objek dalam manifes Anda berada dalam bucket berversi, menentukan versi IDs untuk objek akan mengarahkan Operasi Batch untuk melakukan operasi pada versi tertentu. Jika tidak IDs ada versi yang ditentukan, Operasi Batch melakukan operasi pada versi terbaru objek. Jika manifes menyertakan bidang ID versi, Anda harus memberikan ID versi untuk semua objek dalam manifes.

Untuk informasi selengkapnya, lihat Menentukan manifes.

Prioritas

Gunakan prioritas pekerjaan untuk menunjukkan prioritas relatif dari pekerjaan ini jika dibandingkan tugas lain yang berjalan di akun Anda. Angka yang lebih tinggi menunjukkan prioritas yang lebih tinggi.

Prioritas pekerjaan hanya memiliki makna relatif terhadap prioritas yang ditetapkan untuk pekerjaan lain di akun dan Wilayah yang sama. Anda bisa memilih sistem penomoran yang sesuai. Misalnya, Anda mungkin ingin menetapkan semua pekerjaan Restore (`RestoreObject`) prioritas 1, semua pekerjaan Copy (`CopyObject`) prioritas 2, dan semua pekerjaan Replace access control lists (ACLs**`PutObjectAcl`**) () prioritas 3.

Operasi Batch S3 memprioritaskan pekerjaan sesuai dengan urutan prioritas, tetapi tidak menjamin urutan prioritas secara teratur. Oleh karena itu, jangan gunakan prioritas pekerjaan untuk memastikan bahwa satu pekerjaan dimulai atau selesai sebelum pekerjaan lainnya. Jika Anda harus memastikan urutan secara teratur, tunggu hingga satu pekerjaan selesai sebelum memulai pekerjaan berikutnya.

RoleArn

Tentukan peran AWS Identity and Access Management (IAM) untuk menjalankan pekerjaan. Peran IAM yang Anda gunakan harus memiliki izin yang memadai untuk melakukan operasi yang ditentukan dalam pekerjaan. Misalnya, untuk menjalankan pekerjaan `CopyObject`, peran IAM harus memiliki izin `s3:GetObject` untuk bucket sumber dan izin `s3:PutObject` untuk bucket tujuan. Peran ini juga memerlukan izin untuk membaca manifes dan menulis laporan penyelesaian pekerjaan.

Untuk informasi selengkapnya tentang peran IAM, lihat Peran IAM dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang izin Amazon S3, lihat Tindakan kebijakan untuk Amazon S3.

> **Note**
>
> Pekerjaan Operasi Batch yang menjalankan tindakan pada bucket direktori memerlukan izin tertentu. Untuk informasi selengkapnya, lihat AWS Identity and Access Management (IAM) untuk S3 Express One Zone.

Laporan

Tentukan apakah Anda ingin Operasi Batch S3 membuat laporan penyelesaian. Jika meminta laporan penyelesaian pekerjaan, Anda juga harus memberikan parameter untuk laporan dalam elemen ini. Informasi yang diperlukan meliputi:

- Bucket tempat Anda ingin menyimpan laporan

> **Note**
>
> Laporan harus disimpan dalam bucket yang bersifat umum. Operasi Batch tidak dapat menyimpan laporan ke bucket direktori. Untuk informasi selengkapnya, lihat Bucket direktori.

- Format laporan
- Anda ingin laporan menyertakan detail semua tugas atau hanya tugas yang gagal
- String prefiks opsional

> **Note**
>
> Laporan penyelesaian selalu dienkripsi dengan enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3).

Tanda (opsional)

Anda dapat melabeli dan mengontrol akses ke pekerjaan Operasi Batch S3 Anda dengan menambahkan tag. Anda dapat menggunakan tanda untuk mengidentifikasi pihak yang bertanggung jawab atas pekerjaan Operasi Batch, atau untuk mengontrol cara pengguna berinteraksi dengan pekerjaan Operasi Batch. Adanya tanda pekerjaan dapat memberikan atau membatasi kemampuan pengguna untuk membatalkan pekerjaan, mengaktifkan pekerjaan dalam status konfirmasi, atau mengubah tingkat prioritas pekerjaan. Misalnya, Anda dapat memberikan izin pengguna untuk menginvokasi operasi `CreateJob`, asalkan pekerjaan tersebut dibuat dengan tanda `"Department=Finance"`.

Anda dapat membuat pekerjaan dengan tanda yang tersemat, dan dapat menambahkan tanda ke pekerjaan setelah Anda selesai membuatnya.

Untuk informasi selengkapnya, lihat the section called “Menggunakan tanda”.

Deskripsi (opsional)

Untuk melacak dan memantau pekerjaan, Anda juga dapat memberikan deskripsi hingga 256 karakter. Amazon S3 akan menyertakan deskripsi ini setiap memberikan informasi tentang pekerjaan atau menampilkan detail pekerjaan pada konsol Amazon S3. Kemudian, Anda dapat dengan mudah mengurutkan dan memfilter pekerjaan sesuai dengan deskripsi yang Anda tetapkan. Deskripsi tidak harus unik, sehingga Anda dapat menggunakan deskripsi sebagai kategori (misalnya, "Pekerjaan Menyalin Log Mingguan") untuk memudahkan Anda melacak kelompok pekerjaan yang serupa.

Menentukan manifes

Manifes adalah objek Amazon S3 yang berisi kunci objek yang harus ditindaklanjuti Amazon S3. Anda dapat memberikan manifes dengan salah satu cara berikut:

- Buat file manifes baru secara manual.
- Gunakan manifes yang ada.
- Atur Operasi Batch agar menghasilkan manifes secara otomatis berdasarkan kriteria filter objek yang Anda tentukan saat membuat pekerjaan. Opsi ini tersedia untuk pekerjaan replikasi batch yang Anda buat di konsol Amazon S3, atau untuk jenis pekerjaan apa pun yang Anda buat dengan menggunakan AWS CLI AWS SDKs,, atau Amazon S3 REST API.

> **Note**
>
> - Operasi Batch Amazon S3 tidak mendukung pembuatan manifes lintas wilayah.
> - Terlepas dari cara Anda menentukan manifes, daftar itu harus disimpan di dalam sebuah bucket yang bersifat umum. Operasi Batch tidak dapat mengimpor manifes yang ada dari, atau menyimpan manifes yang dihasilkan ke bucket direktori. Objek yang dijelaskan dalam manifes tetap dapat disimpan dalam direktori bucket. Untuk informasi selengkapnya, lihat Bucket direktori.

Membuat file manifes

Untuk membuat file manifes secara manual, Anda menentukan kunci objek manifes, ETag (tag entitas), dan ID versi opsional dalam daftar berformat CSV. Isi manifes harus dienkode URL.

Secara default, Amazon S3 secara otomatis menggunakan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3) untuk mengenkripsi manifes yang diunggah ke bucket Amazon S3. Manifes yang menggunakan enkripsi di sisi server dengan kunci yang disediakan pelanggan (SSE-C) tidak didukung. Manifes yang menggunakan enkripsi sisi server dengan kunci AWS Key Management Service (AWS KMS) (SSE-KMS) hanya didukung saat Anda menggunakan laporan inventaris berformat CSV. Menggunakan manifes yang dibuat secara manual dengan AWS KMS tidak didukung.

Manifes Anda harus berisi nama bucket, kunci objek, dan secara opsional, versi objek untuk setiap objek. Bidang lain dalam manifes tidak digunakan oleh Operasi Batch S3.

> **Note**
>
> Jika objek dalam manifes Anda berada dalam bucket berversi, menentukan versi IDs untuk objek akan mengarahkan Operasi Batch untuk melakukan operasi pada versi tertentu. Jika tidak IDs ada versi yang ditentukan, Operasi Batch melakukan operasi pada versi terbaru objek. Jika manifes menyertakan bidang ID versi, Anda harus memberikan ID versi untuk semua objek dalam manifes.

Berikut ini adalah contoh manifest dalam format CSV tanpa versi IDs.

```
amzn-s3-demo-bucket1,objectkey1
amzn-s3-demo-bucket1,objectkey2
amzn-s3-demo-bucket1,objectkey3
amzn-s3-demo-bucket1,photos/jpgs/objectkey4
amzn-s3-demo-bucket1,photos/jpgs/newjersey/objectkey5
amzn-s3-demo-bucket1,object%20key%20with%20spaces
```

Berikut ini adalah contoh manifes dalam format CSV yang menyertakan versi IDs.

```
amzn-s3-demo-bucket1,objectkey1,PZ9ibn9D5lP6p298B7S9_ceqx1n5EJ0p
amzn-s3-demo-bucket1,objectkey2,YY_ouuAJByNW1LRBfFMfxMge7XQWxMBF
amzn-s3-demo-bucket1,objectkey3,jbo9_jhdPEyB4RrmOxWS0kU0EoNrU_oI
amzn-s3-demo-bucket1,photos/jpgs/objectkey4,6EqlikJJxLTsHsnbZbSRffn24_eh5Ny4
amzn-s3-demo-bucket1,photos/jpgs/newjersey/objectkey5,imHf3FAiRsvBW_EHB8GOu.NHunHO1gVs
amzn-s3-demo-bucket1,object%20key%20with%20spaces,9HkPvDaZY5MVbMhn6TMn1YTb5ArQAo3w
```

Menentukan file manifes yang ada

Anda dapat menentukan manifes untuk permintaan buat pekerjaan dengan menggunakan salah satu dari dua format berikut:

- Laporan Inventaris Amazon S3–Harus berupa laporan Inventaris Amazon S3 berformat CSV. Anda harus menentukan file `manifest.json` yang terkait dengan laporan inventaris. Untuk informasi selengkapnya tentang laporan inventaris, lihat Membuat katalog dan menganalisis data Anda dengan S3 Inventory. Jika laporan inventaris menyertakan versi IDs, Operasi Batch S3 beroperasi pada versi objek tertentu.

  > **Note**
  >
  > - Operasi Batch S3 mendukung laporan inventaris CSV yang dienkripsi dengan SSE-KMS.
  > - Jika Anda mengirimkan manifes laporan inventaris yang dienkripsi dengan SSE-KMS, kebijakan IAM Anda harus menyertakan izin `"kms:Decrypt"` dan `"kms:GenerateDataKey"` untuk objek `manifest.json` serta semua file data CSV terkait.

- File CSV–Setiap baris dalam file harus menyertakan nama bucket, kunci objek, dan secara opsional, versi objek. Kunci objek harus dienkode URL, seperti yang ditunjukkan dalam contoh berikut. Manifes harus menyertakan versi IDs untuk semua objek atau menghilangkan versi IDs untuk semua objek. Untuk informasi selengkapnya tentang format manifes CSV, lihat JobManifestSpec di Referensi API Amazon Simple Storage Service.

  > **Note**
  >
  > Operasi Batch S3 tidak mendukung file manifes CSV yang dienkripsi dengan SSE-KMS.

> **Important**
>
> Saat menggunakan manifes yang dibuat secara manual dan bucket berversi, sebaiknya tentukan versi IDs untuk objek tersebut. Saat Anda membuat pekerjaan, Operasi Batch S3 mengurai seluruh manifes sebelum menjalankan pekerjaan tersebut. Namun, proses tersebut tidak mengambil "snapshot" dari status bucket.
> Karena manifes dapat berisi miliaran objek, pekerjaan mungkin membutuhkan waktu yang lama untuk dijalankan, sehingga dapat memengaruhi versi objek yang ditindaklanjuti oleh

pekerjaan tersebut. Misalnya, Anda menimpa objek dengan versi baru saat pekerjaan berjalan dan Anda tidak menentukan ID versi untuk objek tersebut. Dalam hal ini, Amazon S3 akan melakukan operasi pada versi terbaru dari objek tersebut, bukan pada versi yang ada ketika Anda membuat pekerjaan. Satu-satunya cara untuk menghindari perilaku ini adalah dengan menentukan versi IDs untuk objek yang terdaftar dalam manifes.

Membuat manifes secara otomatis

Anda dapat mengatur Amazon S3 untuk membuat manifes secara otomatis berdasarkan kriteria filter objek yang Anda tentukan saat membuat pekerjaan. Opsi ini tersedia untuk pekerjaan replikasi batch yang Anda buat di konsol Amazon S3, atau untuk jenis pekerjaan apa pun yang Anda buat dengan menggunakan AWS CLI AWS SDKs,, atau Amazon S3 REST API. Untuk informasi selengkapnya tentang Replikasi Batch, lihat Mereplikasi objek yang ada dengan Batch Replication.

Untuk membuat manifes secara otomatis, tentukan elemen berikut sebagai bagian dari permintaan pembuatan pekerjaan Anda:

- Informasi tentang bucket yang berisi objek sumber Anda, termasuk pemilik bucket dan Amazon Resource Name (ARN)
- Informasi tentang output manifes, termasuk tanda untuk membuat file manifes, pemilik bucket output, ARN, prefiks, format file, dan jenis enkripsi
- Kriteria opsional untuk memfilter objek berdasarkan tanggal pembuatan, nama kunci, ukuran, dan kelas penyimpanannya. Dalam kasus pekerjaan replikasi, Anda juga dapat menggunakan tag untuk memfilter objek.

Kriteria filter objek

Untuk memfilter daftar objek yang akan disertakan dalam manifes yang dibuat secara otomatis, Anda dapat menentukan kriteria berikut. Untuk informasi selengkapnya, lihat JobManifestGeneratorFilter di Referensi API Amazon S3.

CreatedAfter

Jika tersedia, manifes yang dibuat hanya akan mencakup objek bucket sumber yang dibuat setelah waktu ini.

CreatedBefore

Jika tersedia, manifes yang dibuat hanya akan mencakup objek bucket sumber yang dibuat sebelum waktu ini.

EligibleForReplication

Jika tersedia, manifes yang dibuat akan mencakup objek hanya jika objek tersebut memenuhi syarat untuk direplikasi sesuai dengan konfigurasi replikasi pada bucket sumber.

KeyNameConstraint

Jika disediakan, manifes yang dihasilkan hanya mencakup objek bucket sumber yang kunci objeknya cocok dengan batasan string yang ditentukan untuk MatchAnySubstring,, MatchAnyPrefixdan. MatchAnySuffix

MatchAnySubstring— Jika disediakan, manifes yang dihasilkan mencakup objek jika string yang ditentukan muncul di mana saja dalam string kunci objek.

MatchAnyPrefix— Jika disediakan, manifes yang dihasilkan mencakup objek jika string yang ditentukan muncul di awal string kunci objek.

MatchAnySuffix— Jika disediakan, manifes yang dihasilkan mencakup objek jika string yang ditentukan muncul di akhir string kunci objek.

MatchAnyStorageClass

Jika tersedia, manifes yang dibuat hanya akan mencakup objek bucket sumber yang disimpan dengan kelas penyimpanan yang ditentukan.

ObjectReplicationStatuses

Jika tersedia, manifes yang dibuat hanya akan mencakup objek bucket sumber yang memiliki salah satu status replikasi yang ditentukan.

ObjectSizeGreaterThanBytes

Jika tersedia, manifes yang dibuat hanya akan mencakup objek bucket sumber yang ukuran filenya lebih besar dari jumlah byte yang ditentukan.

ObjectSizeLessThanBytes

Jika tersedia, manifes yang dibuat hanya akan mencakup objek bucket sumber yang ukuran filenya kurang dari jumlah byte yang ditentukan.

> **Note**
>
> Anda tidak dapat mengkloning sebagian besar tugas yang telah membuat manifes secara otomatis. Pekerjaan replikasi batch dapat dikloning, kecuali ketika pekerjaan tersebut menggunakan kriteria filter manifes `KeyNameConstraint`, `MatchAnyStorageClass`, `ObjectSizeGreaterThanBytes`, atau `ObjectSizeLessThanBytes`.

Sintaks untuk menentukan kriteria manifes bervariasi tergantung pada metode yang Anda gunakan untuk membuat pekerjaan. Sebagai contoh, lihat Membuat pekerjaan.

Membuat pekerjaan

Anda dapat membuat pekerjaan Operasi Batch S3 dengan menggunakan konsol Amazon S3 AWS CLI, AWS SDKs, atau Amazon S3 REST API.

Untuk informasi selengkapnya tentang cara membuat permintaan pekerjaan, lihat Elemen permintaan pekerjaan Operasi Batch.

Prasyarat

Sebelum membuat pekerjaan Operasi Batch, konfirmasikan bahwa Anda telah mengonfigurasi izin yang relevan. Untuk informasi selengkapnya, lihat Memberikan izin untuk Batch Operations.

Menggunakan konsol S3

Untuk membuat pekerjaan batch

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi di bagian atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah tempat Anda ingin membuat pekerjaan Anda.

   > **Note**
   >
   > Untuk operasi penyalinan, Anda harus membuat pekerjaan di Wilayah yang sama dengan bucket tujuan. Untuk semua operasi lainnya, Anda harus membuat pekerjaan di Wilayah yang sama dengan objek dalam manifes.

3. Pilih Operasi Batch di panel navigasi kiri konsol Amazon S3.

4. Pilih Buat tugas.
5. Lihat di Wilayah AWSmana Anda ingin membuat pekerjaan Anda.
6. Pada Format manifes, pilih jenis objek manifes yang akan digunakan.

   - Jika Anda memilih Laporan inventaris S3, masukkan jalur ke objek manifes.json yang dihasilkan Amazon S3 sebagai bagian dari laporan Inventaris berformat CSV, dan secara opsional, ID versi untuk objek manifes jika Anda ingin menggunakan versi selain yang terbaru.
   - Jika Anda memilih CSV, masukkan jalur ke objek manifes yang diformat CSV. Objek manifes harus mengikuti format yang dijelaskan di konsol. Anda dapat memasukkan ID versi secara opsional untuk objek manifes jika ingin menggunakan versi selain yang terbaru.

   > **Note**
   >
   > Konsol Amazon S3 mendukung pembuatan manifes otomatis untuk pekerjaan replikasi batch saja. Untuk semua jenis pekerjaan lainnya, jika Anda ingin Amazon S3 menghasilkan manifes secara otomatis berdasarkan kriteria filter yang Anda tentukan, Anda harus mengonfigurasi pekerjaan menggunakan AWS CLI, AWS SDKs, atau Amazon S3 REST API.

7. Pilih Berikutnya.
8. Pada Operasi, pilih operasi yang ingin Anda jalankan pada semua objek yang tercantum dalam manifes. Isi informasi untuk operasi yang Anda pilih lalu pilih Selanjutnya.
9. Isi informasi untuk Konfigurasikan opsi tambahan, lalu pilih Selanjutnya.
10. Untuk Peninjauan, verifikasikan pengaturan. Jika Anda perlu membuat perubahan, pilih Sebelumnya. Atau, pilih Buat tugas.

Menggunakan AWS CLI

Untuk membuat pekerjaan Operasi Batch Anda dengan AWS CLI, pilih salah satu contoh berikut, tergantung pada apakah Anda menentukan manifes yang ada atau membuat manifes secara otomatis.

Specify manifest

Contoh berikut menunjukkan cara menggunakan AWS CLI untuk membuat `S3PutObjectTagging` pekerjaan Operasi Batch S3 yang bertindak pada objek yang terdaftar dalam file manifes yang ada.

Untuk membuat **`S3PutObjectTagging`** pekerjaan Operasi Batch dengan menentukan manifes

1. Gunakan perintah berikut untuk membuat peran AWS Identity and Access Management (IAM), lalu buat kebijakan IAM untuk menetapkan izin yang relevan. Peran dan kebijakan berikut memberikan izin Amazon S3 untuk menambahkan tanda objek yang akan Anda perlukan saat membuat pekerjaan di langkah berikutnya.

   a. Gunakan perintah contoh berikut untuk membuat peran IAM untuk Operasi Batch yang akan digunakan. Untuk menggunakan contoh perintah ini, ganti *`S3BatchJobRole`* dengan nama yang ingin Anda berikan ke peran tersebut.

```
aws iam create-role \
 --role-name S3BatchJobRole \
 --assume-role-policy-document '{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"batchoperations.s3.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}'
```

   Catat Amazon Resource Name (ARN) untuk peran. Anda akan membutuhkan ARN saat membuat pekerjaan.

   b. Gunakan contoh berikut untuk membuat kebijakan IAM dengan izin yang diperlukan dan lampirkan ke peran IAM yang Anda buat di langkah sebelumnya. Untuk informasi selengkapnya tentang izin yang diperlukan, lihat Memberikan izin untuk Batch Operations.

> **Note**
>
> Pekerjaan Operasi Batch yang menjalankan tindakan pada bucket direktori memerlukan izin tertentu. Untuk informasi selengkapnya, lihat AWS Identity and Access Management (IAM) untuk S3 Express One Zone.

Untuk menggunakan perintah contoh ini, ganti *user input placeholders* sebagai berikut:

- Ganti *S3BatchJobRole* dengan nama peran IAM Anda. Pastikan nama ini cocok dengan nama yang Anda gunakan sebelumnya.
- Ganti *PutObjectTaggingBatchJobPolicy* dengan nama yang ingin Anda berikan pada kebijakan IAM.
- Ganti *amzn-s3-demo-destination-bucket* dengan nama bucket yang berisi objek yang ingin Anda berikan tanda.
- Ganti *amzn-s3-demo-manifest-bucket* dengan nama bucket yang berisi manifes.
- Ganti *amzn-s3-demo-completion-report-bucket* dengan nama bucket tujuan pengiriman laporan penyelesaian.

```
aws iam put-role-policy \
  --role-name S3BatchJobRole \
  --policy-name PutObjectTaggingBatchJobPolicy \
  --policy-document '{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutObjectTagging",
        "s3:PutObjectVersionTagging"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
    },
    {
      "Effect": "Allow",
      "Action": [
```

```
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-manifest-bucket",
        "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
      ]
    },
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutObject",
        "s3:GetBucketLocation"
      ],
      "Resource":[
        "arn:aws:s3:::amzn-s3-demo-completion-report-bucket",
        "arn:aws:s3:::amzn-s3-demo-completion-report-bucket/*"
      ]
    }
  ]
}'
```

2. Gunakan contoh perintah berikut untuk membuat pekerjaan `S3PutObjectTagging`.

   File `manifest.csv` menyediakan daftar bucket dan nilai kunci objek. Pekerjaan ini menerapkan tanda yang ditentukan ke objek yang diidentifikasi dalam manifes. `ETag`Itu adalah ETag `manifest.csv` objek, yang bisa Anda dapatkan dari konsol Amazon S3. Permintaan ini menentukan parameter `no-confirmation-required`, sehingga Anda dapat menjalankan pekerjaan tanpa harus mengonfirmasinya dengan perintah `update-job-status`. Untuk informasi selengkapnya, lihat create-job di AWS CLI Referensi Perintah.

   Untuk menggunakan perintah contoh ini, ganti *`user input placeholders`* dengan informasi Anda sendiri. Ganti *`IAM-role`* dengan ARN peran IAM yang Anda buat sebelumnya.

```
aws s3control create-job \
    --region us-west-2 \
    --account-id acct-id \
    --operation '{"S3PutObjectTagging": { "TagSet": [{"Key":"keyOne",
 "Value":"ValueOne"}] }}' \
```

```
    --manifest '{"Spec":{"Format":"S3BatchOperations_CSV_20180820","Fields":
["Bucket","Key"]},"Location":{"ObjectArn":"arn:aws:s3:::amzn-s3-demo-manifest-
bucket/manifest.csv","ETag":"60e460c9d1046e73f7dde5043ac3ae85"}}' \
    --report '{"Bucket":"arn:aws:s3:::amzn-s3-demo-
completion-report-bucket","Prefix":"final-reports",
 "Format":"Report_CSV_20180820","Enabled":true,"ReportScope":"AllTasks"}' \
    --priority 42 \
    --role-arn IAM-role \
    --client-request-token $(uuidgen) \
    --description "job description" \
    --no-confirmation-required
```

Sebagai tanggapan, Amazon S3 akan mengembalikan ID pekerjaan (misalnya, `00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c`). Anda akan memerlukan ID pekerjaan untuk mengidentifikasi, memantau, dan mengubah pekerjaan.

Generate manifest

Contoh berikut menunjukkan cara membuat pekerjaan `S3DeleteObjectTagging` Operasi Batch S3 yang secara otomatis membuat manifes berdasarkan kriteria filter objek Anda. Kriteria ini mencakup tanggal pembuatan, nama kunci, ukuran, kelas penyimpanan, dan tanda.

Untuk membuat **`S3DeleteObjectTagging`** pekerjaan Operasi Batch dengan membuat manifes

1. Gunakan perintah berikut untuk membuat peran AWS Identity and Access Management (IAM), lalu buat kebijakan IAM untuk menetapkan izin. Peran dan kebijakan berikut memberikan izin Amazon S3 untuk menghapus tanda objek, yang akan Anda perlukan saat membuat pekerjaan di langkah berikutnya.

   a. Gunakan perintah contoh berikut untuk membuat peran IAM untuk Operasi Batch yang akan digunakan. Untuk menggunakan contoh perintah ini, ganti *`S3BatchJobRole`* dengan nama yang ingin Anda berikan ke peran tersebut.

```
aws iam create-role \
 --role-name S3BatchJobRole \
 --assume-role-policy-document '{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
```

```
            "Principal":{
               "Service":"batchoperations.s3.amazonaws.com"
            },
            "Action":"sts:AssumeRole"
         }
   ]
}'
```

Catat Amazon Resource Name (ARN) untuk peran. Anda akan membutuhkan ARN saat membuat pekerjaan.

b. Gunakan contoh berikut untuk membuat kebijakan IAM dengan izin yang diperlukan dan lampirkan ke peran IAM yang Anda buat di langkah sebelumnya. Untuk informasi selengkapnya tentang izin yang diperlukan, lihat Memberikan izin untuk Batch Operations.

> **Note**
>
> Pekerjaan Operasi Batch yang menjalankan tindakan pada bucket direktori memerlukan izin tertentu. Untuk informasi selengkapnya, lihat AWS Identity and Access Management (IAM) untuk S3 Express One Zone.

Untuk menggunakan perintah contoh ini, ganti *user input placeholders* sebagai berikut:

- Ganti *S3BatchJobRole* dengan nama peran IAM Anda. Pastikan nama ini cocok dengan nama yang Anda gunakan sebelumnya.
- Ganti *DeleteObjectTaggingBatchJobPolicy* dengan nama yang ingin Anda berikan pada kebijakan IAM.
- Ganti *amzn-s3-demo-destination-bucket* dengan nama bucket yang berisi objek yang ingin Anda berikan tanda.
- Ganti *amzn-s3-demo-manifest-bucket* dengan nama bucket tempat Anda ingin menyimpan manifes.
- Ganti *amzn-s3-demo-completion-report-bucket* dengan nama bucket tujuan pengiriman laporan penyelesaian.

```
aws iam put-role-policy \
```

```
--role-name S3BatchJobRole \
--policy-name DeleteObjectTaggingBatchJobPolicy \
--policy-document '{
"Version":"2012-10-17",
"Statement":[
  {
    "Effect":"Allow",
    "Action":[
      "s3:DeleteObjectTagging",
      "s3:DeleteObjectVersionTagging"
    ],
    "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
  },
  {
    "Effect":"Allow",
    "Action":[
      "s3:PutInventoryConfiguration"
    ],
    "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket"
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:GetObject",
      "s3:GetObjectVersion",
      "s3:ListBucket"
    ],
    "Resource": [
      "arn:aws:s3:::amzn-s3-demo-manifest-bucket",
      "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
    ]
  },
  {
    "Effect":"Allow",
    "Action":[
      "s3:PutObject",
      "s3:ListBucket"
    ],
    "Resource":[
      "arn:aws:s3:::amzn-s3-demo-completion-report-bucket",
      "arn:aws:s3:::amzn-s3-demo-completion-report-bucket/*",
      "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
    ]
  }
```

```
  ]
}'
```

2. Gunakan contoh berikut untuk membuat pekerjaan `S3DeleteObjectTagging`.

   Dalam contoh ini, nilai di bagian `--report` menentukan bucket, prefiks, format, dan cakupan laporan pekerjaan yang akan dihasilkan. Bagian `--manifest-generator` ini menentukan informasi tentang bucket sumber yang berisi objek yang akan ditindaklanjuti oleh pekerjaan, informasi tentang daftar output manifes yang akan dibuat untuk pekerjaan, dan kriteria filter untuk mempersempit cakupan objek yang akan disertakan dalam manifes berdasarkan tanggal pembuatan, batasan nama, ukuran, dan kelas penyimpanan. Perintah ini juga menentukan prioritas pekerjaan, peran IAM, dan Wilayah AWS.

   Untuk informasi selengkapnya, lihat create-job di AWS CLI Referensi Perintah.

   Untuk menggunakan perintah contoh ini, ganti *`user input placeholders`* dengan informasi Anda sendiri. Ganti *`IAM-role`* dengan ARN peran IAM yang Anda buat sebelumnya.

```
aws s3control create-job \
    --account-id 012345678901 \
    --operation '{
        "S3DeleteObjectTagging": {}
    }' \
    --report '{
        "Bucket":"arn:aws:s3:::amzn-s3-demo-completion-report-bucket",
        "Prefix":"reports",
        "Format":"Report_CSV_20180820",
        "Enabled":true,
        "ReportScope":"AllTasks"
    }' \
    --manifest-generator '{
        "S3JobManifestGenerator": {
          "ExpectedBucketOwner": "012345678901",
          "SourceBucket": "arn:aws:s3:::amzn-s3-demo-source-bucket",
          "EnableManifestOutput": true,
          "ManifestOutputLocation": {
            "ExpectedManifestBucketOwner": "012345678901",
            "Bucket": "arn:aws:s3:::amzn-s3-demo-manifest-bucket",
            "ManifestPrefix": "prefix",
            "ManifestFormat": "S3InventoryReport_CSV_20211130"
```

```
        },
        "Filter": {
          "CreatedAfter": "2023-09-01",
          "CreatedBefore": "2023-10-01",
          "KeyNameConstraint": {
            "MatchAnyPrefix": [
              "prefix"
            ],
            "MatchAnySuffix": [
              "suffix"
            ]
          },
          "ObjectSizeGreaterThanBytes": 100,
          "ObjectSizeLessThanBytes": 200,
          "MatchAnyStorageClass": [
            "STANDARD",
            "STANDARD_IA"
          ]
        }
      }
    }' \
--priority 2 \
--role-arn IAM-role \
--region us-east-1
```

Sebagai tanggapan, Amazon S3 akan mengembalikan ID pekerjaan (misalnya, `00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c`). Anda akan memerlukan ID pekerjaan ini untuk mengidentifikasi, memantau, atau mengubah pekerjaan.

Menggunakan AWS SDK untuk Java

Untuk membuat pekerjaan Operasi Batch Anda dengan AWS SDK untuk Java, pilih salah satu contoh berikut, tergantung pada apakah Anda menentukan manifes yang ada atau membuat manifes secara otomatis.

Specify manifest

Contoh berikut menunjukkan cara membuat pekerjaan `S3PutObjectTagging` Operasi Batch S3 yang bekerja pada objek yang terdaftar dalam file manifes yang ada. Untuk menggunakan contoh ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

Example

```
package aws.example.s3control;



import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.*;

import java.util.UUID;
import java.util.ArrayList;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class CreateJob {
    public static void main(String[] args) {
        String accountId = "Account ID";
        String iamRoleArn = "IAM Role ARN";
        String reportBucketName = "arn:aws:s3:::amzn-s3-demo-completion-report-
bucket";
        String uuid = UUID.randomUUID().toString();

        ArrayList tagSet = new ArrayList<S3Tag>();
        tagSet.add(new S3Tag().withKey("keyOne").withValue("ValueOne"));


        try {
            JobOperation jobOperation = new JobOperation()
                    .withS3PutObjectTagging(new S3SetObjectTaggingOperation()
                            .withTagSet(tagSet)
                    );

            JobManifest manifest = new JobManifest()
                    .withSpec(new JobManifestSpec()
                            .withFormat("S3BatchOperations_CSV_20180820")
                            .withFields(new String[]{
                                    "Bucket", "Key"
                            }))
                    .withLocation(new JobManifestLocation()
                            .withObjectArn("arn:aws:s3:::my_manifests/manifest.csv")
```

```
                            .withETag("60e460c9d1046e73f7dde5043ac3ae85"));
            JobReport jobReport = new JobReport()
                    .withBucket(reportBucketName)
                    .withPrefix("reports")
                    .withFormat("Report_CSV_20180820")
                    .withEnabled(true)
                    .withReportScope("AllTasks");

            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.createJob(new CreateJobRequest()
                    .withAccountId(accountId)
                    .withOperation(jobOperation)
                    .withManifest(manifest)
                    .withReport(jobReport)
                    .withPriority(42)
                    .withRoleArn(iamRoleArn)
                    .withClientRequestToken(uuid)
                    .withDescription("job description")
                    .withConfirmationRequired(false)
            );

        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Generate manifest

Contoh berikut menunjukkan cara membuat pekerjaan `s3PutObjectCopy` Operasi Batch S3 yang secara otomatis membuat manifes berdasarkan kriteria filter objek, termasuk tanggal pembuatan, nama kunci, dan ukuran. Untuk menggunakan contoh ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

Example

```
        package aws.example.s3control;

    import com.amazonaws.AmazonServiceException;
    import com.amazonaws.SdkClientException;
    import com.amazonaws.auth.profile.ProfileCredentialsProvider;
    import com.amazonaws.services.s3control.AWSS3Control;
    import com.amazonaws.services.s3control.AWSS3ControlClient;
    import com.amazonaws.services.s3control.model.CreateJobRequest;
    import com.amazonaws.services.s3control.model.CreateJobResult;
    import com.amazonaws.services.s3control.model.JobManifestGenerator;
    import com.amazonaws.services.s3control.model.JobManifestGeneratorFilter;
    import com.amazonaws.services.s3control.model.JobOperation;
    import com.amazonaws.services.s3control.model.JobReport;
    import com.amazonaws.services.s3control.model.KeyNameConstraint;
    import com.amazonaws.services.s3control.model.S3JobManifestGenerator;
    import com.amazonaws.services.s3control.model.S3ManifestOutputLocation;
    import com.amazonaws.services.s3control.model.S3SetObjectTaggingOperation;
    import com.amazonaws.services.s3control.model.S3Tag;

    import java.time.Instant;
    import java.util.Date;
    import java.util.UUID;
    import java.util.ArrayList;

    import static com.amazonaws.regions.Regions.US_WEST_2;

    public class test {
        public static void main(String[] args) {
            String accountId = "012345678901";
            String iamRoleArn = "arn:aws:iam::012345678901:role/ROLE";
            String sourceBucketName = "arn:aws:s3:::amzn-s3-demo-source-bucket";
            String reportBucketName = "arn:aws:s3:::amzn-s3-demo-completion-report-
bucket";
            String manifestOutputBucketName = "arn:aws:s3:::amzn-s3-demo-manifest-
bucket";
            String uuid = UUID.randomUUID().toString();
            long minimumObjectSize = 100L;

            ArrayList<S3Tag> tagSet = new ArrayList<>();
            tagSet.add(new S3Tag().withKey("keyOne").withValue("ValueOne"));

            ArrayList<String> prefixes = new ArrayList<>();
```

```
            prefixes.add("s3KeyStartsWith");

            try {
                JobOperation jobOperation = new JobOperation()
                        .withS3PutObjectTagging(new S3SetObjectTaggingOperation()
                                .withTagSet(tagSet)
                        );
                S3ManifestOutputLocation manifestOutputLocation = new
S3ManifestOutputLocation()
                        .withBucket(manifestOutputBucketName)
                        .withManifestPrefix("manifests")
                        .withExpectedManifestBucketOwner(accountId)
                        .withManifestFormat("S3InventoryReport_CSV_20211130");

                JobManifestGeneratorFilter jobManifestGeneratorFilter = new
JobManifestGeneratorFilter()
                        .withEligibleForReplication(true)
                        .withKeyNameConstraint(
                                new KeyNameConstraint()
                                        .withMatchAnyPrefix(prefixes))
                        .withCreatedBefore(Date.from(Instant.now()))
                        .withObjectSizeGreaterThanBytes(minimumObjectSize);

                S3JobManifestGenerator s3JobManifestGenerator = new
S3JobManifestGenerator()
                        .withEnableManifestOutput(true)
                        .withManifestOutputLocation(manifestOutputLocation)
                        .withFilter(jobManifestGeneratorFilter)
                        .withSourceBucket(sourceBucketName);

                JobManifestGenerator jobManifestGenerator = new
JobManifestGenerator()
                        .withS3JobManifestGenerator(s3JobManifestGenerator);

                JobReport jobReport = new JobReport()
                        .withBucket(reportBucketName)
                        .withPrefix("reports")
                        .withFormat("Report_CSV_20180820")
                        .withEnabled(true)
                        .withReportScope("AllTasks");

                AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                        .withCredentials(new ProfileCredentialsProvider())
                        .withRegion(US_WEST_2)
```

```
                        .build();

                CreateJobResult createJobResult = s3ControlClient.createJob(new
CreateJobRequest()
                        .withAccountId(accountId)
                        .withOperation(jobOperation)
                        .withManifestGenerator(jobManifestGenerator)
                        .withReport(jobReport)
                        .withPriority(42)
                        .withRoleArn(iamRoleArn)
                        .withClientRequestToken(uuid)
                        .withDescription("job description")
                        .withConfirmationRequired(true)
                );

                System.out.println("Created job " + createJobResult.getJobId());

            } catch (AmazonServiceException e) {
                // The call was transmitted successfully, but Amazon S3 couldn't
process
                // it and returned an error response.
                e.printStackTrace();
            } catch (SdkClientException e) {
                // Amazon S3 couldn't be contacted for a response, or the client
                // couldn't parse the response from Amazon S3.
                e.printStackTrace();
            }
        }
    }
```

Penggunaan API REST

Anda dapat menggunakan API REST untuk membuat pekerjaan Operasi Batch. Untuk informasi selengkapnya, lihat CreateJob di Referensi API Amazon Simple Storage Service.

Respons pekerjaan

Jika permintaan `CreateJob` berhasil, Amazon S3 akan mengembalikan ID pekerjaan. ID pekerjaan adalah pengenal unik yang dihasilkan secara otomatis oleh Amazon S3 sehingga Anda dapat mengidentifikasi pekerjaan Operasi Batch dan memantau statusnya.

Saat Anda membuat pekerjaan melalui AWS CLI, AWS SDKs, atau REST API, Anda dapat mengatur Operasi Batch S3 untuk mulai memproses pekerjaan secara otomatis. Pekerjaan akan segera dijalankan setelah siap, tanpa harus menunggu pekerjaan yang lebih diprioritaskan.

Saat membuat pekerjaan melalui konsol Amazon S3, Anda harus meninjau detail pekerjaan dan mengonfirmasi bahwa Anda ingin menjalankannya sebelum Operasi Batch dapat mulai memprosesnya. Jika suatu pekerjaan tetap berada dalam status ditangguhkan selama lebih dari 30 hari, pekerjaan tersebut akan gagal.

Operasi yang didukung oleh Operasi Batch S3

Anda dapat menggunakan Operasi Batch S3 untuk melakukan operasi batch berskala besar pada objek Amazon S3. Operasi Batch S3 dapat melakukan satu operasi untuk sejumlah objek Amazon S3 yang telah Anda tentukan. Satu pekerjaan dapat melakukan operasi tertentu pada miliaran objek yang berisi data berukuran exabyte. Amazon S3 melacak kemajuan, mengirim pemberitahuan, dan menyimpan laporan penyelesaian terperinci dari semua tindakan, menyediakan pengalaman yang sepenuhnya terkelola, dapat diaudit, dan nirserver. Anda dapat menggunakan Operasi Batch S3 melalui konsol Amazon S3 AWS CLI, AWS SDKs, atau Amazon S3 REST API.

S3 Batch Operations mendukung operasi berikut:

Menyalin objek

Anda dapat menggunakan Operasi Batch Amazon S3 untuk melakukan operasi batch skala besar pada objek Amazon S3. Operasi Batch Operations Copy menyalin setiap objek yang ditentukan dalam manifes. Anda dapat menyalin objek ke ember yang sama Wilayah AWS atau ke ember di Wilayah yang berbeda. Operasi Batch S3 mendukung sebagian besar opsi yang tersedia melalui Amazon S3 untuk menyalin objek. Opsi ini termasuk mengatur metadata objek, mengatur izin, dan mengubah kelas penyimpanan objek.

Anda juga dapat menggunakan operasi Salin untuk menyalin objek tak terenkripsi yang ada dan menuliskannya kembali ke bucket yang sama dengan objek terenkripsi. Untuk informasi selengkapnya, lihat Mengenkripsi objek dengan Operasi Batch Amazon S3.

Saat menyalin objek, Anda dapat mengubah algoritma checksum yang digunakan untuk menghitung checksum objek. Jika objek tidak memiliki checksum tambahan yang dihitung, Anda juga dapat menambahkannya dengan menentukan algoritma checksum yang akan digunakan oleh Amazon S3. Untuk informasi selengkapnya, lihat Memeriksa integritas objek di Amazon S3.

Untuk informasi selengkapnya tentang menyalin objek di Amazon S3 dan parameter yang diperlukan dan opsional, Menyalin, memindahkan, dan mengganti nama objek lihat di panduan ini CopyObjectdan di Referensi API Layanan Penyimpanan Sederhana Amazon.

Pembatasan dan batasan

Saat Anda menggunakan operasi Salinan Operasi Batch, pembatasan dan batasan berikut berlaku:

- Semua objek sumber harus berada dalam satu bucket.
- Semua objek tujuan harus berada dalam satu bucket.
- Anda harus memiliki izin baca untuk bucket sumber dan izin tulis untuk bucket tujuan.
- Objek yang akan disalin dapat berukuran hingga 5 GB.
- Jika Anda mencoba menyalin objek dari kelas S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive ke kelas penyimpanan Standar S3, Anda harus mengembalikan objek ini terlebih dahulu. Untuk informasi selengkapnya, lihat Memulihkan objek yang diarsipkan.
- Anda harus membuat pekerjaan Salin Operasi Batch di Wilayah tujuan, yang merupakan Wilayah tempat Anda ingin menyalin objek.
- Semua `CopyObject` opsi didukung kecuali untuk pemeriksaan bersyarat pada tag entitas (ETags) dan enkripsi sisi server dengan kunci enkripsi yang disediakan pelanggan (SSE-C).
- Jika bucket tujuan tidak berversi, Anda akan menimpa objek apa pun yang memiliki nama kunci yang sama.
- Objek tidak harus disalin dalam urutan yang sama seperti yang muncul di manifes. Untuk bucket berversi, jika mempertahankan urutan versi saat ini atau tidak saat ini penting, salin semua versi noncurrent terlebih dahulu. Kemudian, setelah pekerjaan pertama selesai, salin versi terbaru pada pekerjaan berikutnya.
- Menyalin objek ke kelas Reduced Redundancy Storage (RRS) tidak didukung.
- Satu pekerjaan Salinan Operasi Batch dapat mendukung manifes dengan hingga 20 miliar objek.

Menyalin objek menggunakan Operasi Batch S3

Anda dapat menggunakan Operasi Batch Amazon S3 untuk melakukan operasi batch skala besar pada objek Amazon S3. Anda dapat menggunakan Operasi Batch S3 untuk membuat tugas Copy (`CopyObject`) untuk menyalin objek dalam akun yang sama atau ke akun tujuan yang berbeda.

Contoh berikut menunjukkan cara menyimpan dan menggunakan manifes yang ada di akun yang berbeda. Contoh pertama menunjukkan bagaimana Anda dapat menggunakan Inventaris Amazon

S3 untuk mengirimkan laporan inventaris ke akun tujuan untuk digunakan selama pembuatan pekerjaan. Contoh kedua menunjukkan cara menggunakan manifes nilai dipisahkan koma (CSV) di akun sumber atau tujuan. Contoh ketiga menunjukkan cara menggunakan operasi Copy untuk mengaktifkan S3 Bucket Keys untuk objek yang ada yang telah dienkripsi dengan menggunakan enkripsi sisi server dengan kunci AWS Key Management Service () (AWS KMS SSE-KMS).

Contoh Operasi Salin

- Menggunakan laporan inventaris untuk menyalin objek Akun AWS
- Menggunakan manifes CSV untuk menyalin objek Akun AWS
- Menggunakan Operasi Batch untuk mengaktifkan Kunci Bucket S3 untuk SSE-KMS

Menggunakan laporan inventaris untuk menyalin objek Akun AWS

Anda dapat menggunakan Operasi Batch Amazon S3 untuk melakukan operasi batch skala besar pada objek Amazon S3. Anda dapat menggunakan Operasi Batch S3 untuk membuat tugas Copy (`CopyObject`) untuk menyalin objek dalam akun yang sama atau ke akun tujuan yang berbeda.

Anda dapat menggunakan Inventaris Amazon S3 untuk membuat laporan inventaris dan menggunakan laporan untuk membuat daftar (manifes) objek yang akan disalin dengan Operasi Batch S3. Untuk informasi selengkapnya tentang menggunakan manifes CSV di akun sumber atau tujuan, lihat the section called “Menggunakan manifes CSV untuk menyalin objek Akun AWS”.

Inventaris Amazon S3 menghasilkan inventaris objek dalam bucket. Daftar yang dihasilkan akan diterbitkan ke file output. Bucket yang diinventarisasi disebut bucket sumber, dan bucket tempat menyimpan file laporan inventaris disebut sebagai bucket tujuan.

Laporan Inventaris Amazon S3 dapat dikonfigurasi untuk dikirim ke yang lain. Akun AWS Melakukannya memungkinkan Operasi Batch S3 untuk membaca laporan inventaris saat pekerjaan dibuat di akun tujuan.

Untuk informasi selengkapnya tentang bucket sumber dan tujuan Inventaris Amazon S3, lihat Bucket sumber dan tujuan.

Cara termudah untuk menyiapkan inventaris adalah dengan menggunakan konsol Amazon S3, tetapi Anda juga dapat menggunakan Amazon S3 REST API AWS Command Line Interface ,AWS CLI(), atau. AWS SDKs

Prosedur konsol berikut berisi langkah-langkah tingkat tinggi untuk mengatur izin tugas Operasi Batch S3. Dalam prosedur ini, Anda perlu menyalin objek dari akun sumber ke akun tujuan, dengan laporan inventaris yang disimpan di akun tujuan.

Untuk menyiapkan inventaris Amazon S3 untuk bucket sumber dan tujuan yang dimiliki oleh akun yang berbeda

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Tentukan (atau buat) bucket manifes tujuan untuk menyimpan laporan inventaris. Dalam prosedur ini, yang dimaksud akun tujuan adalah akun yang memiliki bucket manifes tujuan dan bucket yang menerima salinan objek.
4. Konfigurasikan laporan inventaris untuk bucket sumber. Untuk informasi tentang cara menggunakan konsol untuk mengonfigurasi inventaris atau cara mengenkripsi file daftar inventaris, lihatMengonfigurasi Inventaris Amazon S3.

   Saat mengonfigurasi laporan inventaris, Anda menentukan bucket tujuan tempat penyimpanan daftar. Laporan inventaris untuk bucket sumber diterbitkan ke bucket tujuan. Dalam prosedur ini, akun sumber adalah akun yang memiliki bucket sumber.

   Pilih CSV untuk format output.

   Saat Anda memasukkan informasi untuk bucket tujuan, pilih Bucket di akun lain. Kemudian masukkan nama bucket manifes tujuan. Atau, Anda dapat memasukkan ID akun dari akun tujuan.

   Setelah konfigurasi inventaris disimpan, konsol menampilkan pesan yang serupa dengan pesan berikut:

   Amazon S3 tidak dapat membuat kebijakan bucket pada bucket tujuan. Minta pemilik bucket tujuan untuk menambahkan kebijakan bucket berikut agar Amazon S3 menempatkan data dalam bucket tersebut.

   Konsol kemudian akan menampilkan kebijakan bucket yang dapat Anda gunakan untuk bucket tujuan.
5. Salin kebijakan bucket tujuan yang muncul di konsol.

6. Pada akun tujuan, tambahkan kebijakan bucket yang disalin ke bucket manifes tujuan tempat penyimpanan laporan inventaris.
7. Buat peran di akun tujuan yang didasarkan pada kebijakan kepercayaan Operasi Batch S3. Untuk informasi lebih lanjut tentang kebijakan kepercayaan ini, lihatKebijakan kepercayaan.

   Untuk informasi selengkapnya tentang membuat peran, lihat Membuat peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan Pengguna IAM.

   Masukkan nama untuk peran (contoh peran berikut menggunakan nama*BatchOperationsDestinationRoleCOPY*). Pilih layanan S3, lalu pilih kasus penggunaan Operasi Batch S3, yang menerapkan kebijakan kepercayaan pada peran tersebut.

   Kemudian pilih Buat kebijakan untuk melampirkan kebijakan berikut pada peran tersebut. Untuk menggunakan kebijakan ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowBatchOperationsDestinationObjectCOPY",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:PutObjectVersionAcl",
        "s3:PutObjectAcl",
        "s3:PutObjectVersionTagging",
        "s3:PutObjectTagging",
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetObjectAcl",
        "s3:GetObjectTagging",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionTagging"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-destination-bucket/*",
        "arn:aws:s3:::amzn-s3-demo-source-bucket/*",
        "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
      ]
    }
  ]
```

```
}
```

Peran menggunakan kebijakan untuk memberikan izin `batchoperations.s3.amazonaws.com` untuk membaca manifes di bucket tujuan. Ini juga memberikan izin ke `GET` objek, daftar kontrol akses (ACLs), tag, dan versi di bucket objek sumber. Dan itu memberikan izin ke `PUT` objek, tag ACLs, dan versi ke dalam bucket objek tujuan.

8. Di akun sumber, buat kebijakan bucket untuk bucket sumber yang memberikan peran yang Anda buat di izin langkah sebelumnya ke `GET` objek, tag ACLs, dan versi di bucket sumber. Dengan langkah ini, Operasi Batch S3 akan mendapatkan objek dari bucket sumber melalui peran tepercaya.

   Berikut ini adalah contoh kebijakan bucket untuk akun sumber. Untuk menggunakan kebijakan ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowBatchOperationsSourceObjectCOPY",
      "Effect": "Allow",
      "Principal": {
        "AWS":
 "arn:aws:iam::DestinationAccountNumber:role/BatchOperationsDestinationRoleCOPY"
      },
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetObjectAcl",
        "s3:GetObjectTagging",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionTagging"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
    }
  ]
}
```

9. Setelah laporan inventaris tersedia, buat pekerjaan Salin (`CopyObject`) Operasi Batch S3 di akun tujuan, dan pilih laporan inventaris dari bucket manifes tujuan. Anda memerlukan ARN untuk peran IAM yang Anda buat di akun tujuan.

Untuk informasi umum tentang cara membuat tugas, lihat Membuat pekerjaan Operasi Batch S3.

Untuk informasi tentang membuat pekerjaan menggunakan konsol, lihatMembuat pekerjaan Operasi Batch S3.

Menggunakan manifes CSV untuk menyalin objek Akun AWS

Anda dapat menggunakan Operasi Batch Amazon S3 untuk melakukan operasi batch skala besar pada objek Amazon S3. Anda dapat menggunakan Operasi Batch S3 untuk membuat tugas Copy (`CopyObject`) untuk menyalin objek dalam akun yang sama atau ke akun tujuan yang berbeda.

Anda dapat menggunakan manifes CSV yang disimpan di akun sumber untuk menyalin objek Akun AWS dengan Operasi Batch S3. Untuk menggunakan laporan Inventaris S3 sebagai manifes, lihatthe section called “Menggunakan laporan inventaris untuk menyalin objek Akun AWS”.

Untuk contoh format CSV untuk file manifes, lihatthe section called “Membuat file manifes”.

Prosedur berikut menunjukkan cara mengatur izin saat menggunakan pekerjaan Operasi Batch S3 untuk menyalin objek dari akun sumber ke akun tujuan dengan file manifes CSV yang disimpan di akun sumber.

Untuk menggunakan manifes CSV untuk menyalin objek Akun AWS

1. Buat peran AWS Identity and Access Management (IAM) di akun tujuan yang didasarkan pada kebijakan kepercayaan Operasi Batch S3. Pada prosedur ini, akun tujuan adalah akun tempat objek akan disalin.

   Untuk informasi lebih lanjut tentang kebijakan kepercayaan, lihat Kebijakan kepercayaan.

   Untuk informasi selengkapnya tentang membuat peran, lihat Membuat peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan Pengguna IAM.

   Jika Anda membuat peran menggunakan konsol, masukkan nama untuk peran tersebut (contoh peran berikut menggunakan nama*`BatchOperationsDestinationRoleCOPY`*). Pilih layanan S3, lalu pilih kasus penggunaan Operasi Batch S3, yang menerapkan kebijakan kepercayaan pada peran tersebut.

   Kemudian pilih Buat kebijakan untuk melampirkan kebijakan berikut pada peran tersebut. Untuk menggunakan kebijakan ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowBatchOperationsDestinationObjectCOPY",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:PutObjectVersionAcl",
        "s3:PutObjectAcl",
        "s3:PutObjectVersionTagging",
        "s3:PutObjectTagging",
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetObjectAcl",
        "s3:GetObjectTagging",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionTagging"
      ],
      "Resource": [
        "arn:aws: s3:::amzn-s3-demo-destination-bucket/*",
        "arn:aws: s3:::amzn-s3-demo-source-bucket/*",
        "arn:aws: s3:::amzn-s3-demo-manifest-bucket/*"
      ]
    }
  ]
}
```

Dengan menggunakan kebijakan, peran memberikan izin `batchoperations.s3.amazonaws.com` untuk membaca manifes di bucket manifes sumber. Ini memberikan izin ke GET objek, daftar kontrol akses (ACLs), tag, dan versi di bucket objek sumber. Ini juga memberikan izin ke PUT objek, tag ACLs, dan versi ke dalam bucket objek tujuan.

2. Di akun sumber, buat kebijakan bucket untuk bucket yang berisi manifes untuk memberikan peran yang Anda buat di izin langkah sebelumnya ke GET objek dan versi di bucket manifes sumber.

   Langkah ini memungkinkan Operasi Batch S3 untuk membaca manifes dengan menggunakan peran tepercaya. Terapkan kebijakan bucket ke bucket yang berisi manifes.

Berikut ini adalah contoh kebijakan bucket yang dapat diterapkan pada bucket manifes sumber. Untuk menggunakan kebijakan ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowBatchOperationsSourceManfiestRead",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws: iam::DestinationAccountNumber:user/ConsoleUserCreatingJob",
          "arn:aws:
 iam::DestinationAccountNumber:role/BatchOperationsDestinationRoleCOPY"
        ]
      },
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": "arn:aws: s3:::amzn-s3-demo-manifest-bucket/*"
    }
  ]
}
```

Kebijakan ini juga memberikan izin untuk memungkinkan pengguna konsol yang membuat pekerjaan pada akun tujuan menggunakan izin yang sama dalam bucket manifes sumber melalui kebijakan bucket yang sama.

3. Di akun sumber, buat kebijakan bucket untuk bucket sumber yang memberikan peran yang Anda buat izin ke GET objek, tag ACLs, dan versi di bucket objek sumber. Operasi Batch S3 kemudian dapat memperoleh objek dari bucket sumber melalui peran yang tepercaya.

   Berikut ini adalah contoh kebijakan bucket untuk bucket yang berisi objek sumber. Untuk menggunakan kebijakan ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
      "Sid": "AllowBatchOperationsSourceObjectCOPY",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:
 iam::DestinationAccountNumber:role/BatchOperationsDestinationRoleCOPY"
      },
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetObjectAcl",
        "s3:GetObjectTagging",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionTagging"
      ],
      "Resource": "arn:aws: s3:::amzn-s3-demo-source-bucket/*"
    }
  ]
}
```

4. Membuat pekerjaan Operasi Batch S3 di akun tujuan. Anda memerlukan Amazon Resource Name (ARN) untuk peran yang Anda buat di akun tujuan. Untuk informasi selengkapnya tentang cara membuat pekerjaan, lihat Membuat pekerjaan Operasi Batch S3.

Menggunakan Operasi Batch untuk mengaktifkan Kunci Bucket S3 untuk SSE-KMS

S3 Bucket Keys mengurangi biaya enkripsi sisi server dengan AWS Key Management Service (AWS KMS) (SSE-KMS) dengan mengurangi lalu lintas permintaan dari Amazon S3 ke. AWS KMS Untuk informasi selengkapnya, lihat Mengurangi biaya SSE-KMS dengan Kunci Bucket Amazon S3 dan Mengonfigurasi bucket Anda untuk menggunakan Kunci Bucket S3 dengan SSE-KMS untuk objek baru. Saat Anda melakukan `CopyObject` operasi menggunakan REST API,, atau AWS SDKs AWS CLI, Anda dapat mengaktifkan atau menonaktifkan Kunci Bucket S3 di tingkat objek dengan menambahkan header `x-amz-server-side-encryption-bucket-key-enabled` permintaan dengan `false` nilai `true` atau.

Saat Anda mengonfigurasi Kunci Bucket S3 untuk objek menggunakan `CopyObject` operasi, Amazon S3 hanya memperbarui pengaturan untuk objek tersebut. Pengaturan Kunci Bucket S3 untuk bucket tujuan tidak berubah. Jika Anda mengirimkan `CopyObject` permintaan untuk objek AWS KMS terenkripsi ke bucket yang mengaktifkan Kunci Bucket S3, operasi level objek Anda akan secara otomatis menggunakan Kunci Bucket S3 kecuali Anda menonaktifkan kunci di

header permintaan. Jika Anda tidak menentukan Kunci Bucket S3 untuk objek Anda, Amazon S3 menerapkan pengaturan Kunci Bucket S3 untuk bucket tujuan ke objek.

Untuk mengenkripsi objek Amazon S3 yang ada, Anda dapat menggunakan Operasi Batch S3. Anda juga dapat menggunakan Operasi Salin Operasi Batch untuk menyalin objek tidak terenkripsi yang ada, dan menuliskannya kembali ke bucket yang sama sebagai objek terenkripsi. Untuk informasi selengkapnya, lihat Mengenkripsi objek dengan Operasi Batch Amazon S3 di AWS Blog Penyimpanan.

Dalam contoh berikut, Anda menggunakan operasi Salinan Operasi Batch untuk mengaktifkan Kunci Bucket S3 pada objek yang ada. Untuk informasi selengkapnya, lihat the section called "Mengonfigurasi Kunci Bucket S3 untuk sebuah objek".

Topik

- Pertimbangan untuk menggunakan Operasi Batch S3 untuk mengenkripsi objek dengan S3 Bucket Keys diaktifkan
- Prasyarat
- Langkah 1: Dapatkan daftar objek menggunakan Inventaris Amazon S3
- Langkah 2: Filter daftar objek Anda dengan S3 Select
- Langkah 3: Mengatur dan menjalankan pekerjaan Operasi Batch S3

Pertimbangan untuk menggunakan Operasi Batch S3 untuk mengenkripsi objek dengan S3 Bucket Keys diaktifkan

Pertimbangkan masalah berikut saat Anda menggunakan Operasi Batch S3 untuk mengenkripsi objek dengan S3 Bucket Keys diaktifkan:

- Anda akan dikenakan biaya untuk pekerjaan, objek, dan permintaan Operasi Batch S3 selain biaya lain yang terkait dengan operasi yang dilakukan Operasi Batch S3 atas nama Anda, termasuk transfer data, permintaan, dan biaya lainnya. Untuk informasi selengkapnya, lihat Harga Amazon S3.
- Jika Anda menggunakan bucket yang memiliki versi, setiap pekerjaan Operasi Batch S3 yang dilakukan akan membuat versi terenkripsi baru dari objek Anda. Pekerjaan ini juga mempertahankan versi sebelumnya tanpa Kunci Bucket S3 yang dikonfigurasi. Untuk menghapus versi lama, atur kebijakan kedaluwarsa Siklus Hidup S3 untuk versi yang tidak lagi digunakan seperti yang dijelaskan di Elemen konfigurasi Siklus Hidup.

- Operasi salin menciptakan objek baru dengan tanggal pembuatan baru, yang dapat mempengaruhi tindakan siklus hidup seperti pengarsipan. Jika Anda menyalin semua objek dalam bucket, semua salinan baru memiliki tanggal pembuatan yang sama atau serupa. Untuk mengidentifikasi objek ini lebih lanjut dan membuat aturan siklus hidup yang berbeda untuk berbagai subset data, gunakan tanda objek.

Prasyarat

Sebelum Anda mengonfigurasi objek Anda untuk menggunakan Kunci Bucket S3, tinjauPerubahan yang perlu diperhatikan sebelum mengaktifkan Kunci Bucket S3.

Untuk menggunakan contoh ini, Anda harus memiliki Akun AWS dan setidaknya satu bucket S3 untuk menyimpan file kerja dan hasil terenkripsi. Anda mungkin juga akan menemukan banyak dokumentasi Operasi Batch S3 yang berguna, termasuk topik-topik berikut:

- Dasar-dasar Operasi Batch S3
- Membuat pekerjaan Operasi Batch S3
- Operasi yang didukung oleh Operasi Batch S3
- Mengelola tugas Operasi Batch S3

Langkah 1: Dapatkan daftar objek menggunakan Inventaris Amazon S3

Untuk memulai, identifikasi bucket S3 yang berisi objek untuk mengenkripsi, dan dapatkan daftar isinya. Laporan Inventaris Amazon S3 adalah cara yang paling mudah dan terjangkau untuk melakukan hal ini. Laporan ini menyediakan daftar objek dalam ember bersama dengan metadata terkait. Pada langkah ini, bucket sumber adalah bucket yang diinventarisasi, dan bucket tujuan adalah bucket tempat Anda menyimpan file laporan inventaris. Untuk informasi selengkapnya tentang bucket sumber dan tujuan Inventaris Amazon S3, lihat Membuat katalog dan menganalisis data Anda dengan S3 Inventory.

Cara termudah untuk mengatur inventaris adalah dengan menggunakan AWS Management Console. Tetapi Anda juga dapat menggunakan REST API, AWS Command Line Interface (AWS CLI), atau AWS SDKs. Sebelum mengikuti langkah-langkah ini, pastikan untuk masuk ke konsol dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/ Jika Anda mengalami kesalahan izin ditolak, tambahkan kebijakan bucket ke bucket tujuan Anda. Untuk informasi selengkapnya, lihat Berikan izin untuk Inventaris S3 dan analitik S3.

Untuk mendapatkan daftar objek menggunakan Inventaris S3

1. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket, dan pilih bucket yang berisi objek untuk dienkripsi.
3. Pada tab Manajemen, arahkan ke bagian Konfigurasi inventaris, dan pilih Buat konfigurasi inventaris.
4. Beri nama inventaris baru Anda, masukkan nama bucket S3 tujuan, dan buat prefiks tujuan secara opsional untuk Amazon S3 agar dapat menetapkan objek dalam bucket tersebut.
5. Untuk Format output, pilih CSV.
6. (Opsional) Pada bagian Bidang tambahan–opsional, pilih Enkripsi dan bidang laporan lainnya yang menarik untuk Anda. Atur frekuensi pengiriman laporan ke Harian sehingga laporan pertama akan dikirim ke bucket Anda dengan lebih cepat.
7. Pilih Buat untuk menyimpan konfigurasi Anda.

Amazon S3 membutuhkan waktu hingga 48 jam untuk memberikan laporan pertama, jadi periksa kembali saat laporan pertama tiba. Setelah Anda menerima laporan pertama Anda, lanjutkan ke langkah berikutnya untuk memfilter konten laporan Inventaris S3 Anda. Jika tidak ingin menerima laporan inventaris untuk bucket ini lagi, hapus konfigurasi Inventaris S3 Anda. Jika tidak, Amazon S3 terus memberikan laporan pada jadwal harian atau mingguan.

Daftar inventaris bukanlah point-in-time tampilan tunggal dari semua objek. Daftar inventaris adalah snapshot bergulir dari item bucket, yang pada akhirnya konsisten (misalnya, daftar ini mungkin tidak berisi objek yang baru ditambahkan atau dihapus). Penggabungan Inventaris S3 dan Operasi Batch S3 akan bekerja dengan sangat baik ketika Anda bekerja dengan objek statis, atau dengan kumpulan objek yang Anda buat dua hari yang lalu atau lebih. Untuk bekerja dengan data yang lebih baru, gunakan operasi API ListObjectsV2(GETbucket) untuk membuat daftar objek secara manual. Jika perlu, ulangi proses untuk beberapa hari ke depan atau sampai laporan inventaris Anda menunjukkan status yang diinginkan untuk semua objek.

Langkah 2: Filter daftar objek Anda dengan S3 Select

Setelah menerima laporan Inventaris S3, Anda dapat memfilter konten laporan agar hanya mencantumkan objek yang tidak dienkripsi dengan S3 Bucket Keys yang diaktifkan. Jika Anda ingin semua objek bucket dienkripsi dengan S3 Bucket Keys diaktifkan, Anda dapat mengabaikan langkah ini. Namun, memfilter laporan Inventaris S3 Anda pada tahap ini menghemat waktu dan biaya untuk mengenkripsi ulang objek yang sebelumnya Anda enkripsi dengan S3 Bucket Keys diaktifkan.

Meskipun langkah-langkah berikut menunjukkan cara memfilter dengan menggunakan Amazon S3 Select, Anda juga dapat menggunakan Amazon Athena. Untuk memutuskan alat yang akan digunakan, lihat file `manifest.json` laporan Inventaris S3 Anda. File ini mencantumkan jumlah file data yang terkait dengan laporan tersebut. Jika jumlahnya banyak, gunakan Amazon Athena karena dapat digunakan di beberapa objek S3, sedangkan S3 Select hanya bekerja pada satu objek dalam satu waktu. Untuk informasi selengkapnya tentang menggunakan Amazon S3 dan Athena bersama-sama, lihat dan Membuat kueri inventaris Amazon S3 dengan Amazon Athena "Menggunakan Athena" di posting Blog AWS Penyimpanan Mengenkripsi objek dengan Operasi Batch Amazon S3.

Untuk memfilter laporan Inventaris S3 Anda dengan menggunakan S3 Select

1. Buka file `manifest.json` dari laporan inventaris Anda dan lihat bagian `fileSchema` dari JSON. Langkah ini akan memberikan informasi kueri yang Anda jalankan pada data tersebut.

   JSON berikut adalah contoh file `manifest.json` untuk inventaris berformat CSV pada bucket dengan Penentuan Versi yang diaktifkan. Tergantung pada cara Anda mengonfigurasi laporan inventaris, manifes Anda mungkin akan terlihat berbeda.

```
  {
    "sourceBucket": "batchoperationsdemo",
    "destinationBucket": "arn:aws:s3:::amzn-s3-demo-destination-bucket",
    "version": "2021-05-22",
    "creationTimestamp": "1558656000000",
    "fileFormat": "CSV",
    "fileSchema": "Bucket, Key, VersionId, IsLatest, IsDeleteMarker,
 BucketKeyStatus",
    "files": [
      {
        "key": "demoinv/batchoperationsdemo/DemoInventory/data/009a40e4-
f053-4c16-8c75-6100f8892202.csv.gz",
        "size": 72691,
        "MD5checksum": "c24c831717a099f0ebe4a9d1c5d3935c"
      }
    ]
  }
```

   Jika Penentuan Versi tidak diaktifkan pada bucket, atau jika Anda memilih untuk menjalankan laporan untuk versi terbaru, `fileSchema` adalah `Bucket`, `Key`, dan `BucketKeyStatus`.

Jika Penentuan Versi diaktifkan, tergantung pada cara Anda mengatur laporan inventaris, `fileSchema` mungkin berisi kolom berikut: `Bucket`, `Key`, `VersionId`, `IsLatest`, `IsDeleteMarker`, `BucketKeyStatus`. Jadi, perhatikan kolom 1, 2, 3, dan 6 ketika Anda menjalankan kueri.

Operasi Batch S3 membutuhkan bucket, kunci, dan ID versi sebagai input untuk menjalankan pekerjaan, selain bidang yang akan dicari, yaitu `BucketKeyStatus`. Anda tidak memerlukan `VersionID` bidang, tetapi akan membantu menentukan `VersionID` bidang saat Anda beroperasi pada bucket berversi. Untuk informasi selengkapnya, lihat Bekerja dengan objek di dalam bucket dengan dukungan Penentuan Versi.

2. Temukan file data untuk laporan inventaris. Objek `manifest.json` mencantumkan file data di bawah file.
3. Setelah Anda menemukan dan memilih file data di konsol S3, pilih Tindakan, lalu pilih Kueri dengan S3 Select.
4. Simpan preset bidang CSV, Koma, dan GZIP yang dipilih, lalu pilih Selanjutnya.
5. Untuk meninjau format laporan inventaris Anda sebelum melanjutkan, pilih Tampilkan pratinjau file.
6. Masukkan kolom untuk referensi di bidang ekspresi SQL, dan pilih Jalankan SQL. Ekspresi berikut mengembalikan kolom 1–3 untuk semua objek tanpa Kunci Bucket S3 yang dikonfigurasi.

```
select s._1, s._2, s._3 from s3object s where s._6 = 'DISABLED'
```

Berikut ini adalah contoh hasil.

```
      batchoperationsdemo,0100059%7Ethumb.jpg,lsrtIxksLu0R0ZkYPL.LhgD5caTYn6vu
      batchoperationsdemo,0100074%7Ethumb.jpg,sd2M60g6Fdazoi6D5kNARIE7KzUibmHR
      batchoperationsdemo,0100075%7Ethumb.jpg,TLYESLnl1mXD5c4BwiOIinqFrktddkoL
      batchoperationsdemo,0200147%7Ethumb.jpg,amufzfMi_fEw0Rs99rxR_HrDFlE.l3Y0
      batchoperationsdemo,0301420%7Ethumb.jpg,9qGU2SEscL.C.c_sK89trmXYIwooABSh
      batchoperationsdemo,0401524%7Ethumb.jpg,ORnEWNuB1QhHrrYAGFsZhbyvEYJ3DUor
      batchoperationsdemo,200907200065HQ
%7Ethumb.jpg,d8LgvIVjbDR5mUVwW6pu9ahTfReyn5V4
      batchoperationsdemo,200907200076HQ
%7Ethumb.jpg,XUT25d7.gK40u_GmnupdaZg3BVx2jN40
      batchoperationsdemo,201103190002HQ
%7Ethumb.jpg,z.2sVRh0myqVi0BuIrngWlsRPQdb7qOS
```

7. Unduh hasilnya, simpan ke dalam format CSV, dan unggah ke Amazon S3 sebagai daftar objek untuk pekerjaan Operasi Batch S3.
8. Jika Anda memiliki beberapa file manifes, jalankan Kueri dengan S3 Select pada file-file tersebut. Tergantung pada ukuran hasilnya, Anda dapat menggabungkan daftar dan menjalankan satu pekerjaan Operasi Batch S3 atau menjalankan setiap daftar sebagai pekerjaan terpisah. Untuk menentukan jumlah pekerjaan yang akan dijalankan, pertimbangkan harga menjalankan setiap pekerjaan Operasi Batch S3.

Langkah 3: Mengatur dan menjalankan pekerjaan Operasi Batch S3

Sekarang setelah Anda memiliki daftar CSV objek S3 yang difilter, Anda dapat memulai pekerjaan Operasi Batch S3 untuk mengenkripsi objek dengan S3 Bucket Keys diaktifkan.

Pekerjaan mengacu secara kolektif ke daftar (manifes) dari objek yang diberikan, operasi yang dilakukan, dan parameter yang ditentukan. Cara termudah untuk mengenkripsi kumpulan objek ini dengan S3 Bucket Keys diaktifkan adalah dengan menggunakan operasi Copy dan menentukan awalan tujuan yang sama dengan objek yang tercantum dalam manifes. Dalam bucket yang tidak berversi, operasi ini menimpa objek yang ada. Dalam ember dengan versi dihidupkan, operasi ini membuat versi objek yang lebih baru dan terenkripsi.

Sebagai bagian dari menyalin objek, tentukan bahwa Amazon S3 harus mengenkripsi objek dengan enkripsi SSE-KMS. Pekerjaan ini menyalin objek, sehingga semua objek Anda akan menampilkan tanggal pembuatan yang diperbarui setelah selesai, terlepas dari kapan Anda awalnya menambahkannya ke Amazon S3. Tentukan juga properti lain untuk kumpulan objek Anda sebagai bagian dari pekerjaan Operasi Batch S3, termasuk tanda objek dan kelas penyimpanan.

Sublangkah

- Atur kebijakan IAM Anda
- Atur peran IAM Operasi Batch Anda
- Aktifkan Kunci Bucket S3 untuk bucket yang sudah ada
- Membuat pekerjaan Operasi Batch Anda
- Jalankan pekerjaan Operasi Batch Anda

Atur kebijakan IAM Anda

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.

2. Di panel navigasi kiri, pilih Kebijakan, lalu pilih Buat kebijakan.
3. Pilih tab JSON. Pilih Edit kebijakan dan tambahkan contoh kebijakan IAM yang muncul di blok kode berikut.

   Setelah menyalin contoh kebijakan ke Konsol IAM Anda, ganti yang berikut ini:

   a. Ganti *amzn-s3-demo-source-bucket* dengan nama bucket sumber Anda untuk menyalin objek dari.
   b. Ganti *amzn-s3-demo-destination-bucket* dengan nama bucket tujuan Anda untuk menyalin objek.
   c. Ganti *amzn-s3-demo-manifest-bucket*/*manifest-key* dengan nama objek manifes Anda.
   d. Ganti *amzn-s3-demo-completion-report-bucket* dengan nama bucket tempat Anda ingin menyimpan laporan penyelesaian.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CopyObjectsToEncrypt",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:PutObjectTagging",
        "s3:PutObjectAcl",
        "s3:PutObjectVersionTagging",
        "s3:PutObjectVersionAcl",
        "s3:GetObject",
        "s3:GetObjectAcl",
        "s3:GetObjectTagging",
        "s3:GetObjectVersion",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionTagging"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-source-bucket/*",
        "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
      ]
    },
    {
```

```
        "Sid": "ReadManifest",
        "Effect": "Allow",
        "Action": [
          "s3:GetObject",
          "s3:GetObjectVersion"
        ],
        "Resource": "arn:aws:s3:::amzn-s3-demo-manifest-bucket/manifest-key"
      },
      {
        "Sid": "WriteReport",
        "Effect": "Allow",
        "Action": [
          "s3:PutObject"
        ],
        "Resource": "arn:aws:s3:::amzn-s3-demo-completion-report-bucket/*"
      }
    ]
}
```

4. Pilih Selanjutnya: Tanda.
5. Tambahkan tanda yang Anda inginkan (opsional), lalu pilih Berikutnya: Tinjau.
6. Tambahkan nama kebijakan, tambahkan deskripsi secara opsional, lalu pilih Buat kebijakan.
7. Pilih Tinjau kebijakan dan Simpan perubahan.
8. Dengan kebijakan Operasi Batch S3 yang telah selesai, konsol akan mengembalikan Anda ke halaman Kebijakan IAM. Filter nama kebijakan, pilih tombol di sebelah kiri nama kebijakan, pilih Tindakan kebijakan, lalu pilih Lampirkan.

   Untuk melampirkan kebijakan yang baru dibuat ke peran IAM, pilih pengguna, kelompok, atau peran yang sesuai di akun Anda, lalu pilih Lampirkan kebijakan. Tindakan ini akan membawa Anda kembali ke konsol IAM.

Atur peran IAM Operasi Batch Anda

1. Di Konsol IAM, di panel navigasi, pilih Peran, lalu pilih Buat peran.
2. Pilih Layanan AWS, S3, dan Operasi Batch S3. Kemudian, pilih Selanjutnya: Izin.
3. Mulai masukkan nama kebijakan IAM yang baru saja Anda buat. Pilih kotak centang di samping nama kebijakan ketika muncul, dan pilih Selanjutnya: Tanda.
4. (Opsional) Tambahkan tanda atau simpan bidang kunci dan nilai kosong untuk latihan ini. Pilih Selanjutnya: Tinjau.

5. Masukkan nama peran, lalu terima deskripsi default atau tambahkan nama peran Anda sendiri. Pilih Buat peran.
6. Pastikan bahwa pengguna yang membuat pekerjaan memiliki izin dalam contoh berikut.

   Ganti *account-id* dengan ID Akun AWS Anda dan *IAM-role-name* dengan nama yang ingin Anda tetapkan pada peran IAM yang akan Anda buat dalam langkah pembuatan pekerjaan Batch Operations nanti. Untuk informasi selengkapnya, lihat Memberikan izin untuk Batch Operations.

   ```
   {
   "Sid": "AddIamPermissions",
   "Effect": "Allow",
   "Action": [
   "iam:GetRole",
   "iam:PassRole"
   ],
   "Resource": "arn:aws:iam::account-id:role/IAM-role-name"
   }
   ```

Aktifkan Kunci Bucket S3 untuk bucket yang sudah ada

1. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Dalam daftar Bucket, pilih bucket yang ingin Anda aktifkan Kunci Bucket S3-nya.
3. Pilih Properti.
4. Di bagian bawah Enkripsi default, pilih Edit.
5. Pada Jenis enkripsi, Anda dapat memilih antara kunci terkelola Amazon S3 (SSE-S3) dan kunci AWS Key Management Service (SSE-KMS).
6. Jika Anda memilih AWS Key Management Service kunci (SSE-KMS), di bawah AWS KMS key, Anda dapat menentukan AWS KMS kunci melalui salah satu opsi berikut.

   - Untuk memilih dari daftar kunci KMS yang tersedia, pilih Pilih dari kunci AWS KMS Anda. Dari daftar kunci yang tersedia, pilih kunci KMS enkripsi simetris di Wilayah yang sama dengan bucket Anda. Kunci AWS terkelola (`aws/s3`) dan kunci terkelola pelanggan Anda muncul dalam daftar.
   - Untuk memasukkan ARN kunci KMS, pilih Enter AWS KMS key ARN, lalu masukkan ARN kunci KMS Anda di bidang yang muncul.
   - Untuk membuat kunci terkelola pelanggan baru di AWS KMS konsol, pilih Buat kunci KMS.

7. Pada Kunci Bucket, pilih Aktifkan, lalu pilih Simpan perubahan.

Sekarang setelah Kunci Bucket S3 diaktifkan pada tingkat bucket, objek yang diunggah, dimodifikasi, atau disalin ke dalam bucket ini akan mewarisi konfigurasi enkripsi ini secara default. Hal ini mencakup objek yang disalin dengan menggunakan Operasi Batch Amazon S3.

Membuat pekerjaan Operasi Batch Anda

1. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi, pilih Operasi Batch, lalu pilih Buat Pekerjaan.
3. Pilih Wilayah tempat Anda menyimpan objek, lalu pilih CSV sebagai jenis manifes.
4. Masukkan jalur atau arahkan ke file manifes CSV yang telah Anda buat sebelumnya dari hasil S3 Select (atau Athena). Jika manifes Anda berisi versi IDs, pilih kotak itu. Pilih Berikutnya.
5. Pilih operasi Salin, lalu pilih bucket tujuan salinan. Anda dapat membuat enkripsi di sisi server tetap nonaktif. Selama tujuan bucket mengaktifkan Kunci Bucket S3, operasi penyalinan akan menerapkan Kunci Bucket S3 pada bucket tujuan.
6. (Opsional) Pilih kelas penyimpanan dan parameter lainnya sesuai keinginan. Parameter yang Anda tentukan dalam langkah ini berlaku untuk semua operasi yang dilakukan pada objek yang tercantum dalam manifes. Pilih Selanjutnya.
7. Untuk mengonfigurasi enkripsi di sisi server, ikuti langkah-langkah ini:

    a. Pada enkripsi di sisi server, pilih salah satu langkah berikut:

    - Agar pengaturan bucket tetap menggunakan enkripsi di sisi server default untuk objek saat menyimpannya di Amazon S3, pilih Jangan tentukan kunci enkripsi. Selama tujuan bucket telah mengaktifkan Kunci Bucket S3, operasi penyalinan akan menerapkan Kunci Bucket S3 di bucket tujuan.

      > **Note**
      >
      > Jika kebijakan bucket untuk tujuan yang ditentukan mengharuskan objek dienkripsi sebelum disimpan di Amazon S3, Anda harus menentukan kunci enkripsi. Jika tidak, penyalinan objek ke tujuan akan gagal.

    - Untuk mengenkripsi objek sebelum menyimpannya di Amazon S3, pilih Tentukan kunci enkripsi.

b. Pada Pengaturan enkripsi, jika Anda memilih Tentukan kunci enkripsi, Anda harus memilih Gunakan pengaturan bucket tujuan untuk enkripsi default atau Ganti pengaturan bucket tujuan untuk enkripsi default.

c. Jika memilih Ganti pengaturan bucket tujuan untuk enkripsi default, Anda harus mengonfigurasi pengaturan enkripsi berikut.

   i. Pada Jenis enkripsi, Anda harus memilih kunci terkelola Amazon S3 (SSE-S3) atau kunci AWS Key Management Service (SSE-KMS). SSE-S3 menggunakan salah satu penyandian blok terkuat—Advanced Encryption Standard 256-bit (AES-256) untuk mengenkripsi setiap objek. SSE-KMS memberikan kontrol yang lebih besar terhadap kunci Anda. Untuk informasi selengkapnya, lihat Menggunakan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3) dan Menggunakan enkripsi sisi server dengan AWS KMS kunci (SSE-KMS).

   ii. Jika Anda memilih kunci AWS Key Management Service (SSE-KMS), pada AWS KMS key, Anda dapat menentukan AWS KMS key melalui salah satu opsi berikut.

   - Untuk memilih dari daftar kunci KMS yang tersedia, pilih Pilih dari kunci Anda AWS KMS keys, lalu pilih kunci KMS enkripsi simetris di Wilayah yang sama dengan bucket Anda. Kunci AWS terkelola (`aws/s3`) dan kunci terkelola pelanggan Anda muncul dalam daftar.
   - Untuk memasukkan ARN kunci KMS, pilih Enter AWS KMS key ARN, dan masukkan ARN kunci KMS Anda di bidang yang muncul.
   - Untuk membuat kunci terkelola pelanggan baru di AWS KMS konsol, pilih Buat kunci KMS.

   iii. Pada Kunci Bucket, pilih Aktifkan. Operasi penyalinan akan menerapkan Kunci Bucket S3 pada bucket tujuan.

8. Berikan deskripsi pekerjaan Anda (atau tetap menggunakan default), atur tingkat prioritasnya, pilih jenis laporan, dan tentukan Jalur ke tujuan laporan penyelesaian.

9. Pada bagian Izin, pastikan untuk memilih peran IAM Operasi Batch yang telah Anda tentukan sebelumnya. Pilih Berikutnya.

10. Pada Peninjauan, verifikasikan pengaturan. Jika Anda ingin membuat perubahan, pilih Sebelumnya. Setelah mengonfirmasi pengaturan Operasi Batch, pilih Buat pekerjaan.

    Untuk informasi selengkapnya, lihat Membuat pekerjaan Operasi Batch S3.

Jalankan pekerjaan Operasi Batch Anda

Wizard penyiapan secara otomatis mengembalikan Anda ke bagian Operasi Batch S3 dari konsol Amazon S3. Status pekerjaan baru Anda akan berubah dari Baru ke Mempersiapkan saat S3 memulai proses. Selama status Mempersiapkan, S3 membaca manifes pekerjaan, memeriksa adanya kesalahan, dan menghitung jumlah objek.

1. Pilih tombol penyegaran di konsol Amazon S3 untuk memeriksa kemajuan. Tergantung pada ukuran manifes, pembacaan bisa memerlukan waktu beberapa menit atau bahkan jam.
2. Setelah S3 selesai membaca manifes pekerjaan, pekerjaan tersebut akan berubah ke status Menunggu konfirmasi Anda. Pilih tombol opsi di sebelah kiri ID Pekerjaan, dan pilih Jalankan pekerjaan.
3. Periksa pengaturan pekerjaan, lalu pilih Jalankan pekerjaan di pojok kanan bawah.

   Setelah pekerjaan mulai berjalan, Anda dapat memilih tombol penyegaran untuk memeriksa kemajuan melalui tampilan dasbor konsol atau dengan memilih pekerjaan tertentu.
4. Setelah pekerjaan selesai, Anda dapat melihat jumlah objek yang Berhasil dan yang Gagal untuk mengonfirmasi bahwa semua hasilnya sesuai harapan. Jika Anda mengaktifkan laporan pekerjaan, periksa laporan pekerjaan untuk mengetahui penyebab pasti dari operasi yang gagal.

   Anda juga dapat melakukan langkah-langkah ini dengan menggunakan AWS CLI, AWS SDKs, atau Amazon S3 REST API. Untuk informasi selengkapnya tentang melacak status pekerjaan dan laporan penyelesaian, lihat Melacak status tugas dan laporan penyelesaian.

Untuk contoh yang menunjukkan operasi salin dengan tag menggunakan AWS CLI dan AWS SDK untuk Java, lihatMembuat tugas Operasi Batch dengan tanda tugas yang digunakan untuk pelabelan.

Memanggil fungsi AWS Lambda

Anda dapat menggunakan Operasi Batch Amazon S3 untuk melakukan operasi batch skala besar pada objek Amazon S3. Operasi Operasi Batch AWS Lambda fungsi Invoke memulai AWS Lambda fungsi untuk melakukan tindakan kustom pada objek yang terdaftar dalam manifes. Bagian ini menjelaskan cara membuat fungsi Lambda yang akan digunakan dengan Operasi Batch S3 dan cara membuat tugas untuk menginvokasi fungsi tersebut. Tugas Operasi Batch S3 menggunakan operasi `LambdaInvoke` untuk menjalankan fungsi Lambda di setiap objek yang tercantum dalam manifes.

Anda dapat bekerja dengan Operasi Batch S3 dengan menggunakan konsol Amazon S3 AWS Command Line Interface ,AWS CLI() AWS SDKs, atau Amazon S3 REST API. Untuk informasi

selengkapnya tentang menggunakan Lambda, lihat Memulai AWS Lambda di AWS Lambda Panduan Developer.

Bagian berikut menjelaskan cara mulai menggunakan Operasi Batch S3 dengan Lambda.

Topik

- Menggunakan Lambda dengan Operasi Batch
- Membuat fungsi Lambda untuk digunakan dengan Operasi Batch S3
- Membuat tugas Operasi Batch S3 yang menginvokasi fungsi Lambda
- Memberikan informasi tingkat tugas dalam manifes Lambda
- Tutorial Operasi Batch S3

Menggunakan Lambda dengan Operasi Batch

Saat menggunakan Operasi Batch S3 dengan AWS Lambda, Anda harus membuat fungsi Lambda baru khusus untuk digunakan dengan Operasi Batch S3. Anda tidak dapat menggunakan kembali fungsi berbasis peristiwa Amazon S3 yang ada dengan Operasi Batch S3. Fungsi peristiwa hanya dapat menerima pesan; fungsi tersebut tidak dapat mengembalikan pesan. Fungsi Lambda yang digunakan dengan Operasi Batch S3 harus menerima dan mengembalikan pesan. Untuk informasi selengkapnya tentang penggunaan Lambda dengan peristiwa Amazon S3, lihat Menggunakan dengan AWS Lambda Amazon S3 di Panduan Pengembang.AWS Lambda

Buat tugas Operasi Batch S3 yang menginvokasi fungsi Lambda. Tugas ini menjalankan fungsi Lambda yang sama pada semua objek yang tercantum dalam manifes Anda. Anda dapat mengontrol versi fungsi Lambda yang akan digunakan saat memproses objek dalam manifes Anda. Operasi Batch S3 mendukung Nama Sumber Daya Amazon (ARNs), alias, dan versi tertentu yang tidak memenuhi syarat. Untuk informasi selengkapnya, lihat Pengantar AWS Lambda Versi di Panduan AWS Lambda Pengembang.

Jika Anda menyediakan tugas Operasi Batch S3 dengan fungsi ARN yang menggunakan alias atau pengkualifikasi `$LATEST`, dan Anda memperbarui versi yang ditunjuk oleh salah satu dari keduanya, Operasi Batch S3 akan mulai memanggil versi baru fungsi Lambda. Tindakan ini sangat berguna apabila Anda ingin memperbarui fungsionalitas di tengah-tengah pekerjaan yang besar. Jika Anda tidak ingin Operasi Batch S3 mengubah versi yang digunakan, berikan versi tertentu dalam `FunctionARN` parameter saat Anda membuat pekerjaan.

Satu AWS Lambda pekerjaan dengan Operasi Batch S3 dapat mendukung manifes dengan hingga 20 miliar objek.

Menggunakan Operasi Lambda dan Batch dengan bucket direktori

Bucket direktori adalah jenis bucket Amazon S3 yang dirancang untuk beban kerja atau aplikasi kritis kinerja yang memerlukan latensi milidetik satu digit yang konsisten. Untuk informasi selengkapnya, lihat Bucket direktori.

Ada persyaratan khusus untuk menggunakan Operasi Batch untuk menginvokasi fungsi Lambda yang bertindak pada bucket direktori. Misalnya, Anda harus menyusun permintaan Lambda Anda menggunakan skema JSON yang diperbarui, dan menentukan InvocationSchemaVersion2.0 (bukan 1.0) saat Anda membuat pekerjaan. Skema yang diperbarui ini memungkinkan Anda menentukan pasangan nilai kunci opsional untuk UserArguments, yang dapat Anda gunakan untuk memodifikasi parameter tertentu dari fungsi Lambda yang ada. Untuk informasi selengkapnya, lihat Mengotomatiskan pemrosesan objek di bucket direktori Amazon S3 dengan Operasi Batch S3 AWS Lambda dan di Blog Penyimpanan. AWS

Kode respons dan hasil

Operasi Batch S3 memanggil fungsi Lambda dengan satu atau lebih tombol, yang masing-masing memiliki yang `TaskID` terkait dengannya. Operasi Batch S3 mengharapkan kode hasil per kunci dari fungsi Lambda. Setiap tugas yang IDs dikirim dalam permintaan yang tidak dikembalikan dengan kode hasil per-kunci akan diberikan kode hasil dari bidang. `treatMissingKeysAs` `treatMissingKeysAs`adalah bidang permintaan opsional dan default ke. `TemporaryFailure` Tabel berikut berisi kode hasil dan nilai lain yang mungkin untuk `treatMissingKeysAs` bidang tersebut.

Kode respons	Deskripsi
`Succeeded`	Tugas diselesaikan secara normal. Jika Anda meminta laporan penyelesaian tugas, string hasil tugas disertakan dalam laporan.
`TemporaryFailure`	Tugas mengalami kegagalan sementara dan akan diarahkan ulang sebelum pekerjaan selesai. String hasil diabaikan. Jika ini adalah pengarahan ulang terakhir, pesan kesalahan akan disertakan dalam laporan akhir.
`PermanentFailure`	Tugas tersebut mengalami kegagalan permanen. Jika Anda meminta laporan

Kode respons	Deskripsi
	penyelesaian tugas, tugas tersebut akan ditandai sebagai `Failed` dan menyertakan string pesan kesalahan. String hasil dari tugas yang gagal diabaikan.

Membuat fungsi Lambda untuk digunakan dengan Operasi Batch S3

Bagian ini memberikan contoh izin AWS Identity and Access Management (IAM) yang harus Anda gunakan dengan fungsi Lambda Anda. Bagian juga berisi contoh fungsi Lambda untuk digunakan dengan Operasi Batch S3. Jika Anda belum pernah membuat fungsi Lambda sebelumnya, lihat Tutorial: Menggunakan AWS Lambda dengan Amazon S3 di AWS Lambda Panduan Pengembang.

Anda harus membuat fungsi Lambda khusus yang akan digunakan dengan Operasi Batch S3. Anda tidak dapat menggunakan kembali fungsi Lambda berbasis peristiwa Amazon S3 yang ada karena fungsi Lambda yang digunakan untuk Operasi Batch S3 harus menerima dan mengembalikan bidang data khusus.

Important

AWS Lambda fungsi yang ditulis dalam Java menerima salah satu RequestHandleratau antarmuka RequestStreamHandlerhandler. Namun, untuk mendukung permintaan dan format respons Operasi Batch S3, AWS Lambda diperlukan `RequestStreamHandler` antarmuka untuk serialisasi kustom dan deserialisasi permintaan dan respons. Antarmuka ini memungkinkan Lambda untuk meneruskan InputStream dan OutputStream ke metode Java`handleRequest`.

Pastikan untuk menggunakan antarmuka `RequestStreamHandler` saat menggunakan fungsi Lambda dengan Operasi Batch S3. Jika Anda menggunakan antarmuka `RequestHandler`, tugas batch akan gagal dengan pesan “JSON yang tidak valid dikembalikan di payload Lambda” dalam laporan penyelesaian.

Untuk informasi selengkapnya, lihat Antarmuka pengelola di AWS Lambda Panduan Pengguna.

Contoh izin IAM

Berikut ini adalah contoh izin IAM yang diperlukan untuk menggunakan fungsi Lambda dengan Operasi Batch S3.

Example — Kebijakan kepercayaan Operasi Batch S3

Berikut ini adalah contoh kebijakan kepercayaan yang dapat Anda gunakan untuk peran IAM Batch Operations. Peran IAM ini ditentukan saat Anda membuat pekerjaan dan memberikan izin Operasi Batch untuk menjalankan peran IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "batchoperations.s3.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Example — Kebijakan IAM Lambda

Berikut ini adalah contoh kebijakan IAM yang memberikan izin Operasi Batch S3 untuk menginvokasi fungsi Lambda dan membaca manifes input.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "BatchOperationsLambdaPolicy",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion",
                "s3:PutObject",
                "lambda:InvokeFunction"
            ],
            "Resource": "*"
```

```
        }
    ]
}
```

Contoh permintaan dan respons

Bagian ini memberikan contoh permintaan dan respons untuk fungsi Lambda.

Example Permintaan

Berikut ini adalah contoh JSON dari permintaan untuk fungsi Lambda.

```
{
    "invocationSchemaVersion": "1.0",
    "invocationId": "YXNkbGZqYWRmaiBhc2RmdW9hZHNmZGpmaGFzbGtkaGZza2RmaAo",
    "job": {
        "id": "f3cc4f60-61f6-4a2b-8a21-d07600c373ce"
    },
    "tasks": [
        {
            "taskId": "dGFza2lkZ29lc2hlcmUK",
            "s3Key": "customerImage1.jpg",
            "s3VersionId": "1",
            "s3BucketArn": "arn:aws:s3:us-east-1:0123456788:amzn-s3-demo-bucket1"
        }
    ]
}
```

Example Respons

Berikut ini adalah contoh JSON dari respons untuk fungsi Lambda.

```
{
  "invocationSchemaVersion": "1.0",
  "treatMissingKeysAs" : "PermanentFailure",
  "invocationId" : "YXNkbGZqYWRmaiBhc2RmdW9hZHNmZGpmaGFzbGtkaGZza2RmaAo",
  "results": [
    {
      "taskId": "dGFza2lkZ29lc2hlcmUK",
      "resultCode": "Succeeded",
      "resultString": "[\"Mary Major", \"John Stiles\"]"
    }
  ]
```

```
}
```

Contoh fungsi Lambda untuk Operasi Batch S3

Contoh Python Lambda berikut menghilangkan hapus penanda dari objek berversi.

Seperti yang ditunjukkan contoh tersebut, kunci dari Operasi Batch S3 dikodekan oleh URL. Untuk menggunakan Amazon S3 dengan AWS layanan lain, penting bagi Anda untuk mendekode URL kunci yang diteruskan dari Operasi Batch S3.

```
import logging
from urllib import parse
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)
logger.setLevel("INFO")

s3 = boto3.client("s3")


def lambda_handler(event, context):
    """
    Removes a delete marker from the specified versioned object.

    :param event: The S3 batch event that contains the ID of the delete marker
                  to remove.
    :param context: Context about the event.
    :return: A result structure that Amazon S3 uses to interpret the result of the
             operation. When the result code is TemporaryFailure, S3 retries the
             operation.
    """
    # Parse job parameters from Amazon S3 batch operations
    invocation_id = event["invocationId"]
    invocation_schema_version = event["invocationSchemaVersion"]

    results = []
    result_code = None
    result_string = None

    task = event["tasks"][0]
    task_id = task["taskId"]
```

```
    try:
        obj_key = parse.unquote(task["s3Key"], encoding="utf-8")
        obj_version_id = task["s3VersionId"]
        bucket_name = task["s3BucketArn"].split(":")[-1]

        logger.info(
            "Got task: remove delete marker %s from object %s.", obj_version_id,
obj_key
        )

        try:
            # If this call does not raise an error, the object version is not a delete
            # marker and should not be deleted.
            response = s3.head_object(
                Bucket=bucket_name, Key=obj_key, VersionId=obj_version_id
            )
            result_code = "PermanentFailure"
            result_string = (
                f"Object {obj_key}, ID {obj_version_id} is not " f"a delete marker."
            )

            logger.debug(response)
            logger.warning(result_string)
        except ClientError as error:
            delete_marker = error.response["ResponseMetadata"]["HTTPHeaders"].get(
                "x-amz-delete-marker", "false"
            )
            if delete_marker == "true":
                logger.info(
                    "Object %s, version %s is a delete marker.", obj_key,
obj_version_id
                )
                try:
                    s3.delete_object(
                        Bucket=bucket_name, Key=obj_key, VersionId=obj_version_id
                    )
                    result_code = "Succeeded"
                    result_string = (
                        f"Successfully removed delete marker "
                        f"{obj_version_id} from object {obj_key}."
                    )
                    logger.info(result_string)
                except ClientError as error:
```

```
                    # Mark request timeout as a temporary failure so it will be
retried.
                    if error.response["Error"]["Code"] == "RequestTimeout":
                        result_code = "TemporaryFailure"
                        result_string = (
                            f"Attempt to remove delete marker from  "
                            f"object {obj_key} timed out."
                        )
                        logger.info(result_string)
                    else:
                        raise
            else:
                raise ValueError(
                    f"The x-amz-delete-marker header is either not "
                    f"present or is not 'true'."
                )
    except Exception as error:
        # Mark all other exceptions as permanent failures.
        result_code = "PermanentFailure"
        result_string = str(error)
        logger.exception(error)
    finally:
        results.append(
            {
                "taskId": task_id,
                "resultCode": result_code,
                "resultString": result_string,
            }
        )
    return {
        "invocationSchemaVersion": invocation_schema_version,
        "treatMissingKeysAs": "PermanentFailure",
        "invocationId": invocation_id,
        "results": results,
    }
```

Membuat tugas Operasi Batch S3 yang menginvokasi fungsi Lambda

Saat membuat tugas Operasi Batch S3 untuk menginvokasi fungsi Lambda, Anda harus menyediakan hal berikut:

- ARN fungsi Lambda Anda (yang mencakup alias fungsi atau nomor versi tertentu)
- Peran IAM dengan izin untuk menginvokasi fungsi
- Parameter tindakan `LambdaInvokeFunction`

Untuk informasi selengkapnya tentang cara membuat pekerjaan Operasi Batch S3, lihat Membuat pekerjaan Operasi Batch S3 dan Operasi yang didukung oleh Operasi Batch S3.

Contoh berikut membuat pekerjaan Operasi Batch S3 yang memanggil fungsi Lambda dengan menggunakan. AWS CLI Untuk menggunakan contoh ini, ganti *`user input placeholders`*dengan informasi Anda sendiri.

```
aws s3control create-job
    --account-id account-id
    --operation  '{"LambdaInvoke": { "FunctionArn": "arn:aws:lambda:region:account-
id:function:LambdaFunctionName" } }'
    --manifest '{"Spec":{"Format":"S3BatchOperations_CSV_20180820","Fields":
["Bucket","Key"]},"Location":{"ObjectArn":"arn:aws:s3:::amzn-s3-demo-manifest-
bucket","ETag":"ManifestETag"}}'
    --report '{"Bucket":"arn:aws:s3:::amzn-s3-demo-
bucket","Format":"Report_CSV_20180820","Enabled":true,"Prefix":"ReportPrefix","ReportScope":"Al
    --priority 2
    --role-arn arn:aws:iam::account-id:role/BatchOperationsRole
    --region region
    --description "Lambda Function"
```

Memberikan informasi tingkat tugas dalam manifes Lambda

Saat Anda menggunakan AWS Lambda fungsi dengan Operasi Batch S3, Anda mungkin ingin data tambahan menyertai setiap tugas atau kunci yang dioperasikan. Misalnya, Anda mungkin ingin memiliki kunci objek sumber dan kunci objek baru yang disediakan. Fungsi Lambda Anda kemudian dapat menyalin kunci sumber ke bucket S3 baru dengan nama baru. Secara default, Operasi Batch memungkinkan Anda hanya menentukan bucket tujuan dan daftar kunci sumber dalam manifes masukan untuk pekerjaan Anda. Contoh berikut menjelaskan bagaimana Anda dapat menyertakan data tambahan dalam manifes Anda sehingga Anda dapat menjalankan fungsi Lambda yang lebih kompleks.

Untuk menentukan parameter per kunci dalam manifes Operasi Batch S3 yang akan digunakan dalam kode fungsi Lambda, gunakan format JSON yang dienkode dengan URL berikut ini. Bidang `key` diteruskan ke fungsi Lambda Anda seakan-akan itu adalah kunci objek Amazon S3. Tetapi dapat ditafsirkan oleh fungsi Lambda untuk berisi nilai-nilai lain atau beberapa kunci, seperti yang ditunjukkan pada contoh berikut.

> **Note**
>
> Jumlah karakter maksimum untuk bidang `key` pada manifes adalah 1.024.

Example - Manifest menggantikan “kunci Amazon S3” dengan string JSON

Versi yang dienkode URL harus disediakan untuk Operasi Batch S3.

```
amzn-s3-demo-bucket,{"origKey": "object1key", "newKey": "newObject1Key"}
amzn-s3-demo-bucket,{"origKey": "object2key", "newKey": "newObject2Key"}
amzn-s3-demo-bucket,{"origKey": "object3key", "newKey": "newObject3Key"}
```

Example — Manifest URL yang dikodekan

Versi yang dienkode URL ini harus disediakan untuk Operasi Batch S3. non-URL-encodedVersi tidak berfungsi.

```
amzn-s3-demo-bucket,%7B%22origKey%22%3A%20%22object1key%22%2C%20%22newKey%22%3A
%20%22newObject1Key%22%7D
amzn-s3-demo-bucket,%7B%22origKey%22%3A%20%22object2key%22%2C%20%22newKey%22%3A
%20%22newObject2Key%22%7D
amzn-s3-demo-bucket,%7B%22origKey%22%3A%20%22object3key%22%2C%20%22newKey%22%3A
%20%22newObject3Key%22%7D
```

Example — fungsi Lambda dengan hasil penulisan format manifes ke laporan tugas

Contoh manifes yang dikodekan URL ini berisi kunci objek yang dibatasi pipa untuk fungsi Lambda berikut untuk diuraikan.

```
amzn-s3-demo-bucket,object1key%7Clower
amzn-s3-demo-bucket,object2key%7Cupper
amzn-s3-demo-bucket,object3key%7Creverse
```

```
amzn-s3-demo-bucket,object4key%7Cdelete
```

Fungsi Lambda ini menunjukkan cara mengurai tugas yang dibatasi pipa yang dikodekan ke dalam manifes Operasi Batch S3. Tugas tersebut menunjukkan operasi revisi yang diterapkan ke objek tertentu.

```
import logging
from urllib import parse
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)
logger.setLevel("INFO")

s3 = boto3.resource("s3")


def lambda_handler(event, context):
    """
    Applies the specified revision to the specified object.

    :param event: The Amazon S3 batch event that contains the ID of the object to
                  revise and the revision type to apply.
    :param context: Context about the event.
    :return: A result structure that Amazon S3 uses to interpret the result of the
             operation.
    """
    # Parse job parameters from Amazon S3 batch operations
    invocation_id = event["invocationId"]
    invocation_schema_version = event["invocationSchemaVersion"]

    results = []
    result_code = None
    result_string = None

    task = event["tasks"][0]
    task_id = task["taskId"]
    # The revision type is packed with the object key as a pipe-delimited string.
    obj_key, revision = parse.unquote(task["s3Key"], encoding="utf-8").split("|")
    bucket_name = task["s3BucketArn"].split(":")[-1]

    logger.info("Got task: apply revision %s to %s.", revision, obj_key)
```

```
    try:
        stanza_obj = s3.Bucket(bucket_name).Object(obj_key)
        stanza = stanza_obj.get()["Body"].read().decode("utf-8")
        if revision == "lower":
            stanza = stanza.lower()
        elif revision == "upper":
            stanza = stanza.upper()
        elif revision == "reverse":
            stanza = stanza[::-1]
        elif revision == "delete":
            pass
        else:
            raise TypeError(f"Can't handle revision type '{revision}'.")

        if revision == "delete":
            stanza_obj.delete()
            result_string = f"Deleted stanza {stanza_obj.key}."
        else:
            stanza_obj.put(Body=bytes(stanza, "utf-8"))
            result_string = (
                f"Applied revision type '{revision}' to " f"stanza {stanza_obj.key}."
            )

        logger.info(result_string)
        result_code = "Succeeded"
    except ClientError as error:
        if error.response["Error"]["Code"] == "NoSuchKey":
            result_code = "Succeeded"
            result_string = (
                f"Stanza {obj_key} not found, assuming it was deleted "
                f"in an earlier revision."
            )
            logger.info(result_string)
        else:
            result_code = "PermanentFailure"
            result_string = (
                f"Got exception when applying revision type '{revision}' "
                f"to {obj_key}: {error}."
            )
            logger.exception(result_string)
    finally:
        results.append(
            {
                "taskId": task_id,
```

```
                "resultCode": result_code,
                "resultString": result_string,
            }
        )
    return {
        "invocationSchemaVersion": invocation_schema_version,
        "treatMissingKeysAs": "PermanentFailure",
        "invocationId": invocation_id,
        "results": results,
    }
```

Tutorial Operasi Batch S3

Tutorial berikut menyajikan end-to-end prosedur lengkap untuk beberapa tugas Operasi Batch dengan Lambda. Dalam tutorial ini, Anda mempelajari cara mengatur Operasi Batch untuk menjalankan fungsi Lambda untuk transcoding batch video yang disimpan dalam bucket sumber S3. Fungsi Lambda memanggil AWS Elemental MediaConvert untuk mentranskode video.

- Tutorial: Video transcoding batch dengan Operasi Batch S3

Ganti semua tanda objek

Anda dapat menggunakan Operasi Batch Amazon S3 untuk melakukan operasi batch skala besar pada objek Amazon S3. Operasi Ganti semua tag objek menggantikan tag objek pada setiap objek yang tercantum dalam manifes. Tag objek adalah pasangan nilai kunci string yang dapat Anda gunakan untuk menyimpan metadata tentang suatu objek.

Untuk membuat pekerjaan Ganti semua tag objek, Anda menyediakan satu set tag yang ingin Anda terapkan. Operasi Batch S3 menerapkan set tanda yang sama untuk setiap objek. Set tanda yang Anda siapkan akan menggantikan set tanda apa pun yang sudah terkait dengan objek dalam manifes. Operasi Batch S3 tidak mendukung penambahan tag ke objek sambil membiarkan tag yang ada di tempatnya.

Jika objek dalam manifes berada dalam bucket berversi, Anda dapat menerapkan set tanda ke versi tertentu dari setiap objek. Untuk melakukannya, tentukan ID versi untuk setiap objek dalam manifes. Jika Anda tidak menyertakan ID versi untuk objek apa pun, Operasi Batch S3 menerapkan tag yang

disetel ke versi terbaru dari setiap objek. Untuk informasi selengkapnya tentang manifes Operasi Batch, lihat Menentukan manifes.

Untuk informasi selengkapnya tentang penandaan objek, lihat Mengategorikan penyimpanan Anda menggunakan tag di panduan ini, dan lihat PutObjectTaggingGetObjectTagging, dan DeleteObjectTaggingdi Referensi API Amazon Simple Storage Service.

Untuk menggunakan konsol untuk membuat pekerjaan Ganti semua tag objek, lihatMembuat pekerjaan Operasi Batch S3.

Pembatasan dan batasan

Saat Anda menggunakan Operasi Batch untuk mengganti tag objek, pembatasan dan batasan berikut berlaku:

- Peran AWS Identity and Access Management (IAM) yang Anda tentukan untuk menjalankan pekerjaan Operasi Batch harus memiliki izin untuk melakukan operasi yang mendasarinya`PutObjectTagging`. Untuk informasi selengkapnya tentang izin yang diperlukan, lihat PutObjectTaggingdi Referensi API Amazon Simple Storage Service.
- Operasi Batch S3 menggunakan operasi Amazon PutObjectTaggingS3 untuk menerapkan tag ke setiap objek dalam manifes. Semua pembatasan dan batasan yang berlaku untuk operasi yang mendasarinya juga akan berlaku untuk tugas Operasi Batch S3.
- Pekerjaan penggantian semua tag objek tunggal dapat mendukung manifes dengan hingga 20 miliar objek.

Hapus semua tanda objek

Anda dapat menggunakan Operasi Batch Amazon S3 untuk melakukan operasi batch skala besar pada objek Amazon S3. Operasi Hapus semua tanda objek akan menghapus semua kumpulan tanda objek Amazon S3 yang saat ini terkait dengan objek yang terdaftar dalam manifes. Operasi Batch S3 tidak mendukung penghapusan tag dari objek sambil menjaga tag lain tetap di tempatnya.

Jika objek dalam manifes berada dalam bucket yang memiliki versi, Anda dapat menghapus kumpulan tanda dari versi objek tertentu. Untuk melakukannya, Anda harus menentukan ID versi untuk setiap objek dalam manifes. Jika Anda tidak menyertakan ID versi untuk suatu objek, Operasi Batch S3 akan menghapus kumpulan tanda dari versi terbaru setiap objek. Untuk informasi selengkapnya tentang manifes Operasi Batch, lihat Menentukan manifes.

Untuk detail selengkapnya tentang penandaan objek, lihat Mengategorikan penyimpanan Anda menggunakan tag di panduan ini, dan, PutObjectTaggingGetObjectTagging, dan DeleteObjectTaggingdi Referensi API Amazon Simple Storage Service.

> **Warning**
>
> Menjalankan pekerjaan ini akan menghapus semua kumpulan tanda objek pada setiap objek yang tercantum dalam manifes.

Untuk menggunakan konsol untuk membuat pekerjaan Hapus semua tag objek, lihatMembuat pekerjaan Operasi Batch S3.

Pembatasan dan batasan

Saat Anda menggunakan Operasi Batch untuk menghapus tag objek, pembatasan dan batasan berikut berlaku:

- Peran AWS Identity and Access Management (IAM) yang Anda tentukan untuk menjalankan pekerjaan harus memiliki izin untuk melakukan operasi Amazon `DeleteObjectTagging` S3 yang mendasarinya. Untuk informasi selengkapnya, lihat DeleteObjectTagging dalam Referensi API Amazon Simple Storage Service.
- Operasi Batch S3 menggunakan operasi Amazon DeleteObjectTaggingS3 untuk menghapus kumpulan tag dari setiap objek dalam manifes. Semua pembatasan dan batasan yang berlaku untuk operasi yang mendasarinya juga akan berlaku untuk tugas Operasi Batch S3.
- Pekerjaan penandaan objek hapus tunggal dapat mendukung manifes dengan hingga 20 miliar objek.

Ganti daftar kontrol akses (ACL)

Anda dapat menggunakan Operasi Batch Amazon S3 untuk melakukan operasi batch skala besar pada objek Amazon S3. Operasi Ganti daftar kontrol akses (ACL) menggantikan daftar kontrol akses (ACLs) untuk setiap objek yang tercantum dalam manifes. Dengan menggunakan ACLs, Anda dapat menentukan siapa yang dapat mengakses objek dan tindakan apa yang dapat mereka lakukan.

> **Note**
>
> Mayoritas kasus penggunaan modern di Amazon S3 tidak lagi memerlukan penggunaan. ACLs Kami menyarankan agar Anda tetap ACLs dinonaktifkan, kecuali dalam keadaan yang tidak biasa di mana Anda perlu mengontrol akses untuk setiap objek secara individual. Dengan ACLs dinonaktifkan, Anda dapat menggunakan kebijakan untuk mengontrol akses ke semua objek di bucket, terlepas dari siapa yang mengunggah objek ke bucket Anda. Untuk informasi selengkapnya, lihat Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.

Operasi Batch S3 mendukung kustom ACLs yang Anda tentukan dan kalengan ACLs yang disediakan Amazon S3 dengan sekumpulan izin akses yang telah ditentukan sebelumnya.

Jika objek dalam manifes Anda berada dalam bucket berversi, Anda dapat menerapkan ACLs ke versi tertentu dari setiap objek. Untuk melakukannya, tentukan ID versi untuk setiap objek dalam manifes. Jika Anda tidak menyertakan ID versi untuk objek apa pun, Operasi Batch S3 menerapkan ACL ke versi terbaru objek.

Untuk informasi selengkapnya tentang ACLs Amazon S3, lihat. Gambaran umum daftar kontrol akses (ACL)

Blokir Akses Publik S3

Jika Anda ingin membatasi akses publik ke semua objek dalam bucket, sebaiknya gunakan Amazon S3 Block Public Access alih-alih menggunakan Operasi Batch S3 untuk diterapkan. ACLs Blokir Akses Publik dapat membatasi akses publik pada basis per-bucket atau seluruh akun dengan satu operasi sederhana yang dapat diterapkan dengan cepat. Perilaku ini membuat Amazon S3 Block Public Access menjadi pilihan yang lebih baik ketika tujuan Anda adalah mengontrol akses publik ke semua objek dalam bucket atau akun. Gunakan Operasi Batch S3 hanya jika Anda perlu menerapkan ACL yang disesuaikan ke setiap objek dalam manifes. Untuk informasi selengkapnya tentang Blokir Akses Publik S3, lihat Melakukan blok akses publik ke penyimpanan Amazon S3 Anda.

Kepemilikan Objek S3

Jika objek dalam manifes berada dalam bucket yang menggunakan setelan diberlakukan pemilik Bucket untuk Kepemilikan Objek, operasi Ganti daftar kontrol akses (ACL) hanya dapat menentukan objek ACLs yang memberikan kontrol penuh kepada pemilik bucket. Dalam kasus ini, operasi Ganti daftar kontrol akses (ACL) tidak dapat memberikan izin ACL objek ke grup lain Akun AWS atau grup.

Untuk informasi selengkapnya, lihat Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.

Pembatasan dan batasan

Saat Anda menggunakan Operasi Batch untuk mengganti ACLs, pembatasan dan batasan berikut berlaku:

- Peran AWS Identity and Access Management (IAM) yang Anda tentukan untuk menjalankan tugas Ganti daftar kontrol akses (ACL) harus memiliki izin untuk melakukan operasi Amazon S3 yang mendasarinya. `PutObjectAcl` Untuk informasi selengkapnya tentang izin yang diperlukan, lihat PutObjectAcldi Referensi API Amazon Simple Storage Service.
- Operasi Batch S3 menggunakan operasi `PutObjectAcl` Amazon S3 untuk menerapkan ACL yang telah ditentukan ke setiap objek dalam manifes. Oleh karena itu, semua pembatasan dan batasan yang berlaku untuk `PutObjectAcl` operasi yang mendasarinya juga berlaku untuk pekerjaan S3 Batch Operations Replace access control list (ACL).
- Pekerjaan daftar kontrol akses ganti tunggal dapat mendukung manifes dengan hingga 20 miliar objek.

Memulihkan objek dengan Operasi Batch

Anda dapat menggunakan Operasi Batch Amazon S3 untuk melakukan operasi batch skala besar pada objek Amazon S3. Operasi Pemulihan memulai permintaan pemulihan untuk objek Amazon S3 yang diarsipkan yang tercantum dalam manifes Anda. Objek yang diarsipkan berikut ini harus dipulihkan sebelum dapat diakses secara real time:

- Objek yang diarsipkan di kelas penyimpanan S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive
- Objek yang diarsipkan melalui kelas penyimpanan S3 Intelligent-Tiering di tingkat Archive Access atau Deep Archive Access

Menggunakan operasi Restore (S3InitiateRestoreObjectOperation) dalam pekerjaan Operasi Batch S3 Anda menghasilkan `RestoreObject` permintaan untuk setiap objek yang ditentukan dalam manifes.

> **Important**
>
> Pekerjaan Restore hanya memulai permintaan untuk memulihkan objek. Operasi Batch S3 melaporkan pekerjaan sebagai selesai untuk setiap objek setelah permintaan dimulai untuk objek tersebut. Amazon S3 tidak memperbarui pekerjaan atau mengirim notifikasi saat objek telah dipulihkan. Namun, Anda dapat menggunakan Notifikasi Peristiwa S3 untuk menerima notifikasi ketika objek tersebut tersedia di Amazon S3. Untuk informasi selengkapnya, lihat Notifikasi Peristiwa Amazon S3.

Saat Anda membuat pekerjaan Pulihkan, argumen berikut tersedia:

ExpirationInDays

Argumen ini menentukan berapa lama objek S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive tetap tersedia di Amazon S3. Pulihkan pekerjaan yang menargetkan S3 Glacier Flexible Retrieval dan objek S3 Glacier Deep Archive yang mengharuskan Anda atur atau lebih besar. `ExpirationInDays` `1`

> **Important**
>
> Jangan setel `ExpirationInDays` saat membuat pekerjaan operasi Pulihkan yang menargetkan objek tingkat Akses Arsip Tingkat Cerdas S3 dan Akses Arsip Dalam. Objek di tingkatan akses arsip Tingkat Cerdas S3 tidak dikenakan kedaluwarsa pemulihan, jadi menentukan hasil kegagalan permintaan. `ExpirationInDays` `RestoreObject`

GlacierJobTier

Amazon S3 dapat memulihkan objek menggunakan salah satu dari tiga tingkat pengambilan yang berbeda: `EXPEDITED`, `STANDARD`, dan `BULK`. Namun, fitur Operasi Batch S3 hanya mendukung tingkat pengambilan `STANDARD` dan `BULK`. Untuk informasi selengkapnya tentang perbedaan antar tingkat pengambilan, lihat Memahami opsi pengambilan arsip.

Untuk informasi selengkapnya tentang harga untuk setiap tingkat, lihat bagian Permintaan & pengambilan data di halaman harga Amazon S3.

Perbedaan saat memulihkan dari S3 Glacier dan S3 Intelligent-Tiering

Memulihkan file yang diarsipkan dari kelas penyimpanan S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive berbeda dengan memulihkan file dari kelas penyimpanan S3 Intelligent-Tiering di tingkat Archive Access atau Deep Archive Access.

- Saat Anda memulihkan dari S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive, salinan sementara dari objek akan dibuat. Amazon S3 menghapus salinan ini setelah nilai yang Anda tentukan dalam argumen `ExpirationInDays` telah terlewati. Setelah salinan sementara ini dihapus, Anda harus mengirimkan permintaan pemulihan tambahan untuk mengakses objek.
- Saat memulihkan objek S3 Intelligent-Tiering yang diarsipkan, jangan tentukan argumen `ExpirationInDays`. Saat Anda memulihkan objek dari tingkat Archive Access atau Deep Archive Access S3 Intelligent-Tiering, objek dipindah kembali ke tingkat Frequent Access S3 Intelligent-Tiering. Setelah minimal 90 hari berturut-turut tanpa akses, objek secara otomatis dipindah ke tingkat Archive Access. Setelah minimal 180 hari berturut-turut tanpa akses, objek secara otomatis dipindah ke tingkat Deep Archive Access.
- Tugas Operasi Batch dapat beroperasi baik pada objek kelas penyimpanan S3 Glacier Flexible Retrieval dan S3 Glacier Deep Archive atau di objek tingkatan penyimpanan S3 Intelligent-Tiering Archive Access dan Deep Archive Access. Operasi Batch tidak dapat beroperasi pada kedua jenis objek yang diarsipkan dalam tugas yang sama. Untuk memulihkan kedua jenis objek tersebut, Anda harus membuat pekerjaan Operasi Batch terpisah.

Pemulihan yang tumpang tindih

Jika S3InitiateRestoreObjectOperationpekerjaan Anda mencoba memulihkan objek yang sudah dalam proses dipulihkan, Operasi Batch S3 akan dilanjutkan sebagai berikut.

Operasi pemulihan objek berhasil jika salah satu dari kondisi berikut ini benar:

- Dibandingkan dengan permintaan pemulihan yang sedang berlangsung, nilai `ExpirationInDays` pekerjaan ini sama dan nilai `GlacierJobTier` lebih cepat.
- Permintaan pemulihan sebelumnya telah selesai, dan objek saat ini tersedia. Dalam hal ini, Operasi Batch memperbarui tanggal kedaluwarsa objek yang dipulihkan agar sesuai dengan nilai `ExpirationInDays` yang ditentukan dalam permintaan pemulihan yang sedang berlangsung.

Operasi pemulihan objek akan gagal jika salah satu kondisi berikut ini benar:

- Permintaan restorasi yang sedang berlangsung belum selesai, dan durasi restorasi untuk pekerjaan ini (ditentukan oleh `ExpirationInDays` nilai) berbeda dari durasi restorasi yang ditentukan dalam permintaan restorasi yang sedang berlangsung.
- Tingkat restorasi untuk pekerjaan ini (ditentukan oleh `GlacierJobTier` nilainya) sama atau lebih lambat dari tingkat restorasi yang ditentukan dalam permintaan restorasi yang sedang berlangsung.

Batasan

Pekerjaan `S3InitiateRestoreObjectOperation` memiliki batasan berikut ini:

- Anda harus membuat pekerjaan di Wilayah yang sama dengan objek yang diarsipkan.
- Operasi Batch S3 tidak mendukung tingkat `EXPEDITED` pengambilan.
- Satu pekerjaan Batch Operations Restore dapat mendukung manifes dengan hingga 4 miliar objek.

Untuk informasi selengkapnya tentang memulihkan objek, lihat Memulihkan objek yang diarsipkan.

Retensi Kunci Objek S3

Anda dapat menggunakan Operasi Batch Amazon S3 untuk melakukan operasi batch skala besar pada objek Amazon S3. Anda dapat menggunakan operasi retensi Object Lock untuk menerapkan tanggal retensi objek Anda dengan menggunakan mode tata kelola atau mode kepatuhan. Mode retensi ini menerapkan tingkat perlindungan yang berbeda. Anda dapat menerapkan salah satu mode penyimpanan ke versi objek apa pun. Seperti penyimpanan yang sah, tanggal retensi juga mencegah penimpaan atau penghapusan objek. Amazon S3 menyimpan penyimpanan hingga tanggal yang ditentukan dalam metadata objek dan melindungi versi versi objek yang ditentukan hingga periode retensi berakhir.

Anda dapat menggunakan Operasi Batch S3 dengan Object Lock untuk mengelola tanggal penyimpanan banyak objek Amazon S3 sekaligus. Anda menentukan daftar objek target dalam manifes Anda dan mengirimkan manifes ke Operasi Batch untuk diselesaikan. Untuk informasi selengkapnya, lihat Kunci Objek S3 the section called “Periode retensi”.

Tugas Operasi Batch S3 Anda dengan tanggal retensi akan berjalan hingga selesai, dibatalkan, atau hingga status gagal tercapai. Sebaiknya gunakan Operasi Batch S3 dan retensi Kunci Objek S3 saat Anda ingin menambahkan, mengubah, atau menghapus tanggal retensi untuk banyak objek dengan satu permintaan.

Operasi Batch memverifikasi bahwa Kunci Objek telah diaktifkan di bucket Anda sebelum memproses kunci apa pun dalam manifes. Untuk melakukan operasi dan validasi, Operasi Batch memerlukan `s3:GetBucketObjectLockConfiguration` dan `s3:PutObjectRetention` izin dalam peran AWS Identity and Access Management (IAM) untuk memungkinkan Operasi Batch memanggil Kunci Objek atas nama Anda. Untuk informasi selengkapnya, lihat the section called “Pertimbangan Kunci Objek”.

Untuk informasi tentang cara menggunakan operasi ini dengan REST API, lihat `S3PutObjectRetention` pada operasi CreateJob dalam Referensi Amazon Simple Storage Service API.

Untuk contoh AWS Command Line Interface (AWS CLI) menggunakan operasi ini, lihatthe section called “Menggunakan Operasi Batch dengan retensi Kunci Objek”. AWS SDK untuk Java Sebagai contoh, lihatthe section called “Menggunakan Operasi Batch dengan retensi Kunci Objek”.

Pembatasan dan batasan

Saat Anda menggunakan Operasi Batch untuk menerapkan periode retensi Object Lock, pembatasan dan batasan berikut berlaku:

- Operasi Batch S3 tidak membuat perubahan tingkat ember.
- Penentuan Versi dan Kunci Objek S3 harus dikonfigurasi pada bucket tempat tugas dijalankan.
- Semua objek yang tercantum dalam manifes harus berada dalam bucket yang sama.
- Operasi ini bekerja pada versi terbaru objek tersebut kecuali ada versi yang secara eksplisit telah ditentukan dalam manifes.
- Anda memerlukan `s3:PutObjectRetention` izin dalam peran IAM Anda untuk menggunakan pekerjaan retensi Object Lock.
- `s3:GetBucketObjectLockConfiguration`Izin IAM diperlukan untuk mengonfirmasi bahwa Object Lock diaktifkan untuk bucket S3 tempat Anda melakukan pekerjaan.
- Anda hanya dapat memperpanjang periode retensi objek dengan tanggal retensi `COMPLIANCE` mode yang diterapkan, dan periode retensi ini tidak dapat dipersingkat.
- Satu pekerjaan retensi Object Lock S3 dapat mendukung manifes dengan hingga 20 miliar objek.

Pegangan hukum S3 Object Lock

Anda dapat menggunakan Operasi Batch Amazon S3 untuk melakukan operasi batch skala besar pada objek Amazon S3. Anda dapat menggunakan operasi penahanan hukum Object Lock untuk

menempatkan penahanan hukum pada versi objek. Seperti mengatur periode retensi, penyimpanan yang sah mencegah agar versi objek tidak ditimpa atau dihapus. Namun, penahanan hukum tidak memiliki periode retensi terkait dan tetap berlaku sampai dihapus.

Anda dapat menggunakan Operasi Batch S3 dengan Kunci Objek untuk menambahkan penyimpanan yang sah pada banyak objek Amazon S3 sekaligus. Untuk melakukannya, tentukan daftar objek target dalam manifes Anda dan kirimkan daftar itu ke Operasi Batch. Pekerjaan penahanan hukum Kunci Objek Operasi Batch S3 Anda berjalan hingga selesai, hingga pembatalan, atau hingga status kegagalan tercapai.

Operasi Batch S3 memverifikasi bahwa Object Lock diaktifkan pada bucket S3 Anda sebelum memproses objek apa pun dalam manifes. Untuk melakukan operasi objek dan validasi tingkat ember, Operasi Batch S3 membutuhkan `s3:PutObjectLegalHold` `s3:GetBucketObjectLockConfiguration` dan dalam AWS Identity and Access Management peran (IAM). Izin ini memungkinkan Operasi Batch S3 memanggil Kunci Objek S3 atas nama Anda.

Saat Anda membuat pekerjaan Operasi Batch S3 untuk menghapus penahanan hukum, Anda hanya perlu menentukan status `Off` penahanan hukum. Untuk informasi selengkapnya, lihat the section called “Pertimbangan Kunci Objek”.

Untuk informasi tentang cara menggunakan operasi ini dengan Amazon S3 REST API, lihat `S3PutObjectLegalHold` dalam CreateJoboperasi di Referensi API Layanan Penyimpanan Sederhana Amazon.

Untuk contoh menggunakan operasi ini, lihatMenggunakan AWS SDK for Java.

Pembatasan dan batasan

Saat Anda menggunakan Operasi Batch untuk menerapkan atau menghapus penahanan hukum Object Lock, pembatasan dan batasan berikut berlaku:

- Operasi Batch S3 tidak membuat perubahan tingkat ember.
- Semua objek yang tercantum dalam manifes harus berada dalam bucket yang sama.
- Penentuan Versi dan Kunci Objek S3 harus dikonfigurasi pada bucket tempat tugas dijalankan.
- Operasi penahanan hukum Object Lock berfungsi pada versi terbaru objek kecuali versi ditentukan secara eksplisit dalam manifes.
- `s3:PutObjectLegalHold`Izin diperlukan dalam peran IAM Anda untuk menambah atau menghapus penahanan hukum dari objek.

- `s3:GetBucketObjectLockConfiguration`Izin IAM diperlukan untuk mengonfirmasi bahwa Kunci Objek S3 diaktifkan untuk bucket S3 tempat pekerjaan dilakukan.
- Satu pekerjaan penahanan hukum S3 Object Lock dapat mendukung manifes dengan hingga 20 miliar objek.

- Menyalin objek
- Memanggil fungsi AWS Lambda
- Ganti semua tanda objek
- Hapus semua tanda objek
- Ganti daftar kontrol akses (ACL)
- Memulihkan objek dengan Operasi Batch
- Retensi Kunci Objek S3
- Pegangan hukum S3 Object Lock
- Mereplikasi objek yang ada dengan Batch Replication

Mengelola tugas Operasi Batch S3

Amazon S3 menyediakan seperangkat alat yang andal untuk membantu mengelola tugas Operasi Batch S3 setelah dibuat. Bagian ini menjelaskan operasi yang dapat Anda gunakan untuk mengelola dan melacak pekerjaan Anda menggunakan konsol Amazon S3, AWS Command Line Interface (AWS CLI) AWS SDKs, atau Amazon S3 REST API.

Topik

- Menggunakan konsol Amazon S3 untuk mengelola tugas Operasi Batch S3
- Membuat daftar tugas
- Melihat detail tugas
- Menetapkan prioritas tugas

Menggunakan konsol Amazon S3 untuk mengelola tugas Operasi Batch S3

Anda dapat mengelola pekerjaan Operasi Batch S3 dengan menggunakan konsol. Sebagai contoh, Anda dapat:

- Melihat pekerjaan yang aktif dan yang berada dalam antrean

- Mengubah prioritas tugas
- Mengonfirmasi dan menjalankan tugas
- Mengkloning tugas
- Membatalkan tugas

Untuk mengelola Operasi Batch menggunakan konsol

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Operasi Batch.
3. Pilih tugas spesifik yang ingin Anda kelola.

Membuat daftar tugas

Anda dapat mengambil daftar tugas Operasi Batch S3 Anda. Daftar ini mencakup tugas yang belum dan yang telah selesai dalam 90 hari terakhir. Daftar tugas mencakup informasi untuk setiap tugas, seperti ID, deskripsi, prioritas, status saat ini, dan jumlah tugas yang berhasil dan gagal. Anda dapat memfilter daftar tugas menurut statusnya. Saat mengambil daftar tugas melalui konsol, Anda juga dapat mencari tugas menurut deskripsi atau ID dan memfilternya menurut Wilayah AWS.

Dapatkan daftar **`Active`** dan **`Complete`** pekerjaan

AWS CLI Contoh berikut mendapat daftar `Active` dan `Complete` pekerjaan. Untuk menggunakan contoh ini, ganti *`user input placeholders`*dengan informasi Anda sendiri.

```
aws s3control list-jobs \
    --region us-west-2 \
    --account-id account-id \
    --job-statuses '["Active","Complete"]' \
    --max-results 20
```

Untuk informasi dan contoh selengkapnya, lihat list-jobs di Referensi Perintah AWS CLI .

Melihat detail tugas

Jika Anda menginginkan informasi lebih lanjut tentang pekerjaan Operasi Batch Amazon S3 daripada yang dapat Anda ambil dengan mencantumkan pekerjaan, Anda dapat melihat semua detail untuk

satu pekerjaan. Anda dapat melihat detail untuk tugas yang belum atau yang telah selesai dalam 90 hari terakhir. Selain informasi yang ditampilkan dalam daftar tugas, rincian setiap tugas juga mencakup item lain, seperti:

- Parameter operasi
- Detail tentang manifes
- Informasi tentang laporan penyelesaian (jika Anda mengonfigurasinya saat membuat tugas)
- Nama Sumber Daya Amazon (ARN) dari peran pengguna AWS Identity and Access Management (IAM) yang Anda tetapkan untuk menjalankan pekerjaan

Dengan melihat detail tugas individual, Anda dapat mengakses seluruh konfigurasi tugas. Untuk melihat detail pekerjaan, Anda dapat menggunakan konsol Amazon S3 atau AWS Command Line Interface ()AWS CLI.

Mendapatkan deskripsi tugas Operasi Batch S3 di konsol Amazon S3

Untuk melihat deskripsi tugas Operasi Batch menggunakan konsol

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Operasi Batch.
3. Pilih ID tugas dari tugas tertentu untuk melihat detailnya.

Dapatkan deskripsi pekerjaan Operasi Batch S3 di AWS CLI

Contoh berikut mendapatkan deskripsi tugas Operasi Batch S3 menggunakan AWS CLI. Untuk menggunakan perintah contoh berikut, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3control describe-job \
--region us-west-2 \
--account-id account-id \
--job-id 00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c
```

Untuk informasi dan contoh selengkapnya, lihat describe-job di Referensi Perintah AWS CLI .

Menetapkan prioritas tugas

Anda dapat menetapkan setiap pekerjaan Operasi Batch Amazon S3 sebagai prioritas numerik, yang dapat berupa bilangan bulat positif apa pun. Operasi Batch S3 memprioritaskan pekerjaan sesuai dengan prioritas yang ditetapkan. Tugas dengan prioritas yang lebih tinggi (atau nilai numerik yang lebih tinggi untuk parameter prioritas) akan dievaluasi terlebih dahulu. Prioritas ditentukan dalam urutan menurun. Misalnya, antrean tugas dengan nilai prioritas 10 diberikan preferensi penjadwalan di atas antrean tugas dengan nilai prioritas 1.

Anda dapat mengubah prioritas pekerjaan saat pekerjaan sedang berjalan. Jika Anda mengajukan tugas baru dengan prioritas yang lebih tinggi saat tugas sedang berjalan, tugas dengan prioritas yang lebih rendah dapat berhenti agar tugas dengan prioritas yang lebih tinggi dapat berjalan.

Mengubah prioritas pekerjaan tidak mempengaruhi kecepatan pemrosesan pekerjaan.

> **Note**
>
> Operasi Batch S3 menghormati prioritas pekerjaan dengan upaya terbaik. Meskipun pekerjaan dengan prioritas lebih tinggi umumnya lebih diutamakan daripada pekerjaan dengan prioritas lebih rendah, Amazon S3 tidak menjamin pemesanan pekerjaan yang ketat.

Menggunakan konsol S3

Cara memperbarui prioritas pekerjaan di konsol Amazon S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Operasi Batch.
3. Pilih tugas spesifik yang ingin Anda kelola.
4. Pilih Tindakan. Dalam daftar dropdown, pilih Perbarui prioritas.

Menggunakan AWS CLI

Contoh berikut memperbarui prioritas pekerjaan dengan menggunakan AWS CLI. Angka yang lebih tinggi menunjukkan prioritas eksekusi yang lebih tinggi. Untuk menggunakan perintah contoh berikut, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3control update-job-priority \
```

```
    --region us-west-2 \
    --account-id account-id \
    --priority 98 \
    --job-id 00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c
```

Menggunakan AWS SDK untuk Java

Contoh berikut memperbarui prioritas tugas Operasi Batch S3 menggunakan AWS SDK untuk Java.

Untuk informasi selengkapnya tentang prioritas tugas, lihat Menetapkan prioritas tugas.

Example

```
package aws.example.s3control;


import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.UpdateJobPriorityRequest;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class UpdateJobPriority {
    public static void main(String[] args) {
        String accountId = "Account ID";
        String jobId = "00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c";

        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.updateJobPriority(new UpdateJobPriorityRequest()
                    .withAccountId(accountId)
                    .withJobId(jobId)
                    .withPriority(98));

        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
```

```
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Melacak status tugas dan laporan penyelesaian

Dengan Operasi Batch S3, Anda dapat melihat dan memperbarui status tugas, menambahkan notifikasi dan pencatatan, melacak kegagalan tugas, dan menghasilkan laporan penyelesaian.

Topik

- Status tugas
- Memperbarui status tugas
- Notifikasi dan pencatatan
- Melacak kegagalan pekerjaan
- Laporan penyelesaian
- Contoh: Melacak pekerjaan Operasi Batch S3 di Amazon melalui EventBridge AWS CloudTrail
- Contoh: Laporan penyelesaian Operasi Batch S3

Status tugas

Setelah Anda membuat dan menjalankan tugas, tugas tersebut akan berjalan melalui serangkaian status. Tabel berikut ini menjelaskan status dan transisi yang mungkin terjadi di antara keduanya.

Status	Deskripsi	Transisi
`New`	Tugas dimulai dengan status `New` saat Anda membuatnya.	Tugas secara otomatis akan berpindah ke status `Preparing` saat Amazon S3 mulai memproses objek manifes.

Status	Deskripsi	Transisi
`Preparing`	Amazon S3 memproses objek manifes dan parameter tugas lainnya untuk menyiapkan dan menjalankan tugas.	Tuagas secara otomatis akan berpindah ke status `Ready` setelah Amazon S3 selesai memproses manifes dan parameter lainnya. Pekerjaan ini kemudian siap untuk mulai menjalankan operasi yang ditentukan pada objek yang tercantum dalam manifes. Jika tugas memerlukan konfirmasi sebelum dijalankan, seperti saat Anda membuat tugas menggunakan konsol Amazon S3, maka tugas tersebut akan bertransisi dari `Preparing` ke `Suspended`. Tugas akan tetap di status `Suspended` sampai Anda mengonfirmasi untuk menjalankannya.

Status	Deskripsi	Transisi
`Suspended`	Pekerjaan itu membutuhkan konfirmasi, tetapi Anda belum memastikan bahwa Anda ingin menjalankannya. Hanya tugas yang Anda buat menggunakan konsol Amazon S3 yang memerlukan konfirmasi. Pekerjaan yang dibuat menggunakan konsol memasuki `Suspended` status segera setelah nya `Preparing`. Setelah Anda mengonfirmasi untuk menjalankan tugas dan tugas tersebut telah menjadi `Ready`, tugas itu tidak pernah kembali ke status `Suspended`.	Setelah Anda mengonfirmasi untuk menjalankan tugas, statusnya berubah menjadi `Ready`.
`Ready`	Amazon S3 siap untuk mulai menjalankan pengoperasian objek yang diminta.	Tugas secara otomatis berpindah ke status `Active` saat Amazon S3 mulai menjalankannya. Lama waktu yang diperlukan tugas untuk tetap berada dalam status `Ready` tergantung pada apakah Anda sudah memiliki tugas dengan prioritas yang lebih tinggi dan berapa lama tugas tersebut harus diselesaikan.

Status	Deskripsi	Transisi
`Active`	Amazon S3 menjalankan operasi yang diminta pada objek yang tercantum dalam manifes. Sementara pekerjaan adalah`Active`, Anda dapat memantau kemajuannya menggunakan konsol Amazon S3 atau `DescribeJob` operasi melalui REST API, AWS CLI, atau. AWS SDKs	Sebuah pekerjaan bergerak keluar dari `Active` keadaan ketika pekerjaan tidak lagi menjalankan operasi pada objek. Perilaku ini dapat terjadi secara otomatis, seperti ketika pekerjaan selesai dengan sukses atau gagal. Atau perilaku ini dapat terjadi sebagai akibat dari tindakan pengguna, seperti membatalkan pekerjaan. Status perpindahan tugas tergantung pada alasan transisi.
`Pausing`	Tugas bertransisi ke `Paused` dari status lain.	Tugas secara otomatis berpindah ke `Paused` saat tahap `Pausing` selesai.
`Paused`	Tugas dapat menjadi `Paused` jika Anda mengirimkan tugas lain dengan prioritas yang lebih tinggi ketika tugas saat ini sedang berjalan.	Tugas `Paused` secara otomatis kembali ke `Active` setelah tugas dengan prioritas yang lebih tinggi yang menghalangi eksekusi tugas telah selesai, gagal, atau ditangguhkan.

Status	Deskripsi	Transisi
`Complete`	Tugas telah selesai melakukan operasi yang diminta pada semua objek dalam manifes. Operasi mungkin telah berhasil atau gagal untuk setiap objek. Jika Anda mengonfigurasi tugas untuk membuat laporan penyelesaian, laporan tersebut akan tersedia segera setelah tugas `Complete`.	`Complete` adalah status terminal. Setelah pekerjaan tercapai`Complete`, itu tidak bertransisi ke keadaan lain.
`Cancelling`	Tugas bertransisi ke status `Cancelled` .	Tugas secara otomatis berpindah ke `Cancelled` saat tahap `Cancelling` selesai.
`Cancelled`	Anda meminta agar pekerjaan dibatalkan, dan Operasi Batch S3 telah berhasil membatalk an pekerjaan. Pekerjaan tidak akan mengirimkan permintaan baru ke Amazon S3.	`Cancelled` adalah status terminal. Setelah pekerjaan tercapai`Cancelled` , pekerjaan tidak akan beralih ke keadaan lain.
`Failing`	tugas beralih ke status `Failed`.	Tugas secara otomatis berpindah ke `Failed` setelah tahap `Failing` selesai.
`Failed`	Tugas gagal dan tidak lagi berjalan. Untuk informasi selengkapnya tentang kegagalan tugas, lihat Melacak kegagalan pekerjaan.	`Failed` adalah status terminal. Setelah pekerjaan tercapai`Failed`, itu tidak akan beralih ke keadaan lain.

Memperbarui status tugas

Berikut AWS CLI dan AWS SDK untuk Java contoh memperbarui status pekerjaan Operasi Batch. Untuk informasi selengkapnya tentang menggunakan konsol Amazon S3 untuk mengelola pekerjaan Operasi Batch, lihat. Menggunakan konsol Amazon S3 untuk mengelola tugas Operasi Batch S3

Menggunakan AWS CLI

Untuk menggunakan perintah contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri.

- Jika Anda tidak menentukan `--no-confirmation-required` parameter dalam `create-job` perintah Anda, pekerjaan tetap dalam status ditangguhkan sampai Anda mengonfirmasi pekerjaan dengan menyetel statusnya`Ready`. Amazon S3 kemudian membuat tugas yang memenuhi syarat untuk dijalankan.

```
aws s3control update-job-status \
    --region us-west-2 \
    --account-id 123456789012 \
    --job-id 00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c \
    --requested-job-status 'Ready'
```

- Batalkan tugas dengan mengatur status tugas menjadi `Cancelled`.

```
aws s3control update-job-status \
     --region us-west-2 \
     --account-id 123456789012 \
     --job-id 00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c \
     --status-update-reason "No longer needed" \
     --requested-job-status Cancelled
```

Menggunakan AWS SDK for Java

Contoh berikut memperbarui status pekerjaan Operasi Batch S3 dengan menggunakan. AWS SDK untuk Java

Untuk informasi selengkapnya tentang status tugas, lihat Melacak status tugas dan laporan penyelesaian.

Example

```
package aws.example.s3control;


import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.UpdateJobStatusRequest;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class UpdateJobStatus {
    public static void main(String[] args) {
        String accountId = "Account ID";
        String jobId = "00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c";

        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.updateJobStatus(new UpdateJobStatusRequest()
                    .withAccountId(accountId)
                    .withJobId(jobId)
                    .withRequestedJobStatus("Ready"));

        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Notifikasi dan pencatatan

Selain meminta laporan penyelesaian, Anda juga dapat menangkap, meninjau, dan mengaudit aktivitas Operasi Batch dengan menggunakan AWS CloudTrail. Karena Operasi Batch menggunakan operasi API Amazon S3 yang ada untuk melakukan tugas, tugas tersebut juga memancarkan peristiwa yang sama jika Anda memanggilnya secara langsung. Oleh karena itu, Anda dapat melacak dan merekam kemajuan pekerjaan Anda dan semua tugasnya dengan menggunakan alat dan proses notifikasi, pencatatan, dan audit yang sama yang sudah Anda gunakan dengan Amazon S3. Untuk informasi selengkapnya, lihat contoh di bagian berikut.

> **Note**
>
> Operasi Batch menghasilkan peristiwa manajemen dan data CloudTrail selama pelaksanaan pekerjaan. Volume skala peristiwa ini dengan jumlah kunci dalam manifes setiap pekerjaan. Untuk informasi selengkapnya, lihat halaman CloudTrail harga, yang mencakup contoh perubahan harga tergantung pada jumlah jejak yang telah Anda konfigurasikan di akun Anda. Untuk mempelajari cara mengonfigurasi dan mencatat peristiwa agar sesuai dengan kebutuhan, lihat Buat jejak pertama Anda di Panduan Pengguna AWS CloudTrail .

Untuk informasi selengkapnya tentang peristiwa Amazon S3, lihat Notifikasi Peristiwa Amazon S3.

Melacak kegagalan pekerjaan

Jika tugas Operasi Batch S3 mengalami masalah yang membuatnya mengalami kegagalan, seperti tidak dapat membaca manifes yang ditentukan, maka tugas tersebut akan gagal. Saat mengalami kegagalan, tugas akan menghasilkan satu atau beberapa kode kegagalan atau alasan kegagalan. Operasi Batch S3 menyimpan kode kegagalan dan alasan dengan pekerjaan sehingga Anda dapat melihatnya dengan meminta detail pekerjaan. Jika Anda meminta laporan penyelesaian untuk tugas tersebut, kode dan alasan kegagalan juga akan muncul di sana.

Agar tugas tidak menjalankan banyak operasi yang gagal, Amazon S3 memberlakukan ambang batas kegagalan tugas pada setiap tugas Operasi Batch. Ketika sebuah pekerjaan telah menjalankan setidaknya 1.000 tugas, Amazon S3 memantau tingkat kegagalan tugas. Jika tingkat kegagalan (jumlah tugas yang gagal sebagai proporsi dari jumlah total tugas yang telah berjalan) melebihi 50 persen, maka pekerjaan tersebut gagal. Jika pekerjaan gagal karena melebihi ambang batas kegagalan tugas, Anda dapat mengidentifikasi penyebab kegagalan. Misalnya, Anda mungkin tidak sengaja memasukkan beberapa objek dalam manifes yang tidak ada dalam bucket tertentu. Setelah memperbaiki kesalahan, Anda dapat mengirim ulang tugas tersebut.

> **Note**
>
> Operasi Batch S3 beroperasi secara asinkron dan tugas tidak selalu berjalan dalam urutan bahwa objek terdaftar dalam manifes. Oleh karena itu, Anda tidak dapat menggunakan urutan manifes untuk menentukan tugas objek yang berhasil maupun yang gagal. Sebagai gantinya, Anda dapat memeriksa laporan penyelesaian pekerjaan (jika Anda memintanya) atau melihat log AWS CloudTrail peristiwa Anda untuk membantu menentukan sumber kegagalan.

Laporan penyelesaian

Saat Anda membuat tugas, Anda dapat meminta laporan penyelesaian. Selama Operasi Batch S3 berhasil memanggil setidaknya satu tugas, Amazon S3 menghasilkan laporan penyelesaian setelah pekerjaan selesai menjalankan tugas, gagal, atau dibatalkan. Anda dapat mengonfigurasi laporan penyelesaian untuk menyertakan semua tugas atau hanya tugas yang gagal.

Laporan penyelesaian mencakup konfigurasi pekerjaan, status, dan informasi untuk setiap tugas, termasuk kunci objek dan versi, status, kode kesalahan, dan deskripsi kesalahan apa pun. Laporan penyelesaian memberikan cara mudah untuk melihat hasil tugas Anda dalam format terkonsolidasi tanpa perlu pengaturan tambahan. Laporan penyelesaian dienkripsi dengan menggunakan enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3). Untuk contoh laporan penyelesaian, lihat Contoh: Laporan penyelesaian Operasi Batch S3.

Jika Anda tidak mengonfigurasi laporan penyelesaian, Anda masih dapat memantau dan mengaudit pekerjaan Anda dan tugasnya dengan menggunakan CloudTrail dan Amazon CloudWatch. Untuk informasi selengkapnya, lihat topik berikut:

Topik

- Contoh: Melacak pekerjaan Operasi Batch S3 di Amazon melalui EventBridge AWS CloudTrail
- Contoh: Laporan penyelesaian Operasi Batch S3

Contoh: Melacak pekerjaan Operasi Batch S3 di Amazon melalui EventBridge AWS CloudTrail

Aktivitas pekerjaan Operasi Batch Amazon S3 dicatat sebagai peristiwa di AWS CloudTrail. Anda dapat membuat aturan khusus di Amazon EventBridge dan mengirim peristiwa ini ke sumber notifikasi target pilihan Anda, seperti Amazon Simple Notification Service (Amazon SNS).

> **Note**
>
> Amazon EventBridge adalah cara yang lebih disukai untuk mengelola acara Anda. Amazon CloudWatch Events dan EventBridge merupakan layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di salah satu CloudWatch atau EventBridge muncul di setiap konsol. Untuk informasi selengkapnya, lihat Panduan EventBridge Pengguna Amazon.

Contoh Pelacakan

- Peristiwa Operasi Batch S3 direkam di CloudTrail
- EventBridge aturan untuk melacak peristiwa pekerjaan Operasi Batch S3

Peristiwa Operasi Batch S3 direkam di CloudTrail

Ketika pekerjaan Operasi Batch dibuat, itu dicatat sebagai `JobCreated` peristiwa di CloudTrail. Saat pekerjaan berjalan, itu berubah status selama pemrosesan, dan `JobStatusChanged` peristiwa lainnya dicatat CloudTrail. Anda dapat melihat acara ini di CloudTrail konsol. Untuk informasi selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna.

> **Note**
>
> Hanya `status-change` peristiwa pekerjaan Operasi Batch S3 yang dicatat di CloudTrail.

Example — Peristiwa penyelesaian pekerjaan Operasi Batch S3 dicatat oleh CloudTrail

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "accountId": "123456789012",
        "invokedBy": "s3.amazonaws.com"
    },
    "eventTime": "2020-02-05T18:25:30Z",
    "eventSource": "s3.amazonaws.com",
    "eventName": "JobStatusChanged",
    "awsRegion": "us-west-2",
```

```
    "sourceIPAddress": "s3.amazonaws.com",
    "userAgent": "s3.amazonaws.com",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "f907577b-bf3d-4c53-b9ed-8a83a118a554",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "recipientAccountId": "123412341234",
    "serviceEventDetails": {
        "jobId": "d6e58ec4-897a-4b6d-975f-10d7f0fb63ce",
        "jobArn": "arn:aws:s3:us-west-2:181572960644:job/
d6e58ec4-897a-4b6d-975f-10d7f0fb63ce",
        "status": "Complete",
        "jobEventId": "b268784cf0a66749f1a05bce259804f5",
        "failureCodes": [],
        "statusChangeReason": []
    }
}
```

EventBridge aturan untuk melacak peristiwa pekerjaan Operasi Batch S3

Contoh berikut menunjukkan cara membuat aturan di Amazon EventBridge untuk menangkap peristiwa Operasi Batch S3 yang direkam oleh AWS CloudTrail target pilihan Anda.

Untuk melakukan ini, Anda membuat aturan dengan mengikuti semua langkah dalam Membuat EventBridge aturan yang bereaksi terhadap peristiwa. Tempelkan kebijakan pola peristiwa kustom Operasi Batch S3 berikut yang sesuai, dan pilih layanan target yang Anda inginkan.

Kebijakan pola peristiwa kustom Operasi Batch S3

```
{
    "source": [
        "aws.s3"
    ],
    "detail-type": [
        "AWS Service Event via CloudTrail"
    ],
    "detail": {
        "eventSource": [
            "s3.amazonaws.com"
        ],
        "eventName": [
            "JobCreated",
```

```
            "JobStatusChanged"
        ]
    }
}
```

Contoh berikut adalah dua peristiwa Operasi Batch yang dikirim ke Amazon Simple Queue Service (Amazon SQS) dari aturan peristiwa. EventBridge Pekerjaan Operasi Batch melewati berbagai status yang berbeda saat memproses (`New`, `Preparing`, `Active`, dll.), sehingga Anda mungkin akan menerima beberapa pesan untuk setiap pekerjaan.

Example — JobCreated contoh peristiwa

```
{
    "version": "0",
    "id": "51dc8145-541c-5518-2349-56d7dffdf2d8",
    "detail-type": "AWS Service Event via CloudTrail",
    "source": "aws.s3",
    "account": "123456789012",
    "time": "2020-02-27T15:25:49Z",
    "region": "us-east-1",
    "resources": [],
    "detail": {
        "eventVersion": "1.05",
        "userIdentity": {
            "accountId": "11112223334444",
            "invokedBy": "s3.amazonaws.com"
        },
        "eventTime": "2020-02-27T15:25:49Z",
        "eventSource": "s3.amazonaws.com",
        "eventName": "JobCreated",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "s3.amazonaws.com",
        "userAgent": "s3.amazonaws.com",
        "eventID": "7c38220f-f80b-4239-8b78-2ed867b7d3fa",
        "readOnly": false,
        "eventType": "AwsServiceEvent",
        "serviceEventDetails": {
            "jobId": "e849b567-5232-44be-9a0c-40988f14e80c",
            "jobArn": "arn:aws:s3:us-east-1:181572960644:job/
e849b567-5232-44be-9a0c-40988f14e80c",
            "status": "New",
            "jobEventId": "f177ff24f1f097b69768e327038f30ac",
            "failureCodes": [],
```

```
            "statusChangeReason": []
        }
    }
}
```

Example — acara penyelesaian JobStatusChanged pekerjaan

```
{
  "version": "0",
  "id": "c8791abf-2af8-c754-0435-fd869ce25233",
  "detail-type": "AWS Service Event via CloudTrail",
  "source": "aws.s3",
  "account": "123456789012",
  "time": "2020-02-27T15:26:42Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "eventVersion": "1.05",
    "userIdentity": {
      "accountId": "1111222233334444",
      "invokedBy": "s3.amazonaws.com"
    },
    "eventTime": "2020-02-27T15:26:42Z",
    "eventSource": "s3.amazonaws.com",
    "eventName": "JobStatusChanged",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "s3.amazonaws.com",
    "userAgent": "s3.amazonaws.com",
    "eventID": "0238c1f7-c2b0-440b-8dbd-1ed5e5833afb",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "serviceEventDetails": {
      "jobId": "e849b567-5232-44be-9a0c-40988f14e80c",
      "jobArn": "arn:aws:s3:us-east-1:181572960644:job/
e849b567-5232-44be-9a0c-40988f14e80c",
      "status": "Complete",
      "jobEventId": "51f5ac17dba408301d56cd1b2c8d1e9e",
      "failureCodes": [],
      "statusChangeReason": []
    }
  }
}
```

Contoh: Laporan penyelesaian Operasi Batch S3

Saat membuat pekerjaan Operasi Batch S3, Anda dapat meminta laporan penyelesaian untuk semua tugas atau hanya untuk tugas yang gagal. Asalkan setidaknya satu tugas telah berhasil diinvokasi, Operasi Batch S3 menghasilkan laporan untuk tugas yang telah selesai, gagal, atau dibatalkan.

Laporan penyelesaian berisi informasi tambahan untuk setiap pekerjaan, termasuk nama dan versi kunci objek, status, kode kesalahan, dan deskripsi kesalahan apa pun. Deskripsi kesalahan untuk setiap tugas yang gagal dapat digunakan untuk mendiagnosis masalah yang terjadi selama pembuatan tugas, misalnya izin.

> **Note**
>
> Laporan penyelesaian selalu dienkripsi dengan kunci terkelola Amazon S3 (SSE-S3).

Example — File hasil manifes tingkat atas

File tingkat atas `manifest.json` berisi lokasi setiap laporan yang berhasil dan (jika pekerjaan mengalami kegagalan) lokasi laporan yang gagal, seperti yang ditunjukkan dalam contoh berikut.

```
{
    "Format": "Report_CSV_20180820",
    "ReportCreationDate": "2019-04-05T17:48:39.725Z",
    "Results": [
        {
            "TaskExecutionStatus": "succeeded",
            "Bucket": "my-job-reports",
            "MD5Checksum": "83b1c4cbe93fc893f54053697e10fd6e",
            "Key": "job-f8fb9d89-a3aa-461d-bddc-ea6a1b131955/
results/6217b0fab0de85c408b4be96aeaca9b195a7daa5.csv"
        },
        {
            "TaskExecutionStatus": "failed",
            "Bucket": "my-job-reports",
            "MD5Checksum": "22ee037f3515975f7719699e5c416eaa",
            "Key": "job-f8fb9d89-a3aa-461d-bddc-ea6a1b131955/results/
b2ddad417e94331e9f37b44f1faf8c7ed5873f2e.csv"
        }
    ],
    "ReportSchema": "Bucket, Key, VersionId, TaskStatus, ErrorCode, HTTPStatusCode,
 ResultMessage"
```

```
}
```

Example — Laporan tugas yang gagal

Laporan pekerjaan yang gagal berisi informasi berikut untuk semua pekerjaan yang gagal:

- `Bucket`
- `Key`
- `VersionId`
- `TaskStatus`
- `ErrorCode`
- `HTTPStatusCode`
- `ResultMessage`

Contoh laporan berikut menunjukkan kasus di mana AWS Lambda fungsi habis waktu, menyebabkan kegagalan melebihi ambang kegagalan. Kemudian ditandai sebagai `PermanentFailure`.

```
amzn-s3-demo-bucket1,image_14975,,failed,200,PermanentFailure,"Lambda returned
 function error: {""errorMessage"":""2019-04-05T17:35:21.155Z 2845ca0d-38d9-4c4b-
abcf-379dc749c452 Task timed out after 3.00 seconds""}"
amzn-s3-demo-bucket1,image_15897,,failed,200,PermanentFailure,"Lambda returned
 function error: {""errorMessage"":""2019-04-05T17:35:29.610Z 2d0a330b-de9b-425f-
b511-29232fde5fe4 Task timed out after 3.00 seconds""}"
amzn-s3-demo-bucket1,image_14819,,failed,200,PermanentFailure,"Lambda returned
 function error: {""errorMessage"":""2019-04-05T17:35:22.362Z fcf5efde-74d4-4e6d-b37a-
c7f18827f551 Task timed out after 3.00 seconds""}"
amzn-s3-demo-bucket1,image_15930,,failed,200,PermanentFailure,"Lambda returned
 function error: {""errorMessage"":""2019-04-05T17:35:29.809Z 3dd5b57c-4a4a-48aa-8a35-
cbf027b7957e Task timed out after 3.00 seconds""}"
amzn-s3-demo-bucket1,image_17644,,failed,200,PermanentFailure,"Lambda
 returned function error: {""errorMessage"":""2019-04-05T17:35:46.025Z
 10a764e4-2b26-4d8c-9056-1e1072b4723f Task timed out after 3.00 seconds""}"
amzn-s3-demo-bucket1,image_17398,,failed,200,PermanentFailure,"Lambda returned
 function error: {""errorMessage"":""2019-04-05T17:35:44.661Z 1e306352-4c54-4eba-
aee8-4d02f8c0235c Task timed out after 3.00 seconds""}"
```

Example — Laporan tugas yang berhasil

Laporan tugas yang berhasil berisi hal-hal berikut untuk tugas-tugas yang berhasil:

- `Bucket`
- `Key`
- `VersionId`
- `TaskStatus`
- `ErrorCode`
- `HTTPStatusCode`
- `ResultMessage`

Dalam contoh berikut, fungsi Lambda berhasil menyalin objek Amazon S3 ke bucket lain. Respons Amazon S3 yang dikembalikan akan diteruskan kembali ke S3 Batch Operations dan kemudian ditulis ke dalam laporan penyelesaian akhir.

```
amzn-s3-demo-bucket1,image_17775,,succeeded,200,,"{u'CopySourceVersionId':
 'xVR78haVKlRnurYofbTfYr3ufYbktF8h', u'CopyObjectResult': {u'LastModified':
 datetime.datetime(2019, 4, 5, 17, 35, 39, tzinfo=tzlocal()), u'ETag':
 '""fe66f4390c50f29798f040d7aae72784""'}, 'ResponseMetadata': {'HTTPStatusCode':
 200, 'RetryAttempts': 0, 'HostId': 'nXNaClIMxEJzWNmeMNQV2KpjbaCJLn0OGoXWZpuVOFS/
iQYWxb3QtTvzX9SVfx2lA3oTKLwImKw=', 'RequestId': '3ED5852152014362', 'HTTPHeaders':
 {'content-length': '234', 'x-amz-id-2': 'nXNaClIMxEJzWNmeMNQV2KpjbaCJLn0OGoXWZpuVOFS/
iQYWxb3QtTvzX9SVfx2lA3oTKLwImKw=', 'x-amz-copy-source-version-id':
 'xVR78haVKlRnurYofbTfYr3ufYbktF8h', 'server': 'AmazonS3', 'x-amz-request-id':
 '3ED5852152014362', 'date': 'Fri, 05 Apr 2019 17:35:39 GMT', 'content-type':
 'application/xml'}}}"
amzn-s3-demo-bucket1,image_17763,,succeeded,200,,"{u'CopySourceVersionId':
 '6HjOUSim4Wj6BTcbxToXW44pSZ.40pwq', u'CopyObjectResult': {u'LastModified':
 datetime.datetime(2019, 4, 5, 17, 35, 39, tzinfo=tzlocal()),
 u'ETag': '""fe66f4390c50f29798f040d7aae72784""'}, 'ResponseMetadata':
 {'HTTPStatusCode': 200, 'RetryAttempts': 0, 'HostId': 'GiCZNYr8LHd/
Thyk6beTRP96IGZk2sYxujLe13TuuLpq6U2RD3we0YoluuIdm1PRvkMwnEW1aFc=', 'RequestId':
 '1BC9F5B1B95D7000', 'HTTPHeaders': {'content-length': '234', 'x-amz-id-2':
 'GiCZNYr8LHd/Thyk6beTRP96IGZk2sYxujLe13TuuLpq6U2RD3we0YoluuIdm1PRvkMwnEW1aFc=', 'x-
amz-copy-source-version-id': '6HjOUSim4Wj6BTcbxToXW44pSZ.40pwq', 'server': 'AmazonS3',
 'x-amz-request-id': '1BC9F5B1B95D7000', 'date': 'Fri, 05 Apr 2019 17:35:39 GMT',
 'content-type': 'application/xml'}}}"
amzn-s3-demo-bucket1,image_17860,,succeeded,200,,"{u'CopySourceVersionId':
 'm.MDD0g_QsUnYZ8TBzVFrp.TmjN8PJyX', u'CopyObjectResult': {u'LastModified':
 datetime.datetime(2019, 4, 5, 17, 35, 40, tzinfo=tzlocal()), u'ETag':
 '""fe66f4390c50f29798f040d7aae72784""'}, 'ResponseMetadata': {'HTTPStatusCode':
 200, 'RetryAttempts': 0, 'HostId': 'F9ooZOgpE5g9sNgBZxjdiPHqB4+0DNWgj3qbsir
```

```
+sKai4fv7rQEcF2fBN1VeeFc2WH45a9ygb2g=', 'RequestId': '8D9CA56A56813DF3', 'HTTPHeaders':
 {'content-length': '234', 'x-amz-id-2': 'F9ooZOgpE5g9sNgBZxjdiPHqB4+0DNWgj3qbsir
+sKai4fv7rQEcF2fBN1VeeFc2WH45a9ygb2g=', 'x-amz-copy-source-version-id':
 'm.MDD0g_QsUnYZ8TBzVFrp.TmjN8PJyX', 'server': 'AmazonS3', 'x-amz-request-id':
 '8D9CA56A56813DF3', 'date': 'Fri, 05 Apr 2019 17:35:40 GMT', 'content-type':
 'application/xml'}}}"
```

Mengendalikan pekerjaan akses dan pelabelan menggunakan tag

Anda dapat melabeli dan mengontrol akses ke pekerjaan Operasi Batch S3 Anda dengan menambahkan tag. Tanda dapat digunakan untuk mengidentifikasi siapa yang bertanggung jawab atas tugas Operasi Batch. Adanya tanda pekerjaan dapat memberikan atau membatasi kemampuan pengguna untuk membatalkan pekerjaan, mengaktifkan pekerjaan dalam status konfirmasi, atau mengubah tingkat prioritas pekerjaan. Anda dapat membuat pekerjaan dengan tag yang dilampirkan padanya, dan Anda dapat menambahkan tag ke pekerjaan setelah dibuat. Setiap tanda merupakan pasangan nilai kunci yang dapat disertakan saat Anda membuat pekerjaan atau saat memperbaruinya.

> **Warning**
>
> Pastikan tag pekerjaan Anda tidak berisi informasi rahasia atau data pribadi apa pun.

Perhatikan contoh penandaan berikut ini: Misalnya, Anda ingin departemen Keuangan untuk membuat pekerjaan Operasi Batch. Anda dapat menulis kebijakan AWS Identity and Access Management (IAM) yang memungkinkan pengguna untuk memanggil`CreateJob`, asalkan pekerjaan dibuat dengan `Department` tag yang ditetapkan nilainya. `Finance` Selain itu, Anda dapat melampirkan kebijakan tersebut kepada semua pengguna yang merupakan anggota departemen Keuangan.

Melanjutkan dengan contoh ini, Anda dapat menulis kebijakan yang memungkinkan pengguna untuk memperbarui prioritas semua pekerjaan dengan tanda yang diinginkan, atau membatalkan semua pekerjaan yang memiliki tanda tersebut. Untuk informasi selengkapnya, lihat the section called “Mengontrol izin”.

Anda dapat menambahkan tanda pada pekerjaan Operasi Batch S3 yang baru saat Anda membuatnya, atau menambahkannya ke pekerjaan yang sudah ada.

Perhatikan pembatasan tanda berikut:

- Anda dapat menghubungkan hingga 50 tanda dengan suatu pekerjaan selama tanda tersebut memiliki kunci tanda yang unik.
- Kunci tanda dapat terdiri dari hingga 128 karakter Unicode, dan nilai tanda dapat terdiri dari hingga 256 karakter Unicode.
- Kunci dan nilai peka terhadap huruf besar dan kecil.

Untuk informasi selengkapnya tentang pembatasan tanda, lihat Pembatasan Tanda yang Ditentukan Pengguna dalam Panduan Pengguna AWS Manajemen Penagihan dan Biaya .

Operasi API yang terkait dengan penandaan pekerjaan Operasi Batch S3

Amazon S3 mendukung operasi API berikut ini yang dikhususkan untuk penandaan pekerjaan Operasi Batch S3:

- GetJobTagging— Mengembalikan set tag yang terkait dengan pekerjaan Operasi Batch.
- PutJobTagging— Mengganti set tag yang terkait dengan pekerjaan. Terdapat dua skenario berbeda untuk manajemen penandaan pekerjaan Operasi Batch S3 dengan menggunakan tindakan API ini:
  - Job tidak memiliki tag - Anda dapat menambahkan satu set tag ke pekerjaan (pekerjaan tidak memiliki tag sebelumnya).
  - Job memiliki satu set tag yang ada — Untuk memodifikasi set tag yang ada, Anda dapat mengganti set tag yang ada seluruhnya, atau membuat perubahan dalam set tag yang ada dengan mengambil set tag yang ada menggunakan GetJobTagging, memodifikasi set tag tersebut, dan menggunakan tindakan API ini untuk mengganti set tag dengan yang telah Anda modifikasi.

    > **Note**
    >
    > Jika Anda mengirim permintaan ini dengan kumpulan tanda kosong, Operasi Batch S3 akan menghapus kumpulan tanda yang sudah ada pada objek. Jika Anda menggunakan metode ini, Anda akan dikenakan biaya untuk Permintaan Tingkat 1 (PUT). Untuk informasi selengkapnya, lihat Harga Amazon S3.
    >
    > Untuk menghapus tanda yang sudah ada pada pekerjaan Operasi Batch Anda, tindakan `DeleteJobTagging` lebih disukai karena memberikan hasil yang sama tanpa dikenakan biaya.

- DeleteJobTagging— Menghapus set tag yang terkait dengan pekerjaan Operasi Batch.

Membuat tugas Operasi Batch dengan tanda tugas yang digunakan untuk pelabelan

Anda dapat memberi label dan mengontrol akses ke pekerjaan Operasi Batch Amazon S3 Anda dengan menambahkan tag. Tanda dapat digunakan untuk mengidentifikasi siapa yang bertanggung jawab atas tugas Operasi Batch. Anda dapat membuat tugas dengan tanda yang telah dilampirkan, atau menambahkan tanda ke tugas setelah tugas dibuat. Untuk informasi selengkapnya, lihat the section called “Menggunakan tanda”.

Menggunakan AWS CLI

AWS CLI Contoh berikut membuat `S3PutObjectCopy` pekerjaan Operasi Batch S3 menggunakan tag pekerjaan sebagai label untuk pekerjaan tersebut.

1. Pilih tindakan atau `OPERATION` yang akan dilaksanakan oleh Operasi Batch, lalu pilih `TargetResource`.

```
read -d '' OPERATION <<EOF
{
  "S3PutObjectCopy": {
    "TargetResource": "arn:aws:s3:::amzn-s3-demo-destination-bucket"
  }
}
EOF
```

2. Identifikasi tugas `TAGS` yang Anda inginkan untuk tugas tersebut. Dalam hal ini, Anda akan menerapkan dua tanda, `department` dan `FiscalYear`, masing-masing dengan nilai `Marketing` dan `2020`.

```
read -d '' TAGS <<EOF
[
  {
    "Key": "department",
    "Value": "Marketing"
  },
  {
    "Key": "FiscalYear",
    "Value": "2020"
  }
]
EOF
```

3. Tentukan `MANIFEST` untuk tugas Operasi Batch.

```
read -d '' MANIFEST <<EOF
{
  "Spec": {
    "Format": "EXAMPLE_S3BatchOperations_CSV_20180820",
    "Fields": [
      "Bucket",
      "Key"
    ]
  },
  "Location": {
    "ObjectArn": "arn:aws:s3:::amzn-s3-demo-manifest-bucket/example_manifest.csv",
    "ETag": "example-5dc7a8bfb90808fc5d546218"
  }
}
EOF
```

4. Konfigurasikan REPORT untuk tugas Operasi Batch.

```
read -d '' REPORT <<EOF
{
  "Bucket": "arn:aws:s3:::amzn-s3-demo-completion-report-bucket",
  "Format": "Example_Report_CSV_20180820",
  "Enabled": true,
  "Prefix": "reports/copy-with-replace-metadata",
  "ReportScope": "AllTasks"
}
EOF
```

5. Jalankan tindakan `create-job` untuk membuat tugas Operasi Batch dengan input yang telah ditetapkan pada langkah sebelumnya.

```
aws \
    s3control create-job \
    --account-id 123456789012 \
    --manifest "${MANIFEST//$'\n'}" \
    --operation "${OPERATION//$'\n'/}" \
    --report "${REPORT//$'\n'}" \
    --priority 10 \
    --role-arn arn:aws:iam::123456789012:role/batch-operations-role \
    --tags "${TAGS//$'\n'/}" \
    --client-request-token "$(uuidgen)" \
    --region us-west-2 \
```

```
    --description "Copy with Replace Metadata";
```

Menggunakan AWS SDK for Java

Example

Contoh berikut membuat tugas Operasi Batch S3 dengan tanda menggunakan AWS SDK untuk Java.

```
public String createJob(final AWSS3ControlClient awss3ControlClient) {
    final String manifestObjectArn = "arn:aws:s3:::amzn-s3-demo-manifest-bucket/
manifests/10_manifest.csv";
    final String manifestObjectVersionId = "example-5dc7a8bfb90808fc5d546218";

    final JobManifestLocation manifestLocation = new JobManifestLocation()
            .withObjectArn(manifestObjectArn)
            .withETag(manifestObjectVersionId);

    final JobManifestSpec manifestSpec =
            new
 JobManifestSpec().withFormat(JobManifestFormat.S3InventoryReport_CSV_20161130);

    final JobManifest manifestToPublicApi = new JobManifest()
            .withLocation(manifestLocation)
            .withSpec(manifestSpec);

    final String jobReportBucketArn = "arn:aws:s3:::amzn-s3-demo-completion-report-
bucket";
    final String jobReportPrefix = "example-job-reports";

    final JobReport jobReport = new JobReport()
            .withEnabled(true)
            .withReportScope(JobReportScope.AllTasks)
            .withBucket(jobReportBucketArn)
            .withPrefix(jobReportPrefix)
            .withFormat(JobReportFormat.Report_CSV_20180820);

    final String lambdaFunctionArn = "arn:aws:lambda:us-
west-2:123456789012:function:example-function";

    final JobOperation jobOperation = new JobOperation()
            .withLambdaInvoke(new
 LambdaInvokeOperation().withFunctionArn(lambdaFunctionArn));
```

```
    final S3Tag departmentTag = new
 S3Tag().withKey("department").withValue("Marketing");
    final S3Tag fiscalYearTag = new S3Tag().withKey("FiscalYear").withValue("2020");

    final String roleArn = "arn:aws:iam::123456789012:role/example-batch-operations-
role";
    final Boolean requiresConfirmation = true;
    final int priority = 10;

    final CreateJobRequest request = new CreateJobRequest()
            .withAccountId("123456789012")
            .withDescription("Test lambda job")
            .withManifest(manifestToPublicApi)
            .withOperation(jobOperation)
            .withPriority(priority)
            .withRoleArn(roleArn)
            .withReport(jobReport)
            .withTags(departmentTag, fiscalYearTag)
            .withConfirmationRequired(requiresConfirmation);

    final CreateJobResult result = awss3ControlClient.createJob(request);

    return result.getJobId();
}
```

Menghapus tanda dari tugas Operasi Batch S3

Anda dapat menggunakan contoh ini untuk menghapus tag dari pekerjaan Operasi Batch Amazon S3.

Menggunakan AWS CLI

Contoh berikut menghapus tanda dari tugas Operasi Batch menggunakan AWS CLI.

```
aws \
    s3control delete-job-tagging \
    --account-id 123456789012 \
    --job-id Example-e25a-4ed2-8bee-7f8ed7fc2f1c \
    --region us-east-1
```

Menghapus tanda tugas dari tugas Operasi Batch

Example

Contoh berikut menghapus tanda dari tugas Operasi Batch S3 menggunakan AWS SDK untuk Java.

```
public void deleteJobTagging(final AWSS3ControlClient awss3ControlClient,
                             final String jobId) {
    final DeleteJobTaggingRequest deleteJobTaggingRequest = new
 DeleteJobTaggingRequest()
            .withJobId(jobId);

    final DeleteJobTaggingResult deleteJobTaggingResult =
                awss3ControlClient.deleteJobTagging(deleteJobTaggingRequest);
}
```

Menambahkan tag pekerjaan ke pekerjaan Operasi Batch yang ada

Anda dapat menggunakan operasi PutJobTaggingAPI untuk menambahkan tag pekerjaan ke pekerjaan Operasi Batch Amazon S3 yang ada. Untuk informasi selengkapnya, lihat contoh berikut ini.

Menggunakan AWS CLI

Berikut ini adalah contoh penggunaan `s3control put-job-tagging` untuk menambahkan tag pekerjaan ke pekerjaan Operasi Batch S3 Anda dengan menggunakan. AWS CLI Untuk menggunakan contoh, ganti *`user input placeholders`* dengan informasi Anda sendiri.

> **Note**
>
> Jika Anda mengirim permintaan ini dengan set tag kosong, Operasi Batch menghapus set tag yang ada pada objek. Namun, jika Anda menggunakan pendekatan ini, Anda akan dikenakan biaya untuk Permintaan Tingkat 1 (PUT). Untuk informasi selengkapnya, lihat Harga Amazon S3.
>
> Sebagai gantinya, untuk menghapus tag yang ada untuk pekerjaan Operasi Batch Anda, sebaiknya gunakan `DeleteJobTagging` operasi karena tag tersebut mencapai hasil yang sama tanpa menimbulkan biaya.

1. Identifikasi tugas TAGS yang Anda inginkan untuk tugas tersebut. Dalam hal ini, Anda akan menerapkan dua tanda, *department* dan *FiscalYear*, masing-masing dengan nilai *Marketing* dan *2020*.

```
read -d '' TAGS <<EOF
[
  {
    "Key": "department",
    "Value": "Marketing"
  },
  {
    "Key": "FiscalYear",
    "Value": "2020"
  }
]
EOF
```

2. Jalankan `put-job-tagging` perintah berikut dengan parameter yang diperlukan:

```
aws \
    s3control put-job-tagging \
    --account-id 123456789012 \
    --tags "${TAGS//$'\n'/}" \
    --job-id Example-e25a-4ed2-8bee-7f8ed7fc2f1c \
    --region us-east-1
```

Menggunakan AWS SDK for Java

Example

Contoh berikut menempatkan tag pekerjaan Operasi Batch S3 dengan menggunakan. AWS SDK untuk Java

```
public void putJobTagging(final AWSS3ControlClient awss3ControlClient,
                          final String jobId) {
    final S3Tag departmentTag = new
 S3Tag().withKey("department").withValue("Marketing");
    final S3Tag fiscalYearTag = new S3Tag().withKey("FiscalYear").withValue("2020");

    final PutJobTaggingRequest putJobTaggingRequest = new PutJobTaggingRequest()
            .withJobId(jobId)
```

```
            .withTags(departmentTag, fiscalYearTag);

    final PutJobTaggingResult putJobTaggingResult =
 awss3ControlClient.putJobTagging(putJobTaggingRequest);
}
```

Mendapatkan tanda dari tugas Operasi Batch S3

Untuk mengambil tag pekerjaan Operasi Batch Amazon S3, Anda dapat `GetJobTagging` menggunakan operasi API. Untuk informasi selengkapnya, lihat contoh berikut ini.

Menggunakan AWS CLI

Contoh berikut menunjukkan cara mendapatkan tanda tugas Operasi Batch menggunakan AWS CLI. Untuk menggunakan contoh ini, ganti *`user input placeholders`*dengan informasi Anda sendiri.

```
aws \
    s3control get-job-tagging \
    --account-id 123456789012 \
    --job-id Example-e25a-4ed2-8bee-7f8ed7fc2f1c \
    --region us-east-1
```

Menggunakan AWS SDK for Java

Example

Contoh berikut menunjukkan cara mendapatkan tanda tugas Operasi Batch S3 menggunakan AWS SDK untuk Java.

```
public List<S3Tag> getJobTagging(final AWSS3ControlClient awss3ControlClient,
                                 final String jobId) {
    final GetJobTaggingRequest getJobTaggingRequest = new GetJobTaggingRequest()
            .withJobId(jobId);

    final GetJobTaggingResult getJobTaggingResult =
            awss3ControlClient.getJobTagging(getJobTaggingRequest);

    final List<S3Tag> tags = getJobTaggingResult.getTags();

    return tags;
}
```

Mengontrol izin untuk Operasi Batch menggunakan tag pekerjaan

Untuk membantu mengelola pekerjaan Operasi Batch Amazon S3, Anda dapat menambahkan tag pekerjaan. Dengan tanda tugas, Anda dapat mengontrol akses atas tugas Operasi Batch dan menerapkan tanda tersebut saat pekerjaan lain dibuat.

Anda dapat menerapkan hingga 50 tanda tugas untuk setiap tugas Operasi Batch. Dengan menggunakan tag, Anda dapat mengatur kebijakan terperinci untuk membatasi kumpulan pengguna yang dapat mengedit pekerjaan. Tanda tugas dapat memberikan atau membatasi kemampuan pengguna untuk membatalkan tugas, mengaktifkan tugas dalam status konfirmasi, atau mengubah tingkat prioritas tugas. Selain itu, Anda dapat memberlakukan tanda yang diterapkan ke semua tugas baru, dan menentukan pasangan nilai kunci yang diizinkan untuk tanda tersebut. Anda dapat mengungkapkan semua kondisi ini dengan menggunakan bahasa kebijakan AWS Identity and Access Management (IAM). Untuk informasi selengkapnya, lihat Kunci tindakan, sumber daya, dan kondisi untuk Amazon S3 di Referensi Otorisasi Layanan.

Untuk informasi selengkapnya tentang izin operasi S3 API menurut jenis sumber daya S3, lihat. Izin yang diperlukan untuk operasi API Amazon S3

Contoh berikut menunjukkan cara menggunakan tanda tugas Operasi Batch S3 untuk mengizinkan pengguna dalam membuat dan mengedit tugas yang khusus dijalankan di dalam departemen (misalnya, departemen Keuangan atau Kepatuhan). Anda juga dapat menentukan tugas berdasarkan tahap pengembangan yang terkait dengan tugas tersebut, seperti QA atau Produksi.

Dalam contoh ini, Anda menggunakan tag pekerjaan Operasi Batch S3 dalam kebijakan IAM untuk memberikan izin kepada pengguna untuk membuat dan mengedit hanya pekerjaan yang sedang dijalankan dalam departemen mereka. Tentukan pekerjaan berdasarkan tahap pengembangan yang terkait dengan pekerjaan tersebut, seperti QA atau Produksi.

Contoh berikut menggunakan departemen berikut, dengan masing-masing departemen menggunakan Operasi Batch dengan cara yang berbeda:

- Keuangan
- Kepatuhan
- Kecerdasan Bisnis
- Teknik

Topik

- Mengontrol akses dengan menetapkan tanda ke pengguna dan sumber daya
- Menandai pekerjaan Operasi Batch berdasarkan tahap dan memberlakukan batas prioritas pekerjaan

Mengontrol akses dengan menetapkan tanda ke pengguna dan sumber daya

Dalam skenario ini, administrator menggunakan kontrol akses berbasis atribut (ABAC). ABAC adalah strategi otorisasi IAM yang mendefinisikan izin dengan melampirkan tag ke pengguna dan sumber daya. AWS

Pengguna dan tugas diberi salah satu tanda departemen berikut ini:

Kunci : Nilai

- `department : Finance`
- `department : Compliance`
- `department : BusinessIntelligence`
- `department : Engineering`

> **Note**
>
> Kunci dan nilai tanda tugas peka huruf besar kecil.

Dengan menggunakan strategi kontrol akses ABAC, Anda mengizinkan pengguna di departemen Keuangan untuk membuat dan mengelola pekerjaan Operasi Batch S3 dalam departemen mereka dengan mengasosiasikan tanda `department=Finance` dengan pengguna mereka.

Selain itu, Anda dapat melampirkan kebijakan terkelola ke pengguna IAM yang memungkinkan setiap pengguna di perusahaan untuk membuat atau memodifikasi tugas Operasi Batch S3 dalam departemen mereka masing-masing.

Kebijakan dalam contoh ini mencakup tiga pernyataan kebijakan:

- Pernyataan pertama dalam kebijakan mengizinkan pengguna untuk membuat tugas Operasi Batch asalkan permintaan pembuatan tugas menyertakan tanda tugas yang sesuai dengan departemen masing-masing. Ini ditunjukkan dengan menggunakan sintaks `"${aws:PrincipalTag/department}"`, yang diganti dengan tanda departemen pengguna pada waktu evaluasi kebijakan.

Syarat ini terpenuhi ketika nilai yang diberikan untuk tanda departemen dalam permintaan (`"aws:RequestTag/department"`) sesuai dengan departemen pengguna.

- Pernyataan kedua dalam kebijakan ini memungkinkan pengguna untuk mengubah prioritas tugas atau memperbarui statusnya asalkan tugas yang sedang diperbarui sesuai dengan departemen pengguna.
- Pernyataan ketiga memungkinkan pengguna untuk memperbarui tanda tugas Operasi Batch setiap saat melalui permintaan `PutJobTagging` asalkan (1) tanda departemen mereka dipertahankan dan (2) pekerjaan yang diperbarui berada dalam departemen mereka.

```
{
      "Version": "2012-10-17",
      "Statement": [
            {
                  "Effect": "Allow",
                  "Action": "s3:CreateJob",
                  "Resource": "*",
                  "Condition": {
                        "StringEquals": {
                              "aws:RequestTag/department": "${aws:PrincipalTag/
department}"
                }
            }
        },
            {
                  "Effect": "Allow",
                  "Action": [
                        "s3:UpdateJobPriority",
                        "s3:UpdateJobStatus"
            ],
                  "Resource": "*",
                  "Condition": {
                        "StringEquals": {
                              "aws:ResourceTag/department": "${aws:PrincipalTag/
department}"
                }
            }
        },
            {
                  "Effect": "Allow",
                  "Action": "s3:PutJobTagging",
                  "Resource": "*",
```

```
                "Condition": {
                      "StringEquals": {
                            "aws:RequestTag/department": "${aws:PrincipalTag/
department}",
                            "aws:ResourceTag/department": "${aws:PrincipalTag/
department}"
                }
             }
        }
    ]
}
```

Menandai pekerjaan Operasi Batch berdasarkan tahap dan memberlakukan batas prioritas pekerjaan

Semua pekerjaan Operasi Batch S3 memiliki prioritas numerik, yang digunakan Amazon S3 untuk memutuskan urutan pelaksanaan tugas. Untuk contoh ini, Anda membatasi prioritas maksimum yang dapat ditetapkan oleh sebagian besar pengguna ke tugas, dengan rentang prioritas lebih tinggi yang diperuntukkan bagi pengguna dengan hak istimewa dalam jumlah terbatas, sebagai berikut:

- Kisaran prioritas tahap QA (rendah): 1-100
- Kisaran prioritas tahap produksi (tinggi): 1-300

Untuk melakukannya, gunakan set tanda baru yang mewakili tahapan tugas:

Kunci : Nilai

- `stage : QA`
- `stage : Production`

Membuat dan memperbarui pekerjaan prioritas rendah dalam sebuah departemen

Kebijakan ini memperkenalkan dua pembatasan baru pada pembuatan dan pembaruan tugas Operasi Batch S3, selain pembatasan berbasis departemen:

- Ini memungkinkan pengguna membuat atau memperbarui tugas di departemen mereka dengan ketentuan baru yang memerlukan tugas tersebut untuk menyertakan tanda `stage=QA`.
- Ini memungkinkan pengguna membuat atau memperbarui prioritas tugas hingga prioritas maksimum baru sebesar 100.

```
{
        "Version": "2012-10-17",
        "Statement": [
        {
        "Effect": "Allow",
        "Action": "s3:CreateJob",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:RequestTag/department": "${aws:PrincipalTag/department}",
                "aws:RequestTag/stage": "QA"
            },
            "NumericLessThanEquals": {
                "s3:RequestJobPriority": 100
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": [
            "s3:UpdateJobStatus"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:ResourceTag/department": "${aws:PrincipalTag/department}"
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": "s3:UpdateJobPriority",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:ResourceTag/department": "${aws:PrincipalTag/department}",
                "aws:ResourceTag/stage": "QA"
            },
            "NumericLessThanEquals": {
                "s3:RequestJobPriority": 100
            }
        }
    },
```

```
    {
        "Effect": "Allow",
        "Action": "s3:PutJobTagging",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:RequestTag/department" : "${aws:PrincipalTag/department}",
                "aws:ResourceTag/department": "${aws:PrincipalTag/department}",
                "aws:RequestTag/stage": "QA",
                "aws:ResourceTag/stage": "QA"
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": "s3:GetJobTagging",
        "Resource": "*"
    }
    ]
}
```

Membuat dan memperbarui tugas prioritas tinggi dalam sebuah departemen

Sebagian kecil pengguna mungkin memerlukan kemampuan untuk membuat tugas dengan prioritas tinggi di QA maupun Produksi. Untuk mendukung kebutuhan ini, buat kebijakan terkelola yang diadaptasi dari kebijakan prioritas rendah pada bagian sebelumnya.

Kebijakan ini mengatur hal-hal berikut:

- Memungkinkan pengguna membuat atau memperbarui tugas di departemen mereka dengan tanda `stage=QA` atau `stage=Production`.
- Memungkinkan pengguna membuat atau memperbarui prioritas pekerjaan hingga maksimum 300.

```
{
      "Version": "2012-10-17",
      "Statement": [
          {
                "Effect": "Allow",
                "Action": "s3:CreateJob",
                "Resource": "*",
                "Condition": {
                      "ForAnyValue:StringEquals": {
```

```
                        "aws:RequestTag/stage": [
                              "QA",
                              "Production"
                    ]
                },
                      "StringEquals": {
                            "aws:RequestTag/department": "${aws:PrincipalTag/
department}"
                },
                      "NumericLessThanEquals": {
                            "s3:RequestJobPriority": 300
                }
            }
        },
          {
                "Effect": "Allow",
                "Action": [
                      "s3:UpdateJobStatus"
            ],
                "Resource": "*",
                "Condition": {
                      "StringEquals": {
                            "aws:ResourceTag/department": "${aws:PrincipalTag/
department}"
                }
            }
        },
          {
                "Effect": "Allow",
                "Action": "s3:UpdateJobPriority",
                "Resource": "*",
                "Condition": {
                      "ForAnyValue:StringEquals": {
                            "aws:ResourceTag/stage": [
                                  "QA",
                                  "Production"
                    ]
                },
                      "StringEquals": {
                            "aws:ResourceTag/department": "${aws:PrincipalTag/
department}"
                },
                      "NumericLessThanEquals": {
                            "s3:RequestJobPriority": 300
```

```
                }
            }
        },
          {
                "Effect": "Allow",
                "Action": "s3:PutJobTagging",
                "Resource": "*",
                "Condition": {
                      "StringEquals": {
                            "aws:RequestTag/department": "${aws:PrincipalTag/
department}",
                            "aws:ResourceTag/department": "${aws:PrincipalTag/
department}"
                },
                      "ForAnyValue:StringEquals": {
                            "aws:RequestTag/stage": [
                                  "QA",
                                  "Production"
                    ],
                            "aws:ResourceTag/stage": [
                                  "QA",
                                  "Production"
                    ]
                }
            }
        }
    ]
}
```

Mengelola S3 Object Lock menggunakan Operasi Batch S3

Anda dapat menggunakan Operasi Batch S3 untuk melakukan operasi batch berskala besar pada objek Amazon S3. Operasi Batch S3 dapat melakukan satu operasi untuk sejumlah objek Amazon S3 yang telah Anda tentukan. Satu pekerjaan dapat melakukan operasi tertentu pada miliaran objek yang berisi data berukuran exabyte. Amazon S3 melacak kemajuan, mengirim pemberitahuan, dan menyimpan laporan penyelesaian terperinci dari semua tindakan, menyediakan pengalaman yang sepenuhnya terkelola, dapat diaudit, dan nirserver. Anda dapat menggunakan Operasi Batch S3 melalui konsol Amazon S3 AWS CLI, AWS SDKs, atau Amazon S3 REST API.

Dengan Kunci Objek S3, Anda dapat menambahkan penahanan hukum pada versi objek. Seperti mengatur periode retensi, penyimpanan yang sah mencegah agar versi objek tidak ditimpa atau

dihapus. Namun, penahanan hukum tidak memiliki periode retensi terkait dan tetap berlaku sampai penahanan hukum dihapus. Untuk informasi selengkapnya, lihat Pegangan hukum S3 Object Lock.

Untuk menggunakan Operasi Batch S3 dengan Kunci Objek untuk menambahkan penahanan legal ke banyak objek Amazon S3 sekaligus, lihat topik berikut.

Topik

- Mengaktifkan Kunci Objek S3 menggunakan Operasi Batch S3
- Mengatur retensi Kunci Objek menggunakan Operasi Batch
- Menggunakan Operasi Batch S3 dengan mode kepatuhan retensi Kunci Objek S3
- Gunakan Operasi Batch S3 dengan mode tata kelola penyimpanan Kunci Objek S3
- Menggunakan Operasi Batch S3 untuk mematikan penahanan hukum Kunci Objek S3

Mengaktifkan Kunci Objek S3 menggunakan Operasi Batch S3

Anda dapat menggunakan Operasi Batch Amazon S3 dengan Kunci Objek S3 untuk mengelola retensi atau mengaktifkan penahanan hukum untuk banyak objek Amazon S3 sekaligus. Anda menetapkan daftar objek target dalam manifest Anda dan mengirimkannya ke Batch Operations untuk diselesaikan. Untuk informasi selengkapnya, lihat the section called “Retensi Kunci Objek” dan the section called “Penyimpanan yang sah Kunci Objek”.

Contoh berikut menunjukkan cara membuat peran AWS Identity and Access Management (IAM) dengan izin Operasi Batch S3 dan memperbarui izin peran untuk membuat pekerjaan yang mengaktifkan Object Lock. Anda juga harus memiliki `CSV` manifes yang mengidentifikasi objek untuk pekerjaan Operasi Batch S3 Anda. Untuk informasi selengkapnya, lihat the section called “Menentukan manifes”.

Untuk menggunakan contoh berikut, ganti *`user input placeholders`* dengan informasi Anda sendiri.

Menggunakan AWS CLI

1. Membuat peran IAM dan menetapkan izin Operasi Batch S3 yang akan dijalankan.

   Langkah ini diperlukan untuk semua pekerjaan Operasi Batch S3.

```
export AWS_PROFILE='aws-user'

read -d '' batch_operations_trust_policy <<EOF
```

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "batchoperations.s3.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
aws iam create-role --role-name batch_operations-objectlock \
--assume-role-policy-document "${batch_operations_trust_policy}"
```

2. Menetapkan Operasi Batch S3 dengan Kunci Objek S3 yang akan dijalankan.

   Pada langkah ini, izinkan peran untuk melakukan hal-hal berikut ini:

   a. Menjalankan Kunci Objek pada bucket S3 yang berisi objek target yang akan dijalankan oleh Operasi Batch.

   b. Membaca bucket S3 tempat file CSV manifes dan objeknya berada.

   c. Menulis hasil pekerjaan Operasi Batch S3 ke bucket pelaporan.

```
read -d '' batch_operations_permissions <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:GetBucketObjectLockConfiguration",
            "Resource": [
                "arn:aws:s3:::{{amzn-s3-demo-manifest-bucket}}"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
```

```
                "s3:GetObjectVersion",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::{{amzn-s3-demo-manifest-bucket}}/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::{{amzn-s3-demo-completion-report-bucket}}/*"
            ]
        }
    ]
}
EOF

aws iam put-role-policy --role-name batch_operations-objectlock \
--policy-name object-lock-permissions \
--policy-document "${batch_operations_permissions}"
```

Menggunakan AWS SDK for Java

Contoh berikut menunjukkan cara membuat peran IAM dengan izin Operasi Batch S3, dan memperbarui izin peran untuk membuat pekerjaan yang mengaktifkan Kunci Objek dengan menggunakan. AWS SDK untuk Java Anda juga harus memiliki manifes CSV yang mengidentifikasi objek untuk pekerjaan Operasi Batch S3 Anda. Untuk informasi selengkapnya, lihat the section called “Menentukan manifes”.

Lakukan langkah-langkah berikut ini:

1. Membuat peran IAM dan menetapkan izin Operasi Batch S3 yang akan dijalankan. Langkah ini diperlukan untuk semua pekerjaan Operasi Batch S3.
2. Menetapkan Operasi Batch S3 dengan Kunci Objek S3 yang akan dijalankan.

   Izinkan peran tersebut untuk melakukan hal-hal berikut ini:

1. Menjalankan Kunci Objek pada bucket S3 yang berisi objek target yang akan dijalankan oleh Operasi Batch.
2. Membaca bucket S3 tempat file CSV manifes dan objeknya berada.
3. Menulis hasil pekerjaan Operasi Batch S3 ke bucket pelaporan.

```
public void createObjectLockRole() {
    final String roleName = "batch_operations-object-lock";

    final String trustPolicy = "{" +
            "  \"Version\": \"2012-10-17\", " +
            "  \"Statement\": [ " +
            "    { " +
            "      \"Effect\": \"Allow\", " +
            "      \"Principal\": { " +
            "        \"Service\": [" +
            "          \"batchoperations.s3.amazonaws.com\"" +
            "        ]" +
            "      }, " +
            "      \"Action\": \"sts:AssumeRole\" " +
            "    } " +
            "  ]" +
            "}";

    final String bopsPermissions = "{" +
            "    \"Version\": \"2012-10-17\"," +
            "    \"Statement\": [" +
            "        {" +
            "            \"Effect\": \"Allow\"," +
            "            \"Action\": \"s3:GetBucketObjectLockConfiguration\"," +
            "            \"Resource\": [" +
            "                \"arn:aws:s3:::amzn-s3-demo-manifest-bucket\"" +
            "            ]" +
            "        }," +
            "        {" +
            "            \"Effect\": \"Allow\"," +
            "            \"Action\": [" +
            "                \"s3:GetObject\"," +
            "                \"s3:GetObjectVersion\"," +
            "                \"s3:GetBucketLocation\"" +
            "            ]," +
            "            \"Resource\": [" +
```

```
            "                \"arn:aws:s3:::amzn-s3-demo-manifest-bucket/*\"" +
            "            ]" +
            "        }," +
            "        {" +
            "            \"Effect\": \"Allow\"," +
            "            \"Action\": [" +
            "                \"s3:PutObject\"," +
            "                \"s3:GetBucketLocation\"" +
            "            ]," +
            "            \"Resource\": [" +
            "                \"arn:aws:s3:::amzn-s3-demo-completion-report-bucket/*\""
 +
            "            ]" +
            "        }" +
            "    ]" +
            "}";

    final AmazonIdentityManagement iam =
            AmazonIdentityManagementClientBuilder.defaultClient();

    final CreateRoleRequest createRoleRequest = new CreateRoleRequest()
            .withAssumeRolePolicyDocument(bopsPermissions)
            .withRoleName(roleName);

    final CreateRoleResult createRoleResult = iam.createRole(createRoleRequest);

    final PutRolePolicyRequest putRolePolicyRequest = new PutRolePolicyRequest()
            .withPolicyDocument(bopsPermissions)
            .withPolicyName("batch_operations-permissions")
            .withRoleName(roleName);

    final PutRolePolicyResult putRolePolicyResult =
 iam.putRolePolicy(putRolePolicyRequest);
}
```

Mengatur retensi Kunci Objek menggunakan Operasi Batch

Anda dapat menggunakan Operasi Batch Amazon S3 dengan Kunci Objek S3 untuk mengelola retensi untuk banyak objek Amazon S3 sekaligus. Anda menetapkan daftar objek target dalam manifest Anda dan mengirimkannya ke Batch Operations untuk diselesaikan. Untuk informasi selengkapnya, lihat the section called “Retensi Kunci Objek” dan the section called “Penyimpanan yang sah Kunci Objek”.

Contoh berikut menunjukkan cara membuat peran AWS Identity and Access Management (IAM) dengan izin Operasi Batch S3 dan memperbarui izin peran untuk menyertakan `s3:PutObjectRetention` izin sehingga Anda dapat menjalankan retensi Kunci Objek S3 pada objek di bucket manifes. Anda juga harus memiliki `CSV` manifes yang mengidentifikasi objek untuk pekerjaan Operasi Batch S3 Anda. Untuk informasi selengkapnya, lihat the section called "Menentukan manifes".

Untuk menggunakan contoh berikut, ganti *`user input placeholders`* dengan informasi Anda sendiri.

Menggunakan AWS CLI

AWS CLI Contoh berikut menunjukkan cara menggunakan Operasi Batch untuk menerapkan retensi Kunci Objek S3 di beberapa objek.

```
export AWS_PROFILE='aws-user'

read -d '' retention_permissions <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObjectRetention"
            ],
            "Resource": [
                "arn:aws:s3:::{{amzn-s3-demo-manifest-bucket}}/*"
            ]
        }
    ]
}
EOF

aws iam put-role-policy --role-name batch_operations-objectlock --policy-
name retention-permissions --policy-document "${retention_permissions}"
```

Menggunakan AWS SDK for Java

AWS SDK untuk Java Contoh berikut menunjukkan cara menggunakan Operasi Batch untuk menerapkan retensi Kunci Objek S3 di beberapa objek.

```
public void allowPutObjectRetention() {
    final String roleName = "batch_operations-object-lock";

    final String retentionPermissions = "{" +
            "    \"Version\": \"2012-10-17\"," +
            "    \"Statement\": [" +
            "        {" +
            "            \"Effect\": \"Allow\"," +
            "            \"Action\": [" +
            "                \"s3:PutObjectRetention\"" +
            "            ]," +
            "            \"Resource\": [" +
            "                \"arn:aws:s3:::amzn-s3-demo-manifest-bucket*\"" +
            "            ]" +
            "        }" +
            "    ]" +
            "}";

    final AmazonIdentityManagement iam =
            AmazonIdentityManagementClientBuilder.defaultClient();

    final PutRolePolicyRequest putRolePolicyRequest = new PutRolePolicyRequest()
            .withPolicyDocument(retentionPermissions)
            .withPolicyName("retention-permissions")
            .withRoleName(roleName);

    final PutRolePolicyResult putRolePolicyResult =
 iam.putRolePolicy(putRolePolicyRequest);
}
```

Menggunakan Operasi Batch S3 dengan mode kepatuhan retensi Kunci Objek S3

Contoh berikut dibangun berdasarkan contoh sebelumnya tentang pembuatan kebijakan kepercayaan dan pengaturan izin konfigurasi Operasi Batch S3 dan Kunci Objek S3 pada objek Anda. Contoh ini menetapkan mode retensi ke `COMPLIANCE` dan `retain until date` ke 1 Januari 2025. Contoh ini membuat job yang menargetkan objek dalam bucket manifes dan melaporkan hasilnya di bucket laporan yang Anda identifikasi.

Untuk menggunakan contoh berikut, ganti *`user input placeholders`* dengan informasi Anda sendiri.

Menggunakan AWS CLI

AWS CLI Contoh berikut menunjukkan cara menggunakan Operasi Batch untuk menerapkan mode kepatuhan retensi Kunci Objek S3 di beberapa objek.

Example - Atur mode kepatuhan retensi Kunci Objek S3 di beberapa objek

```
export AWS_PROFILE='aws-user'
export AWS_DEFAULT_REGION='us-west-2'
export ACCOUNT_ID=123456789012
export ROLE_ARN='arn:aws:iam::123456789012:role/batch_operations-objectlock'

read -d '' OPERATION <<EOF
{
  "S3PutObjectRetention": {
    "Retention": {
      "RetainUntilDate":"2025-01-01T00:00:00",
      "Mode":"COMPLIANCE"
    }
  }
}
EOF

read -d '' MANIFEST <<EOF
{
  "Spec": {
    "Format": "S3BatchOperations_CSV_20180820",
    "Fields": [
      "Bucket",
      "Key"
    ]
  },
  "Location": {
    "ObjectArn": "arn:aws:s3:::amzn-s3-demo-manifest-bucket/compliance-objects-
manifest.csv",
    "ETag": "Your-manifest-ETag"
  }
}
EOF

read -d '' REPORT <<EOF
{
  "Bucket": "arn:aws:s3:::ReportBucket",
  "Format": "Report_CSV_20180820",
```

```
  "Enabled": true,
  "Prefix": "amzn-s3-demo-completion-report-bucket/compliance-objects-batch-
operations",
  "ReportScope": "AllTasks"
}
EOF

aws \
    s3control create-job \
    --account-id "${ACCOUNT_ID}" \
    --manifest "${MANIFEST//$'\n'}" \
    --operation "${OPERATION//$'\n'/}" \
    --report "${REPORT//$'\n'}" \
    --priority 10 \
    --role-arn "${ROLE_ARN}" \
    --client-request-token "$(uuidgen)" \
    --region "${AWS_DEFAULT_REGION}" \
    --description "Set compliance retain-until to 1 Jul 2030";
```

Example — Perpanjang **`COMPLIANCE`** mode **`retain until date`** hingga 15 Januari 2025

Contoh berikut memperpanjang `retain until date` mode `COMPLIANCE` hingga 15 Januari 2025.

```
export AWS_PROFILE='aws-user'
export AWS_DEFAULT_REGION='us-west-2'
export ACCOUNT_ID=123456789012
export ROLE_ARN='arn:aws:iam::123456789012:role/batch_operations-objectlock'

read -d '' OPERATION <<EOF
{
  "S3PutObjectRetention": {
    "Retention": {
      "RetainUntilDate":"2025-01-15T00:00:00",
      "Mode":"COMPLIANCE"
    }
  }
}
EOF

read -d '' MANIFEST <<EOF
{
  "Spec": {
    "Format": "S3BatchOperations_CSV_20180820",
    "Fields": [
```

```
      "Bucket",
      "Key"
    ]
  },
  "Location": {
    "ObjectArn": "arn:aws:s3:::amzn-s3-demo-manifest-bucket/compliance-objects-
manifest.csv",
    "ETag": "Your-manifest-ETag"
  }
}
EOF

read -d '' REPORT <<EOF
{
  "Bucket": "arn:aws:s3:::amzn-s3-demo-completion-report-bucket",
  "Format": "Report_CSV_20180820",
  "Enabled": true,
  "Prefix": "reports/compliance-objects-batch_operations",
  "ReportScope": "AllTasks"
}
EOF

aws \
    s3control create-job \
    --account-id "${ACCOUNT_ID}" \
    --manifest "${MANIFEST//$'\n'}" \
    --operation "${OPERATION//$'\n'/}" \
    --report "${REPORT//$'\n'}" \
    --priority 10 \
    --role-arn "${ROLE_ARN}" \
    --client-request-token "$(uuidgen)" \
    --region "${AWS_DEFAULT_REGION}" \
    --description "Extend compliance retention to 15 Jan 2025";
```

Menggunakan AWS SDK for Java

AWS SDK untuk Java Contoh berikut menunjukkan cara menggunakan Operasi Batch untuk menerapkan mode kepatuhan retensi Kunci Objek S3 di beberapa objek.

Example — Atur mode retensi ke KEPATUHAN dan simpan hingga tanggal 1 Januari 2025

```
public String createComplianceRetentionJob(final AWSS3ControlClient awss3ControlClient)
 throws ParseException {
```

```
    final String manifestObjectArn = "arn:aws:s3:::amzn-s3-demo-manifest-bucket/
compliance-objects-manifest.csv";
    final String manifestObjectVersionId = "your-object-version-Id";

    final JobManifestLocation manifestLocation = new JobManifestLocation()
            .withObjectArn(manifestObjectArn)
            .withETag(manifestObjectVersionId);

    final JobManifestSpec manifestSpec =
            new JobManifestSpec()
                    .withFormat(JobManifestFormat.S3BatchOperations_CSV_20180820)
                    .withFields("Bucket", "Key");

    final JobManifest manifestToPublicApi = new JobManifest()
            .withLocation(manifestLocation)
            .withSpec(manifestSpec);

    final String jobReportBucketArn = "arn:aws:s3:::amzn-s3-demo-completion-report-
bucket";
    final String jobReportPrefix = "reports/compliance-objects-bops";

    final JobReport jobReport = new JobReport()
            .withEnabled(true)
            .withReportScope(JobReportScope.AllTasks)
            .withBucket(jobReportBucketArn)
            .withPrefix(jobReportPrefix)
            .withFormat(JobReportFormat.Report_CSV_20180820);

    final SimpleDateFormat format = new SimpleDateFormat("dd/MM/yyyy");
    final Date janFirst = format.parse("01/01/2025");

    final JobOperation jobOperation = new JobOperation()
            .withS3PutObjectRetention(new S3SetObjectRetentionOperation()
                    .withRetention(new S3Retention()
                            .withMode(S3ObjectLockRetentionMode.COMPLIANCE)
                            .withRetainUntilDate(janFirst)));

    final String roleArn = "arn:aws:iam::123456789012:role/batch_operations-object-
lock";
    final Boolean requiresConfirmation = true;
    final int priority = 10;

    final CreateJobRequest request = new CreateJobRequest()
            .withAccountId("123456789012")
```

```
            .withDescription("Set compliance retain-until to 1 Jan 2025")
            .withManifest(manifestToPublicApi)
            .withOperation(jobOperation)
            .withPriority(priority)
            .withRoleArn(roleArn)
            .withReport(jobReport)
            .withConfirmationRequired(requiresConfirmation);

    final CreateJobResult result = awss3ControlClient.createJob(request);

    return result.getJobId();
}
```

Example — Memperluas mode **`COMPLIANCEretain until date`**

Contoh berikut memperluas `COMPLIANCE` mode `retain until date` hingga 15 Januari 2025.

```
public String createExtendComplianceRetentionJob(final AWSS3ControlClient
 awss3ControlClient) throws ParseException {
    final String manifestObjectArn = "arn:aws:s3:::amzn-s3-demo-manifest-bucket/
compliance-objects-manifest.csv";
    final String manifestObjectVersionId = "15ad5ba069e6bbc465c77bf83d541385";

    final JobManifestLocation manifestLocation = new JobManifestLocation()
            .withObjectArn(manifestObjectArn)
            .withETag(manifestObjectVersionId);

    final JobManifestSpec manifestSpec =
            new JobManifestSpec()
                    .withFormat(JobManifestFormat.S3BatchOperations_CSV_20180820)
                    .withFields("Bucket", "Key");

    final JobManifest manifestToPublicApi = new JobManifest()
            .withLocation(manifestLocation)
            .withSpec(manifestSpec);

    final String jobReportBucketArn = "arn:aws:s3:::amzn-s3-demo-completion-report-
bucket";
    final String jobReportPrefix = "reports/compliance-objects-batch_operations";

    final JobReport jobReport = new JobReport()
            .withEnabled(true)
            .withReportScope(JobReportScope.AllTasks)
            .withBucket(jobReportBucketArn)
```

```
            .withPrefix(jobReportPrefix)
            .withFormat(JobReportFormat.Report_CSV_20180820);

    final SimpleDateFormat format = new SimpleDateFormat("dd/MM/yyyy");
    final Date jan15th = format.parse("15/01/2025");

    final JobOperation jobOperation = new JobOperation()
            .withS3PutObjectRetention(new S3SetObjectRetentionOperation()
                    .withRetention(new S3Retention()
                            .withMode(S3ObjectLockRetentionMode.COMPLIANCE)
                            .withRetainUntilDate(jan15th)));

    final String roleArn = "arn:aws:iam::123456789012:role/batch_operations-object-
lock";
    final Boolean requiresConfirmation = true;
    final int priority = 10;

    final CreateJobRequest request = new CreateJobRequest()
            .withAccountId("123456789012")
            .withDescription("Extend compliance retention to 15 Jan 2025")
            .withManifest(manifestToPublicApi)
            .withOperation(jobOperation)
            .withPriority(priority)
            .withRoleArn(roleArn)
            .withReport(jobReport)
            .withConfirmationRequired(requiresConfirmation);

    final CreateJobResult result = awss3ControlClient.createJob(request);

    return result.getJobId();
}
```

Gunakan Operasi Batch S3 dengan mode tata kelola penyimpanan Kunci Objek S3

Contoh berikut dibangun berdasarkan contoh sebelumnya tentang pembuatan kebijakan kepercayaan, dan pengaturan izin konfigurasi Operasi Batch S3 serta Kunci Objek S3. Contoh ini menunjukkan cara menerapkan tata kelola retensi Kunci Objek S3 dengan tanggal 30 Januari 2025, di `retain until date` beberapa objek. Contoh ini membuat pekerjaan Operasi Batch yang menggunakan bucket manifes dan melaporkan hasilnya di bucket laporan.

Untuk menggunakan contoh berikut, ganti *`user input placeholders`* dengan informasi Anda sendiri.

Menggunakan AWS CLI

AWS CLI Contoh berikut menunjukkan cara menggunakan Operasi Batch untuk menerapkan mode tata kelola retensi Kunci Objek S3 di beberapa objek.

Example — Terapkan tata kelola retensi Kunci Objek S3 di beberapa objek dengan penyimpanan hingga tanggal 30 Januari 2025

```
export AWS_PROFILE='aws-user'
export AWS_DEFAULT_REGION='us-west-2'
export ACCOUNT_ID=123456789012
export ROLE_ARN='arn:aws:iam::123456789012:role/batch_operations-objectlock'

read -d '' OPERATION <<EOF
{
  "S3PutObjectRetention": {
    "Retention": {
      "RetainUntilDate":"2025-01-30T00:00:00",
      "Mode":"GOVERNANCE"
    }
  }
}
EOF

read -d '' MANIFEST <<EOF
{
  "Spec": {
    "Format": "S3BatchOperations_CSV_20180820",
    "Fields": [
      "Bucket",
      "Key"
    ]
  },
  "Location": {
    "ObjectArn": "arn:aws:s3:::amzn-s3-demo-manifest-bucket/governance-objects-
manifest.csv",
    "ETag": "Your-manifest-ETag"
  }
}
EOF

read -d '' REPORT <<EOF
{
```

```
  "Bucket": "arn:aws:s3:::amzn-s3-demo-completion-report-bucketT",
  "Format": "Report_CSV_20180820",
  "Enabled": true,
  "Prefix": "reports/governance-objects",
  "ReportScope": "AllTasks"
}
EOF

aws \
    s3control create-job \
    --account-id "${ACCOUNT_ID}" \
    --manifest "${MANIFEST//$'\n'}" \
    --operation "${OPERATION//$'\n'/}" \
    --report "${REPORT//$'\n'}" \
    --priority 10 \
    --role-arn "${ROLE_ARN}" \
    --client-request-token "$(uuidgen)" \
    --region "${AWS_DEFAULT_REGION}" \
    --description "Put governance retention";
```

Example — Lewati tata kelola retensi di beberapa objek

Contoh berikut dibangun berdasarkan contoh sebelumnya tentang pembuatan kebijakan kepercayaan, dan pengaturan izin konfigurasi Operasi Batch S3 serta Kunci Objek S3. Contoh ini menunjukkan cara memintas tata kelola retensi pada beberapa objek dan membuat pekerjaan Operasi Batch yang menggunakan bucket manifes dan melaporkan hasil di bucket laporan.

```
export AWS_PROFILE='aws-user'

read -d '' bypass_governance_permissions <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:BypassGovernanceRetention"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
            ]
        }
    ]
```

```
}
EOF

aws iam put-role-policy --role-name batch-operations-objectlock --policy-name bypass-
governance-permissions --policy-document "${bypass_governance_permissions}"

export AWS_PROFILE='aws-user'
export AWS_DEFAULT_REGION='us-west-2'
export ACCOUNT_ID=123456789012
export ROLE_ARN='arn:aws:iam::123456789012:role/batch_operations-objectlock'

read -d '' OPERATION <<EOF
{
  "S3PutObjectRetention": {
    "BypassGovernanceRetention": true,
    "Retention": {
    }
  }
}
EOF

read -d '' MANIFEST <<EOF
{
  "Spec": {
    "Format": "S3BatchOperations_CSV_20180820",
    "Fields": [
      "Bucket",
      "Key"
    ]
  },
  "Location": {
    "ObjectArn": "arn:aws:s3:::amzn-s3-demo-manifest-bucket/governance-objects-
manifest.csv",
    "ETag": "Your-manifest-ETag"
  }
}
EOF

read -d '' REPORT <<EOF
{
  "Bucket": "arn:aws:s3:::amzn-s3-demo-completion-report-bucket",
  "Format": "Report_CSV_20180820",
  "Enabled": true,
  "Prefix": "reports/batch_operations-governance",
```

```
  "ReportScope": "AllTasks"
}
EOF

aws \
    s3control create-job \
    --account-id "${ACCOUNT_ID}" \
    --manifest "${MANIFEST//$'\n'}" \
    --operation "${OPERATION//$'\n'/}" \
    --report "${REPORT//$'\n'}" \
    --priority 10 \
    --role-arn "${ROLE_ARN}" \
    --client-request-token "$(uuidgen)" \
    --region "${AWS_DEFAULT_REGION}" \
    --description "Remove governance retention";
```

Menggunakan AWS SDK for Java

Contoh berikut dibangun berdasarkan contoh sebelumnya tentang pembuatan kebijakan kepercayaan, dan pengaturan izin konfigurasi S3 Batch Operations dan S3 Object Lock. Contoh ini menunjukkan cara menerapkan tata kelola retensi Kunci Objek S3 dengan `retain until date` set ke 30 Januari 2025 di beberapa objek. Contoh ini membuat pekerjaan Operasi Batch yang menggunakan bucket manifes dan melaporkan hasilnya di bucket laporan.

Example — Terapkan tata kelola retensi Kunci Objek S3 di beberapa objek dengan penyimpanan hingga tanggal 30 Januari 2025

```
public String createGovernanceRetentionJob(final AWSS3ControlClient awss3ControlClient)
 throws ParseException {
    final String manifestObjectArn = "arn:aws:s3:::amzn-s3-demo-manifest-bucket/
governance-objects-manifest.csv";
    final String manifestObjectVersionId = "15ad5ba069e6bbc465c77bf83d541385";

    final JobManifestLocation manifestLocation = new JobManifestLocation()
            .withObjectArn(manifestObjectArn)
            .withETag(manifestObjectVersionId);

    final JobManifestSpec manifestSpec =
            new JobManifestSpec()
                    .withFormat(JobManifestFormat.S3BatchOperations_CSV_20180820)
                    .withFields("Bucket", "Key");

    final JobManifest manifestToPublicApi = new JobManifest()
```

```
            .withLocation(manifestLocation)
            .withSpec(manifestSpec);

    final String jobReportBucketArn = "arn:aws:s3:::amzn-s3-demo-completion-report-
bucket";
    final String jobReportPrefix = "reports/governance-objects";

    final JobReport jobReport = new JobReport()
            .withEnabled(true)
            .withReportScope(JobReportScope.AllTasks)
            .withBucket(jobReportBucketArn)
            .withPrefix(jobReportPrefix)
            .withFormat(JobReportFormat.Report_CSV_20180820);

    final SimpleDateFormat format = new SimpleDateFormat("dd/MM/yyyy");
    final Date jan30th = format.parse("30/01/2025");

    final JobOperation jobOperation = new JobOperation()
            .withS3PutObjectRetention(new S3SetObjectRetentionOperation()
                    .withRetention(new S3Retention()
                            .withMode(S3ObjectLockRetentionMode.GOVERNANCE)
                            .withRetainUntilDate(jan30th)));

    final String roleArn = "arn:aws:iam::123456789012:role/batch_operations-object-
lock";
    final Boolean requiresConfirmation = true;
    final int priority = 10;

    final CreateJobRequest request = new CreateJobRequest()
            .withAccountId("123456789012")
            .withDescription("Put governance retention")
            .withManifest(manifestToPublicApi)
            .withOperation(jobOperation)
            .withPriority(priority)
            .withRoleArn(roleArn)
            .withReport(jobReport)
            .withConfirmationRequired(requiresConfirmation);

    final CreateJobResult result = awss3ControlClient.createJob(request);

    return result.getJobId();
}
```

Example — Lewati tata kelola retensi di beberapa objek

Contoh berikut dibangun berdasarkan contoh sebelumnya tentang pembuatan kebijakan kepercayaan, dan pengaturan izin konfigurasi Operasi Batch S3 serta Kunci Objek S3. Contoh ini menunjukkan cara mem-bypass tata kelola retensi di beberapa objek dan membuat pekerjaan Operasi Batch yang menggunakan bucket manifes dan melaporkan hasilnya di bucket laporan.

```
public void allowBypassGovernance() {
    final String roleName = "batch_operations-object-lock";

    final String bypassGovernancePermissions = "{" +
            "    \"Version\": \"2012-10-17\"," +
            "    \"Statement\": [" +
            "        {" +
            "            \"Effect\": \"Allow\"," +
            "            \"Action\": [" +
            "                \"s3:BypassGovernanceRetention\"" +
            "            ]," +
            "            \"Resource\": [" +
            "                \"arn:aws:s3:::amzn-s3-demo-manifest-bucket/*\"" +
            "            ]" +
            "        }" +
            "    ]" +
            "}";

    final AmazonIdentityManagement iam =
            AmazonIdentityManagementClientBuilder.defaultClient();

    final PutRolePolicyRequest putRolePolicyRequest = new PutRolePolicyRequest()
            .withPolicyDocument(bypassGovernancePermissions)
            .withPolicyName("bypass-governance-permissions")
            .withRoleName(roleName);

    final PutRolePolicyResult putRolePolicyResult =
 iam.putRolePolicy(putRolePolicyRequest);
}
public String createRemoveGovernanceRetentionJob(final AWSS3ControlClient
 awss3ControlClient) {
    final String manifestObjectArn = "arn:aws:s3:::amzn-s3-demo-manifest-bucket/
governance-objects-manifest.csv";
    final String manifestObjectVersionId = "15ad5ba069e6bbc465c77bf83d541385";

    final JobManifestLocation manifestLocation = new JobManifestLocation()
```

```
            .withObjectArn(manifestObjectArn)
            .withETag(manifestObjectVersionId);

    final JobManifestSpec manifestSpec =
            new JobManifestSpec()
                    .withFormat(JobManifestFormat.S3BatchOperations_CSV_20180820)
                    .withFields("Bucket", "Key");

    final JobManifest manifestToPublicApi = new JobManifest()
            .withLocation(manifestLocation)
            .withSpec(manifestSpec);

    final String jobReportBucketArn = "arn:aws:s3:::amzn-s3-demo-completion-report-
bucket";
    final String jobReportPrefix = "reports/batch_operations-governance";

    final JobReport jobReport = new JobReport()
            .withEnabled(true)
            .withReportScope(JobReportScope.AllTasks)
            .withBucket(jobReportBucketArn)
            .withPrefix(jobReportPrefix)
            .withFormat(JobReportFormat.Report_CSV_20180820);

    final JobOperation jobOperation = new JobOperation()
            .withS3PutObjectRetention(new S3SetObjectRetentionOperation()
                    .withRetention(new S3Retention()));

    final String roleArn = "arn:aws:iam::123456789012:role/batch_operations-object-
lock";
    final Boolean requiresConfirmation = true;
    final int priority = 10;

    final CreateJobRequest request = new CreateJobRequest()
            .withAccountId("123456789012")
            .withDescription("Remove governance retention")
            .withManifest(manifestToPublicApi)
            .withOperation(jobOperation)
            .withPriority(priority)
            .withRoleArn(roleArn)
            .withReport(jobReport)
            .withConfirmationRequired(requiresConfirmation);

    final CreateJobResult result = awss3ControlClient.createJob(request);
```

```
    return result.getJobId();
}
```

Menggunakan Operasi Batch S3 untuk mematikan penahanan hukum Kunci Objek S3

Contoh berikut dibangun berdasarkan contoh sebelumnya tentang pembuatan kebijakan kepercayaan, dan pengaturan izin konfigurasi Operasi Batch S3 serta Kunci Objek S3. Contoh ini menunjukkan cara menonaktifkan penahanan hukum Object Lock pada objek menggunakan Operasi Batch.

Contoh ini pertama-tama akan memperbarui peran untuk memberikan izin `s3:PutObjectLegalHold`, membuat pekerjaan Operasi Batch yang menonaktifkan (menghapus) penahanan hukum dari objek yang diidentifikasi dalam manifes, kemudian melaporkan hal tersebut.

Untuk menggunakan contoh berikut, ganti *`user input placeholders`* dengan informasi Anda sendiri.

Menggunakan AWS CLI

AWS CLI Contoh berikut menunjukkan cara menggunakan Operasi Batch untuk mematikan penahanan legal Kunci Objek S3 di beberapa objek.

Example — Memperbarui peran untuk memberikan **`s3:PutObjectLegalHold`** izin

```
export AWS_PROFILE='aws-user'

read -d '' legal_hold_permissions <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObjectLegalHold"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
            ]
        }
    ]

EOF
```

```
aws iam put-role-policy --role-name batch_operations-objectlock --policy-name legal-
hold-permissions --policy-document "${legal_hold_permissions}"
```

Example — Matikan penahanan hukum

Contoh berikut menunjukkan cara menonaktifkan penahanan hukum.

```
export AWS_PROFILE='aws-user'
export AWS_DEFAULT_REGION='us-west-2'
export ACCOUNT_ID=123456789012
export ROLE_ARN='arn:aws:iam::123456789012:role/batch_operations-objectlock'

read -d '' OPERATION <<EOF
{
  "S3PutObjectLegalHold": {
    "LegalHold": {
      "Status":"OFF"
    }
  }
}
EOF

read -d '' MANIFEST <<EOF
{
  "Spec": {
    "Format": "S3BatchOperations_CSV_20180820",
    "Fields": [
      "Bucket",
      "Key"
    ]
  },
  "Location": {
    "ObjectArn": "arn:aws:s3:::amzn-s3-demo-manifest-bucket/legalhold-object-
manifest.csv",
    "ETag": "Your-manifest-ETag"
  }
}
EOF

read -d '' REPORT <<EOF
{
  "Bucket": "arn:aws:s3:::amzn-s3-demo-completion-report-bucket",
  "Format": "Report_CSV_20180820",
```

```
  "Enabled": true,
  "Prefix": "reports/legalhold-objects-batch_operations",
  "ReportScope": "AllTasks"
}
EOF

aws \
    s3control create-job \
    --account-id "${ACCOUNT_ID}" \
    --manifest "${MANIFEST//$'\n'}" \
    --operation "${OPERATION//$'\n'/}" \
    --report "${REPORT//$'\n'}" \
    --priority 10 \
    --role-arn "${ROLE_ARN}" \
    --client-request-token "$(uuidgen)" \
    --region "${AWS_DEFAULT_REGION}" \
    --description "Turn off legal hold";
```

Menggunakan AWS SDK for Java

AWS SDK untuk Java Contoh berikut menunjukkan cara menggunakan Operasi Batch untuk mematikan penahanan legal Kunci Objek S3 di beberapa objek.

Example — Memperbarui peran untuk memberikan **`s3:PutObjectLegalHold`** izin

```
public void allowPutObjectLegalHold() {
    final String roleName = "batch_operations-object-lock";

    final String legalHoldPermissions = "{" +
            "    \"Version\": \"2012-10-17\"," +
            "    \"Statement\": [" +
            "        {" +
            "            \"Effect\": \"Allow\"," +
            "            \"Action\": [" +
            "                \"s3:PutObjectLegalHold\"" +
            "            ]," +
            "            \"Resource\": [" +
            "                \"arn:aws:s3:::amzn-s3-demo-manifest-bucket/*\"" +
            "            ]" +
            "        }" +
            "    ]" +
            "}";
```

```
    final AmazonIdentityManagement iam =
            AmazonIdentityManagementClientBuilder.defaultClient();

    final PutRolePolicyRequest putRolePolicyRequest = new PutRolePolicyRequest()
            .withPolicyDocument(legalHoldPermissions)
            .withPolicyName("legal-hold-permissions")
            .withRoleName(roleName);

    final PutRolePolicyResult putRolePolicyResult =
 iam.putRolePolicy(putRolePolicyRequest);
}
```

Example — Matikan penahanan hukum

Gunakan contoh di bawah ini jika Anda ingin menonaktifkan penahanan hukum.

```
public String createLegalHoldOffJob(final AWSS3ControlClient awss3ControlClient) {
    final String manifestObjectArn = "arn:aws:s3:::amzn-s3-demo-manifest-bucket/
legalhold-object-manifest.csv";
    final String manifestObjectVersionId = "15ad5ba069e6bbc465c77bf83d541385";

    final JobManifestLocation manifestLocation = new JobManifestLocation()
            .withObjectArn(manifestObjectArn)
            .withETag(manifestObjectVersionId);

    final JobManifestSpec manifestSpec =
            new JobManifestSpec()
                    .withFormat(JobManifestFormat.S3BatchOperations_CSV_20180820)
                    .withFields("Bucket", "Key");

    final JobManifest manifestToPublicApi = new JobManifest()
            .withLocation(manifestLocation)
            .withSpec(manifestSpec);

    final String jobReportBucketArn = "arn:aws:s3:::amzn-s3-demo-completion-report-
bucket";
    final String jobReportPrefix = "reports/legalhold-objects-batch_operations";

    final JobReport jobReport = new JobReport()
            .withEnabled(true)
            .withReportScope(JobReportScope.AllTasks)
            .withBucket(jobReportBucketArn)
            .withPrefix(jobReportPrefix)
            .withFormat(JobReportFormat.Report_CSV_20180820);
```

```
    final JobOperation jobOperation = new JobOperation()
            .withS3PutObjectLegalHold(new S3SetObjectLegalHoldOperation()
                    .withLegalHold(new S3ObjectLockLegalHold()
                            .withStatus(S3ObjectLockLegalHoldStatus.OFF)));

    final String roleArn = "arn:aws:iam::123456789012:role/batch_operations-object-
lock";
    final Boolean requiresConfirmation = true;
    final int priority = 10;

    final CreateJobRequest request = new CreateJobRequest()
            .withAccountId("123456789012")
            .withDescription("Turn off legal hold")
            .withManifest(manifestToPublicApi)
            .withOperation(jobOperation)
            .withPriority(priority)
            .withRoleArn(roleArn)
            .withReport(jobReport)
            .withConfirmationRequired(requiresConfirmation);

    final CreateJobResult result = awss3ControlClient.createJob(request);

    return result.getJobId();
}
```

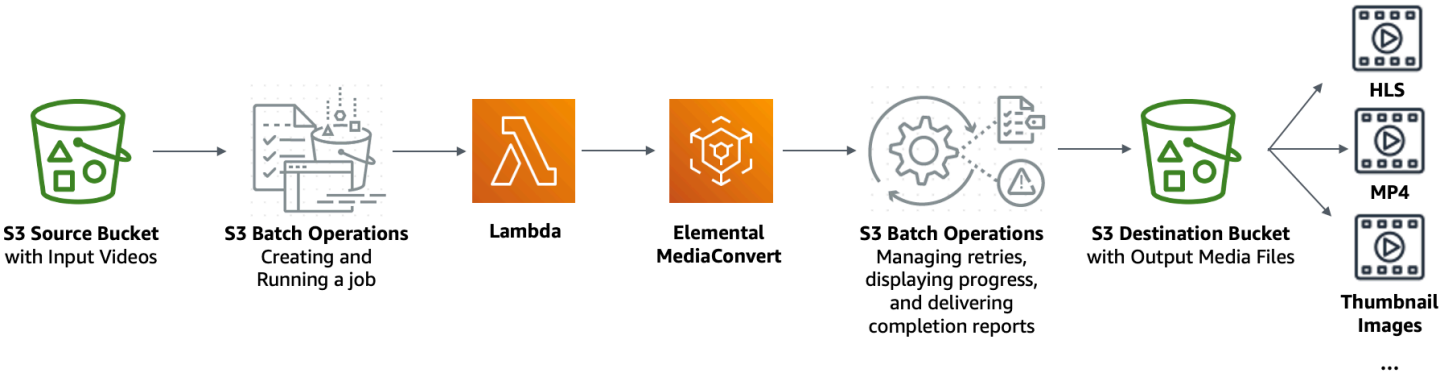
Tutorial: Video transcoding batch dengan Operasi Batch S3

Konsumen video menggunakan perangkat dari segala bentuk, ukuran, dan vintage untuk menikmati konten media. Beragam perangkat ini menghadirkan tantangan bagi pembuat konten dan distributor. Alih-alih dalam one-size-fits-all format, video harus dikonversi sehingga dapat menjangkau berbagai ukuran, format, dan bitrate. Tugas konversi ini bahkan lebih menantang ketika Anda memiliki sejumlah besar video yang harus dikonversi.

AWS menawarkan Anda metode untuk membangun arsitektur terdistribusi yang dapat diskalakan yang melakukan hal berikut:

- Menelan video masukan
- Memproses video untuk pemutaran di berbagai perangkat
- Menyimpan file media yang ditranskode
- Mengirimkan file media keluaran untuk memenuhi permintaan

Jika Anda memiliki repositori video ekstensif yang disimpan di Amazon S3, Anda dapat mentranskode video ini dari format sumbernya ke beberapa jenis file dalam ukuran, resolusi, dan format yang diperlukan oleh pemutar video atau perangkat tertentu. Secara khusus, Operasi Batch S3 memberi Anda solusi untuk menjalankan AWS Lambda fungsi untuk video input yang ada di bucket sumber S3. Kemudian, fungsi Lambda memanggil AWS Elemental MediaConvertuntuk melakukan tugas transcoding video skala besar. File media keluaran yang dikonversi disimpan dalam bucket tujuan S3.



Tujuan

Dalam tutorial ini, Anda mempelajari cara mengatur Operasi Batch S3 untuk menjalankan fungsi Lambda untuk transcoding batch video yang disimpan dalam bucket sumber S3. Fungsi Lambda memanggil MediaConvert untuk mentranskode video. Output untuk setiap video di bucket sumber S3 adalah sebagai berikut:

- Streaming bitrate adaptif HTTP Live Streaming (HLS) untuk pemutaran pada perangkat dengan berbagai ukuran dan bandwidth yang bervariasi
- File MP4 video
- Gambar thumbnail dikumpulkan pada interval

Topik

- Prasyarat
- Langkah 1: Buat bucket S3 untuk file media keluaran
- Langkah 2: Buat peran IAM untuk MediaConvert
- Langkah 3: Buat peran IAM untuk fungsi Lambda Anda
- Langkah 4: Buat fungsi Lambda untuk transcoding video
- Langkah 5: Konfigurasikan Inventaris Amazon S3 untuk bucket sumber S3 Anda

- Langkah 6: Buat peran IAM untuk Operasi Batch S3
- Langkah 7: Buat dan jalankan pekerjaan Operasi Batch S3
- Langkah 8: Memeriksa file media output dari bucket tujuan S3 Anda
- Langkah 9: Membersihkan
- Langkah selanjutnya

Prasyarat

Sebelum Anda memulai tutorial ini, Anda harus memiliki bucket sumber Amazon S3 (misalnya,*amzn-s3-demo-source-bucket*) dengan video yang akan ditranskode sudah disimpan di dalamnya.

Anda dapat memberi ember nama lain jika Anda mau. Untuk informasi selengkapnya tentang nama bucket di Amazon S3, lihat. Aturan penamaan bucket tujuan umum

Untuk bucket sumber S3, pertahankan pengaturan yang terkait dengan pengaturan Blokir Akses Publik untuk bucket ini disetel ke default (Blokir semua akses publik diaktifkan). Untuk informasi selengkapnya, lihat Membuat ember tujuan umum.

Untuk informasi selengkapnya tentang mengunggah video ke bucket sumber S3, lihat. Mengunggah Objek Jika Anda mengunggah banyak file video besar ke S3, Anda mungkin ingin menggunakan Amazon S3 Transfer Acceleration untuk mengonfigurasi transfer file yang cepat dan aman. Transfer Acceleration dapat mempercepat pengunggahan video ke bucket S3 Anda untuk transfer jarak jauh video yang lebih besar. Untuk informasi selengkapnya, lihat Mengonfigurasi transfer file yang cepat dan aman menggunakan Amazon S3 Transfer Acceleration.

Langkah 1: Buat bucket S3 untuk file media keluaran

Pada langkah ini, Anda membuat bucket tujuan S3 untuk menyimpan file media keluaran yang dikonversi. Anda juga membuat konfigurasi Cross Origin Resource Sharing (CORS) untuk memungkinkan akses lintas asal ke file media transkode yang disimpan di bucket tujuan S3 Anda.

Sublangkah

- Buat ember untuk file media keluaran
- Tambahkan konfigurasi CORS ke bucket keluaran S3

Buat ember untuk file media keluaran

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Pilih Buat bucket.
4. Untuk nama Bucket, masukkan nama untuk bucket Anda (misalnya,*amzn-s3-demo-destination-bucket1*).
5. Untuk Wilayah, pilih Wilayah AWS tempat Anda ingin ember berada.
6. Untuk memastikan akses publik ke file media keluaran Anda, di setelan Blokir Akses Publik untuk bucket ini, hapus Blokir semua akses publik.

   > **Warning**
   >
   > Sebelum Anda menyelesaikan langkah ini, tinjau Melakukan blok akses publik ke penyimpanan Amazon S3 Anda untuk memastikan bahwa Anda telah memahami dan menerima risiko yang terkait dengan mengizinkan akses publik. Saat Anda menonaktifkan pengaturan Blokir Akses Publik untuk membuat bucket Anda menjadi publik, siapa pun di internet dapat mengakses bucket Anda. Kami sarankan agar Anda memblokir semua akses publik ke bucket Anda.
   > Jika Anda tidak ingin menghapus pengaturan Blokir Akses Publik, Anda dapat menggunakan Amazon CloudFront untuk mengirimkan file media yang ditranskode ke pemirsa (pengguna akhir). Untuk informasi selengkapnya, lihat Tutorial: Hosting video streaming sesuai permintaan dengan Amazon S3, Amazon, dan CloudFront Amazon Route 53.

7. Pilih kotak centang di sebelah Saya mengakui bahwa pengaturan saat ini dapat mengakibatkan bucket ini dan objek dalam menjadi publik.
8. Pertahankan pengaturan yang tersisa disetel ke default.
9. Pilih Buat bucket.

Tambahkan konfigurasi CORS ke bucket keluaran S3

Konfigurasi JSON CORS mendefinisikan cara untuk aplikasi web klien (pemutar video dalam konteks ini) yang dimuat dalam satu domain untuk memutar file media keluaran transkode dalam domain yang berbeda.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Dalam daftar Bucket, pilih nama bucket yang Anda buat sebelumnya (misalnya,*amzn-s3-demo-destination-bucket1*).
4. Pilih tab Izin.
5. Di bagian Berbagi sumber daya lintas asal (CORS), pilih Edit.
6. Di kotak teks konfigurasi CORS, salin dan tempel konfigurasi CORS berikut.

   Konfigurasi CORS harus dalam format JSON. Dalam contoh ini, `AllowedOrigins` atribut menggunakan karakter wildcard (*) untuk menentukan semua asal. Jika Anda tahu asal spesifik Anda, Anda dapat membatasi `AllowedOrigins` atribut ke URL pemain spesifik Anda. Untuk informasi selengkapnya tentang mengonfigurasi atribut ini dan atribut lainnya, lihatElemen konfigurasi CORS.

   ```
   [
       {
           "AllowedOrigins": [
               "*"
           ],
           "AllowedMethods": [
               "GET"
           ],
           "AllowedHeaders": [
               "*"
           ],
           "ExposeHeaders": []
   
       }
   ]
   ```

7. Pilih Simpan perubahan.

Langkah 2: Buat peran IAM untuk MediaConvert

Untuk digunakan AWS Elemental MediaConvert untuk mentranskode video input yang disimpan di bucket S3, Anda harus memiliki peran layanan AWS Identity and Access Management (IAM) untuk memberikan MediaConvert izin membaca dan menulis file video dari dan ke bucket sumber dan

tujuan S3 Anda. Saat Anda menjalankan pekerjaan transcoding, MediaConvert konsol menggunakan peran ini.

Untuk membuat peran IAM untuk MediaConvert

1. Buat peran IAM dengan nama peran yang Anda pilih (misalnya,**`tutorial-mediaconvert-role`**). Untuk membuat peran ini, ikuti langkah-langkah di Buat MediaConvert peran Anda di IAM (konsol) di Panduan AWS Elemental MediaConvert Pengguna.
2. Setelah Anda membuat peran IAM untuk MediaConvert, dalam daftar Peran, pilih nama peran MediaConvert yang Anda buat (misalnya,**`tutorial-mediaconvert-role`**).
3. Pada halaman Ringkasan, salin ARN Peran (yang dimulai dengan`arn:aws:iam::`), dan simpan ARN untuk digunakan nanti.

   Untuk informasi selengkapnya ARNs, lihat Amazon Resource Names (ARNs) di Referensi AWS Umum.

Langkah 3: Buat peran IAM untuk fungsi Lambda Anda

Untuk mentranskode video batch dengan dan Operasi Batch MediaConvert S3, Anda menggunakan fungsi Lambda untuk menghubungkan kedua layanan ini untuk mengonversi video. Fungsi Lambda ini harus memiliki peran IAM yang memberikan izin fungsi Lambda untuk MediaConvert mengakses dan Operasi Batch S3.

Sublangkah

- Buat peran IAM untuk fungsi Lambda Anda
- Sematkan kebijakan inline untuk peran IAM fungsi Lambda Anda

Buat peran IAM untuk fungsi Lambda Anda

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi kiri, pilih Peran, lalu pilih Buat peran.
3. Pilih jenis peran AWS layanan, lalu di bawah Kasus penggunaan umum, pilih Lambda.
4. Pilih Berikutnya: Izin.
5. Pada halaman Lampirkan kebijakan izin, masukkan **`AWSLambdaBasicExecutionRole`** kotak Filter kebijakan. Untuk melampirkan kebijakan terkelola AWSLambdaBasicExecutionRoleke

peran ini untuk memberikan izin menulis ke CloudWatch Log Amazon, pilih kotak centang di AWSLambdaBasicExecutionRolesebelahnya.

6. Pilih Berikutnya.
7. Untuk Nama peran, masukkan **`tutorial-lambda-transcode-role`**.
8. (Opsional) Tambahkan tag ke kebijakan terkelola.
9. Pilih Buat peran.

Sematkan kebijakan inline untuk peran IAM fungsi Lambda Anda

Untuk memberikan izin ke MediaConvert sumber daya yang diperlukan agar fungsi Lambda dapat dijalankan, Anda harus menggunakan kebijakan sebaris.

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi sebelah kiri, pilih Peran.
3. Dalam daftar Peran, pilih nama peran IAM yang Anda buat sebelumnya untuk fungsi Lambda Anda (misalnya**`tutorial-lambda-transcode-role`**,).
4. Pilih tab Izin.
5. Pilih Menambahkan kebijakan inline.
6. Pilih tab JSON, lalu salin dan tempel kebijakan JSON berikut.

   Dalam kebijakan JSON, ganti contoh nilai ARN dengan peran ARN `Resource` dari peran IAM MediaConvert untuk yang Anda buat di Langkah 2 (misalnya,). **`tutorial-mediaconvert-role`**

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "Logging"
```

```
        },
        {
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::111122223333:role/tutorial-mediaconvert-role"
            ],
            "Effect": "Allow",
            "Sid": "PassRole"
        },
        {
            "Action": [
                "mediaconvert:*"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "MediaConvertService"
        },
        {
            "Action": [
                "s3:*"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "S3Service"
        }
    ]
}
```

7. Pilih Tinjau Kebijakan.
8. Untuk Nama, masukkan **`tutorial-lambda-policy`**.
9. Pilih Buat Kebijakan.

   Setelah Anda membuat kebijakan sebaris, kebijakan tersebut secara otomatis disematkan dalam peran IAM fungsi Lambda Anda.

Langkah 4: Buat fungsi Lambda untuk transcoding video

Di bagian tutorial ini, Anda membangun fungsi Lambda menggunakan SDK untuk Python untuk diintegrasikan dengan Operasi Batch S3 dan. MediaConvert Untuk memulai transcoding video yang sudah disimpan di bucket sumber S3, Anda menjalankan tugas Operasi Batch S3 yang secara langsung memanggil fungsi Lambda untuk setiap video di bucket sumber S3. Kemudian, fungsi Lambda mengirimkan pekerjaan transcoding untuk setiap video. MediaConvert

Sublangkah

- Tulis kode fungsi Lambda dan buat paket penerapan
- Buat fungsi Lambda dengan peran eksekusi (konsol)
- Terapkan fungsi Lambda Anda dengan arsip file.zip dan konfigurasikan fungsi Lambda (konsol)

Tulis kode fungsi Lambda dan buat paket penerapan

1. Di mesin lokal Anda, buat folder bernama`batch-transcode`.
2. Di `batch-transcode` folder, buat file dengan pengaturan pekerjaan JSON. Misalnya, Anda dapat menggunakan pengaturan yang disediakan di bagian ini, dan beri nama file`job.json`.

   Sebuah `job.json` file menentukan berikut:

   - File mana yang akan ditranskode
   - Bagaimana Anda ingin mentranskode video input Anda
   - File media keluaran apa yang ingin Anda buat
   - Apa yang harus diberi nama file yang ditranskode
   - Tempat menyimpan file yang ditranskode
   - Fitur canggih mana yang akan diterapkan, dan sebagainya

   Dalam tutorial ini, kita menggunakan `job.json` file berikut untuk membuat output berikut untuk setiap video di bucket sumber S3:

   - Streaming bitrate adaptif HTTP Live Streaming (HLS) untuk pemutaran di beberapa perangkat dengan ukuran berbeda dan bandwidth yang bervariasi
   - File MP4 video
   - Gambar thumbnail dikumpulkan pada interval

`job.json`File contoh ini menggunakan Quality-Defined Variable Bitrate (QVBR) untuk mengoptimalkan kualitas video. Output HLS sesuai dengan Apple (audio tidak dicampur dari video, durasi segmen 6 detik, dan kualitas video yang dioptimalkan melalui QVBR otomatis).

Jika Anda tidak ingin menggunakan contoh pengaturan yang disediakan di sini, Anda dapat membuat `job.json` spesifikasi berdasarkan kasus penggunaan Anda. Untuk memastikan konsistensi di seluruh output Anda, pastikan file input Anda memiliki konfigurasi video dan audio yang serupa. Untuk file input apa pun dengan konfigurasi video dan audio yang berbeda, buat otomatisasi terpisah (`job.json`pengaturan unik). Untuk informasi selengkapnya, lihat Contoh pengaturan AWS Elemental MediaConvert pekerjaan di JSON di Panduan AWS Elemental MediaConvert Pengguna.

```
{
  "OutputGroups": [
    {
      "CustomName": "HLS",
      "Name": "Apple HLS",
      "Outputs": [
        {
          "ContainerSettings": {
            "Container": "M3U8",
            "M3u8Settings": {
              "AudioFramesPerPes": 4,
              "PcrControl": "PCR_EVERY_PES_PACKET",
              "PmtPid": 480,
              "PrivateMetadataPid": 503,
              "ProgramNumber": 1,
              "PatInterval": 0,
              "PmtInterval": 0,
              "TimedMetadata": "NONE",
              "VideoPid": 481,
              "AudioPids": [
                482,
                483,
                484,
                485,
                486,
                487,
                488,
                489,
```

```
            490,
            491,
            492
          ]
        }
      },
      "VideoDescription": {
        "Width": 640,
        "ScalingBehavior": "DEFAULT",
        "Height": 360,
        "TimecodeInsertion": "DISABLED",
        "AntiAlias": "ENABLED",
        "Sharpness": 50,
        "CodecSettings": {
          "Codec": "H_264",
          "H264Settings": {
            "InterlaceMode": "PROGRESSIVE",
            "NumberReferenceFrames": 3,
            "Syntax": "DEFAULT",
            "Softness": 0,
            "GopClosedCadence": 1,
            "GopSize": 2,
            "Slices": 1,
            "GopBReference": "DISABLED",
            "MaxBitrate": 1200000,
            "SlowPal": "DISABLED",
            "SpatialAdaptiveQuantization": "ENABLED",
            "TemporalAdaptiveQuantization": "ENABLED",
            "FlickerAdaptiveQuantization": "DISABLED",
            "EntropyEncoding": "CABAC",
            "FramerateControl": "INITIALIZE_FROM_SOURCE",
            "RateControlMode": "QVBR",
            "CodecProfile": "MAIN",
            "Telecine": "NONE",
            "MinIInterval": 0,
            "AdaptiveQuantization": "HIGH",
            "CodecLevel": "AUTO",
            "FieldEncoding": "PAFF",
            "SceneChangeDetect": "TRANSITION_DETECTION",
            "QualityTuningLevel": "SINGLE_PASS_HQ",
            "FramerateConversionAlgorithm": "DUPLICATE_DROP",
            "UnregisteredSeiTimecode": "DISABLED",
            "GopSizeUnits": "SECONDS",
            "ParControl": "INITIALIZE_FROM_SOURCE",
```

```
              "NumberBFramesBetweenReferenceFrames": 2,
              "RepeatPps": "DISABLED"
            }
          },
          "AfdSignaling": "NONE",
          "DropFrameTimecode": "ENABLED",
          "RespondToAfd": "NONE",
          "ColorMetadata": "INSERT"
        },
        "OutputSettings": {
          "HlsSettings": {
            "AudioGroupId": "program_audio",
            "AudioRenditionSets": "program_audio",
            "SegmentModifier": "$dt$",
            "IFrameOnlyManifest": "EXCLUDE"
          }
        },
        "NameModifier": "_360"
      },
      {
        "ContainerSettings": {
          "Container": "M3U8",
          "M3u8Settings": {
            "AudioFramesPerPes": 4,
            "PcrControl": "PCR_EVERY_PES_PACKET",
            "PmtPid": 480,
            "PrivateMetadataPid": 503,
            "ProgramNumber": 1,
            "PatInterval": 0,
            "PmtInterval": 0,
            "TimedMetadata": "NONE",
            "TimedMetadataPid": 502,
            "VideoPid": 481,
            "AudioPids": [
              482,
              483,
              484,
              485,
              486,
              487,
              488,
              489,
              490,
              491,
```

```
            492
          ]
        }
      },
      "VideoDescription": {
        "Width": 960,
        "ScalingBehavior": "DEFAULT",
        "Height": 540,
        "TimecodeInsertion": "DISABLED",
        "AntiAlias": "ENABLED",
        "Sharpness": 50,
        "CodecSettings": {
          "Codec": "H_264",
          "H264Settings": {
            "InterlaceMode": "PROGRESSIVE",
            "NumberReferenceFrames": 3,
            "Syntax": "DEFAULT",
            "Softness": 0,
            "GopClosedCadence": 1,
            "GopSize": 2,
            "Slices": 1,
            "GopBReference": "DISABLED",
            "MaxBitrate": 3500000,
            "SlowPal": "DISABLED",
            "SpatialAdaptiveQuantization": "ENABLED",
            "TemporalAdaptiveQuantization": "ENABLED",
            "FlickerAdaptiveQuantization": "DISABLED",
            "EntropyEncoding": "CABAC",
            "FramerateControl": "INITIALIZE_FROM_SOURCE",
            "RateControlMode": "QVBR",
            "CodecProfile": "MAIN",
            "Telecine": "NONE",
            "MinIInterval": 0,
            "AdaptiveQuantization": "HIGH",
            "CodecLevel": "AUTO",
            "FieldEncoding": "PAFF",
            "SceneChangeDetect": "TRANSITION_DETECTION",
            "QualityTuningLevel": "SINGLE_PASS_HQ",
            "FramerateConversionAlgorithm": "DUPLICATE_DROP",
            "UnregisteredSeiTimecode": "DISABLED",
            "GopSizeUnits": "SECONDS",
            "ParControl": "INITIALIZE_FROM_SOURCE",
            "NumberBFramesBetweenReferenceFrames": 2,
            "RepeatPps": "DISABLED"
```

```
            }
          },
          "AfdSignaling": "NONE",
          "DropFrameTimecode": "ENABLED",
          "RespondToAfd": "NONE",
          "ColorMetadata": "INSERT"
        },
        "OutputSettings": {
          "HlsSettings": {
            "AudioGroupId": "program_audio",
            "AudioRenditionSets": "program_audio",
            "SegmentModifier": "$dt$",
            "IFrameOnlyManifest": "EXCLUDE"
          }
        },
        "NameModifier": "_540"
      },
      {
        "ContainerSettings": {
          "Container": "M3U8",
          "M3u8Settings": {
            "AudioFramesPerPes": 4,
            "PcrControl": "PCR_EVERY_PES_PACKET",
            "PmtPid": 480,
            "PrivateMetadataPid": 503,
            "ProgramNumber": 1,
            "PatInterval": 0,
            "PmtInterval": 0,
            "TimedMetadata": "NONE",
            "VideoPid": 481,
            "AudioPids": [
              482,
              483,
              484,
              485,
              486,
              487,
              488,
              489,
              490,
              491,
              492
            ]
          }
```

```
            },
            "VideoDescription": {
              "Width": 1280,
              "ScalingBehavior": "DEFAULT",
              "Height": 720,
              "TimecodeInsertion": "DISABLED",
              "AntiAlias": "ENABLED",
              "Sharpness": 50,
              "CodecSettings": {
                "Codec": "H_264",
                "H264Settings": {
                  "InterlaceMode": "PROGRESSIVE",
                  "NumberReferenceFrames": 3,
                  "Syntax": "DEFAULT",
                  "Softness": 0,
                  "GopClosedCadence": 1,
                  "GopSize": 2,
                  "Slices": 1,
                  "GopBReference": "DISABLED",
                  "MaxBitrate": 5000000,
                  "SlowPal": "DISABLED",
                  "SpatialAdaptiveQuantization": "ENABLED",
                  "TemporalAdaptiveQuantization": "ENABLED",
                  "FlickerAdaptiveQuantization": "DISABLED",
                  "EntropyEncoding": "CABAC",
                  "FramerateControl": "INITIALIZE_FROM_SOURCE",
                  "RateControlMode": "QVBR",
                  "CodecProfile": "MAIN",
                  "Telecine": "NONE",
                  "MinIInterval": 0,
                  "AdaptiveQuantization": "HIGH",
                  "CodecLevel": "AUTO",
                  "FieldEncoding": "PAFF",
                  "SceneChangeDetect": "TRANSITION_DETECTION",
                  "QualityTuningLevel": "SINGLE_PASS_HQ",
                  "FramerateConversionAlgorithm": "DUPLICATE_DROP",
                  "UnregisteredSeiTimecode": "DISABLED",
                  "GopSizeUnits": "SECONDS",
                  "ParControl": "INITIALIZE_FROM_SOURCE",
                  "NumberBFramesBetweenReferenceFrames": 2,
                  "RepeatPps": "DISABLED"
                }
              },
              "AfdSignaling": "NONE",
```

```
            "DropFrameTimecode": "ENABLED",
            "RespondToAfd": "NONE",
            "ColorMetadata": "INSERT"
          },
          "OutputSettings": {
            "HlsSettings": {
              "AudioGroupId": "program_audio",
              "AudioRenditionSets": "program_audio",
              "SegmentModifier": "$dt$",
              "IFrameOnlyManifest": "EXCLUDE"
            }
          },
          "NameModifier": "_720"
        },
        {
          "ContainerSettings": {
            "Container": "M3U8",
            "M3u8Settings": {}
          },
          "AudioDescriptions": [
            {
              "AudioSourceName": "Audio Selector 1",
              "CodecSettings": {
                "Codec": "AAC",
                "AacSettings": {
                  "Bitrate": 96000,
                  "CodingMode": "CODING_MODE_2_0",
                  "SampleRate": 48000
                }
              }
            }
          ],
          "OutputSettings": {
            "HlsSettings": {
              "AudioGroupId": "program_audio",
              "AudioTrackType": "ALTERNATE_AUDIO_AUTO_SELECT_DEFAULT"
            }
          },
          "NameModifier": "_audio"
        }
      ],
      "OutputGroupSettings": {
        "Type": "HLS_GROUP_SETTINGS",
        "HlsGroupSettings": {
```

```
          "ManifestDurationFormat": "INTEGER",
          "SegmentLength": 6,
          "TimedMetadataId3Period": 10,
          "CaptionLanguageSetting": "OMIT",
          "Destination": "s3://EXAMPLE-BUCKET/HLS/",
          "DestinationSettings": {
            "S3Settings": {
              "AccessControl": {
                "CannedAcl": "PUBLIC_READ"
              }
            }
          },
          "TimedMetadataId3Frame": "PRIV",
          "CodecSpecification": "RFC_4281",
          "OutputSelection": "MANIFESTS_AND_SEGMENTS",
          "ProgramDateTimePeriod": 600,
          "MinSegmentLength": 0,
          "DirectoryStructure": "SINGLE_DIRECTORY",
          "ProgramDateTime": "EXCLUDE",
          "SegmentControl": "SEGMENTED_FILES",
          "ManifestCompression": "NONE",
          "ClientCache": "ENABLED",
          "StreamInfResolution": "INCLUDE"
        }
      }
    },
    {
      "CustomName": "MP4",
      "Name": "File Group",
      "Outputs": [
        {
          "ContainerSettings": {
            "Container": "MP4",
            "Mp4Settings": {
              "CslgAtom": "INCLUDE",
              "FreeSpaceBox": "EXCLUDE",
              "MoovPlacement": "PROGRESSIVE_DOWNLOAD"
            }
          },
          "VideoDescription": {
            "Width": 1280,
            "ScalingBehavior": "DEFAULT",
            "Height": 720,
            "TimecodeInsertion": "DISABLED",
```

```
            "AntiAlias": "ENABLED",
            "Sharpness": 100,
            "CodecSettings": {
              "Codec": "H_264",
              "H264Settings": {
                "InterlaceMode": "PROGRESSIVE",
                "ParNumerator": 1,
                "NumberReferenceFrames": 3,
                "Syntax": "DEFAULT",
                "Softness": 0,
                "GopClosedCadence": 1,
                "HrdBufferInitialFillPercentage": 90,
                "GopSize": 2,
                "Slices": 2,
                "GopBReference": "ENABLED",
                "HrdBufferSize": 10000000,
                "MaxBitrate": 5000000,
                "ParDenominator": 1,
                "EntropyEncoding": "CABAC",
                "RateControlMode": "QVBR",
                "CodecProfile": "HIGH",
                "MinIInterval": 0,
                "AdaptiveQuantization": "AUTO",
                "CodecLevel": "AUTO",
                "FieldEncoding": "PAFF",
                "SceneChangeDetect": "ENABLED",
                "QualityTuningLevel": "SINGLE_PASS_HQ",
                "UnregisteredSeiTimecode": "DISABLED",
                "GopSizeUnits": "SECONDS",
                "ParControl": "SPECIFIED",
                "NumberBFramesBetweenReferenceFrames": 3,
                "RepeatPps": "DISABLED",
                "DynamicSubGop": "ADAPTIVE"
              }
            },
            "AfdSignaling": "NONE",
            "DropFrameTimecode": "ENABLED",
            "RespondToAfd": "NONE",
            "ColorMetadata": "INSERT"
          },
          "AudioDescriptions": [
            {
              "AudioTypeControl": "FOLLOW_INPUT",
              "AudioSourceName": "Audio Selector 1",
```

```
            "CodecSettings": {
              "Codec": "AAC",
              "AacSettings": {
                "AudioDescriptionBroadcasterMix": "NORMAL",
                "Bitrate": 160000,
                "RateControlMode": "CBR",
                "CodecProfile": "LC",
                "CodingMode": "CODING_MODE_2_0",
                "RawFormat": "NONE",
                "SampleRate": 48000,
                "Specification": "MPEG4"
              }
            },
            "LanguageCodeControl": "FOLLOW_INPUT",
            "AudioType": 0
          }
        ]
      }
    ],
    "OutputGroupSettings": {
      "Type": "FILE_GROUP_SETTINGS",
      "FileGroupSettings": {
        "Destination": "s3://EXAMPLE-BUCKET/MP4/",
        "DestinationSettings": {
          "S3Settings": {
            "AccessControl": {
              "CannedAcl": "PUBLIC_READ"
            }
          }
        }
      }
    }
  },
  {
    "CustomName": "Thumbnails",
    "Name": "File Group",
    "Outputs": [
      {
        "ContainerSettings": {
          "Container": "RAW"
        },
        "VideoDescription": {
          "Width": 1280,
          "ScalingBehavior": "DEFAULT",
```

```
            "Height": 720,
            "TimecodeInsertion": "DISABLED",
            "AntiAlias": "ENABLED",
            "Sharpness": 50,
            "CodecSettings": {
              "Codec": "FRAME_CAPTURE",
              "FrameCaptureSettings": {
                "FramerateNumerator": 1,
                "FramerateDenominator": 5,
                "MaxCaptures": 500,
                "Quality": 80
              }
            },
            "AfdSignaling": "NONE",
            "DropFrameTimecode": "ENABLED",
            "RespondToAfd": "NONE",
            "ColorMetadata": "INSERT"
          }
        }
      ],
      "OutputGroupSettings": {
        "Type": "FILE_GROUP_SETTINGS",
        "FileGroupSettings": {
          "Destination": "s3://EXAMPLE-BUCKET/Thumbnails/",
          "DestinationSettings": {
            "S3Settings": {
              "AccessControl": {
                "CannedAcl": "PUBLIC_READ"
              }
            }
          }
        }
      }
    }
  ],
  "AdAvailOffset": 0,
  "Inputs": [
    {
      "AudioSelectors": {
        "Audio Selector 1": {
          "Offset": 0,
          "DefaultSelection": "DEFAULT",
          "ProgramSelection": 1
        }
```

```
      },
      "VideoSelector": {
        "ColorSpace": "FOLLOW"
      },
      "FilterEnable": "AUTO",
      "PsiControl": "USE_PSI",
      "FilterStrength": 0,
      "DeblockFilter": "DISABLED",
      "DenoiseFilter": "DISABLED",
      "TimecodeSource": "EMBEDDED",
      "FileInput": "s3://EXAMPLE-INPUT-BUCKET/input.mp4"
    }
  ]
}
```

3. Di `batch-transcode` folder, buat file dengan fungsi Lambda. Anda dapat menggunakan contoh Python berikut dan memberi nama file. `convert.py`

   Operasi Batch S3 mengirimkan data tugas tertentu ke fungsi Lambda dan memerlukan data hasil kembali. Untuk contoh permintaan dan respons untuk fungsi Lambda, informasi tentang kode respons dan hasil, dan contoh fungsi Lambda untuk Operasi Batch S3, lihat. Memanggil fungsi AWS Lambda

```
import json
import os
from urllib.parse import urlparse
import uuid
import boto3

"""
When you run an S3 Batch Operations job, your job
invokes this Lambda function. Specifically, the Lambda function is
invoked on each video object listed in the manifest that you specify
for the S3 Batch Operations job in Step 5.

Input parameter "event": The S3 Batch Operations event as a request
                         for the Lambda function.

Input parameter "context": Context about the event.

Output: A result structure that Amazon S3 uses to interpret the result
        of the operation. It is a job response returned back to S3 Batch
 Operations.
```

```
"""
def handler(event, context):

    invocation_schema_version = event['invocationSchemaVersion']
    invocation_id = event['invocationId']
    task_id = event['tasks'][0]['taskId']

    source_s3_key = event['tasks'][0]['s3Key']
    source_s3_bucket = event['tasks'][0]['s3BucketArn'].split(':::')[-1]
    source_s3 = 's3://' + source_s3_bucket + '/' + source_s3_key

    result_list = []
    result_code = 'Succeeded'
    result_string = 'The input video object was converted successfully.'

    # The type of output group determines which media players can play
    # the files transcoded by MediaConvert.
    # For more information, see Creating outputs with AWS Elemental MediaConvert.
    output_group_type_dict = {
        'HLS_GROUP_SETTINGS': 'HlsGroupSettings',
        'FILE_GROUP_SETTINGS': 'FileGroupSettings',
        'CMAF_GROUP_SETTINGS': 'CmafGroupSettings',
        'DASH_ISO_GROUP_SETTINGS': 'DashIsoGroupSettings',
        'MS_SMOOTH_GROUP_SETTINGS': 'MsSmoothGroupSettings'
    }

    try:
        job_name = 'Default'
        with open('job.json') as file:
            job_settings = json.load(file)

        job_settings['Inputs'][0]['FileInput'] = source_s3

        # The path of each output video is constructed based on the values of
        # the attributes in each object of OutputGroups in the job.json file.
        destination_s3 = 's3://{0}/{1}/{2}' \
            .format(os.environ['amzn-s3-demo-destination-bucket'],
                    os.path.splitext(os.path.basename(source_s3_key))[0],
                    os.path.splitext(os.path.basename(job_name))[0])

        for output_group in job_settings['OutputGroups']:
            output_group_type = output_group['OutputGroupSettings']['Type']
            if output_group_type in output_group_type_dict.keys():
                output_group_type = output_group_type_dict[output_group_type]
```

```
                output_group['OutputGroupSettings'][output_group_type]
['Destination'] = \
                    "{0}{1}".format(destination_s3,
                                    urlparse(output_group['OutputGroupSettings']
[output_group_type]['Destination']).path)
            else:
                raise ValueError("Exception: Unknown Output Group Type {}."
                                 .format(output_group_type))

        job_metadata_dict = {
            'assetID': str(uuid.uuid4()),
            'application': os.environ['Application'],
            'input': source_s3,
            'settings': job_name
        }

        region = os.environ['AWS_DEFAULT_REGION']
        endpoints = boto3.client('mediaconvert', region_name=region) \
            .describe_endpoints()
        client = boto3.client('mediaconvert', region_name=region,
                              endpoint_url=endpoints['Endpoints'][0]['Url'],
                              verify=False)

        try:
            client.create_job(Role=os.environ['MediaConvertRole'],
                              UserMetadata=job_metadata_dict,
                              Settings=job_settings)
        # You can customize error handling based on different error codes that
        # MediaConvert can return.
        # For more information, see MediaConvert error codes.
        # When the result_code is TemporaryFailure, S3 Batch Operations retries
        # the task before the job is completed. If this is the final retry,
        # the error message is included in the final report.
        except Exception as error:
            result_code = 'TemporaryFailure'
            raise

    except Exception as error:
        if result_code != 'TemporaryFailure':
            result_code = 'PermanentFailure'
        result_string = str(error)

    finally:
        result_list.append({
```

```
            'taskId': task_id,
            'resultCode': result_code,
            'resultString': result_string,
        })

    return {
        'invocationSchemaVersion': invocation_schema_version,
        'treatMissingKeyAs': 'PermanentFailure',
        'invocationId': invocation_id,
        'results': result_list
    }
```

4. Untuk membuat paket penyebaran dengan `convert.py` dan `job.json` sebagai `.zip` file bernama`lambda.zip`, di terminal lokal Anda, buka `batch-transcode` folder yang Anda buat sebelumnya, dan jalankan perintah berikut.

   Untuk pengguna macOS, jalankan perintah berikut:

   ```
   zip -r lambda.zip convert.py job.json
   ```

   Untuk pengguna Windows, jalankan perintah berikut:

   ```
   powershell Compress-Archive convert.py lambda.zip
   ```

   ```
   powershell Compress-Archive -update job.json lambda.zip
   ```

Buat fungsi Lambda dengan peran eksekusi (konsol)

1. Buka AWS Lambda konsol di https://console.aws.amazon.com/lambda/.
2. Di panel navigasi kiri, pilih Fungsi.
3. Pilih Buat fungsi.
4. Pilih Penulis dari scratch.
5. Di bagian Informasi dasar, lakukan hal berikut:
   a. Untuk Nama fungsi, masukkan **`tutorial-lambda-convert`**.
   b. Untuk Runtime, pilih Python 3.13.
6. Pilih Ubah peran eksekusi default, dan di bawah Peran eksekusi, pilih Gunakan peran yang ada.

7. Di bawah Peran yang ada, pilih nama peran IAM yang Anda buat untuk fungsi Lambda di Langkah 3 (misalnya**`tutorial-lambda-transcode-role`**,).
8. Untuk pengaturan yang tersisa, pertahankan default.
9. Pilih Buat fungsi.

Terapkan fungsi Lambda Anda dengan arsip file.zip dan konfigurasikan fungsi Lambda (konsol)

1. Di bagian Sumber Kode halaman untuk fungsi Lambda yang Anda buat (misalnya,**`tutorial-lambda-convert`**), pilih Unggah dari lalu file.zip.
2. Pilih Unggah untuk memilih `.zip` file lokal Anda.
3. Pilih `lambda.zip` file yang Anda buat sebelumnya, dan pilih Buka.
4. Pilih Simpan.
5. Di bagian Pengaturan waktu proses, pilih Edit.
6. Untuk memberi tahu runtime Lambda metode handler mana dalam kode fungsi Lambda Anda yang akan dipanggil, masukkan di bidang Handler. **`convert.handler`**

   Ketika Anda mengkonfigurasi fungsi dalam Python, nilai pengaturan handler adalah nama file dan nama modul handler, dipisahkan oleh titik (). `.` Misalnya, `convert.handler` memanggil `handler` metode yang didefinisikan dalam `convert.py` file.
7. Pilih Simpan.
8. Pada halaman fungsi Lambda Anda, pilih tab Konfigurasi. Di panel navigasi kiri pada tab Konfigurasi, pilih Variabel lingkungan, lalu pilih Edit.
9. Pilih Tambahkan variabel lingkungan. Kemudian, masukkan Kunci dan Nilai yang ditentukan untuk masing-masing variabel lingkungan berikut:

   - Kunci: **`DestinationBucket`** Nilai: **`amzn-s3-demo-destination-bucket1`**

     Nilai ini adalah bucket S3 untuk file media keluaran yang Anda buat di Langkah 1.
   - Kunci: **`MediaConvertRole`** Nilai: **`arn:aws:iam::`*`111122223333`*`:role/tutorial-mediaconvert-role`**

     Nilai ini adalah ARN dari peran IAM untuk MediaConvert yang Anda buat di Langkah 2. Pastikan untuk mengganti ARN ini dengan ARN sebenarnya dari peran IAM Anda.
   - Kunci: **`Application`** Nilai: **`Batch-Transcoding`**

     Nilai ini adalah nama aplikasi.

10. Pilih Simpan.
11. (Opsional) Pada tab Konfigurasi, di bagian Konfigurasi umum pada panel navigasi kiri, pilih Edit. Di bidang Timeout, masukkan **2** min **0** sec. Lalu, pilih Simpan.

    Timeout adalah jumlah waktu yang memungkinkan Lambda menjalankan fungsi untuk pemanggilan sebelum menghentikannya. Default-nya adalah 3 detik. Harga didasarkan pada jumlah memori yang dikonfigurasi dan jumlah waktu kode Anda berjalan. Untuk informasi selengkapnya, lihat harga AWS Lambda.

Langkah 5: Konfigurasikan Inventaris Amazon S3 untuk bucket sumber S3 Anda

Setelah mengatur fungsi transcoding Lambda, buat pekerjaan Operasi Batch S3 untuk mentranskode satu set video. Pertama, Anda memerlukan daftar objek video masukan yang Anda inginkan Operasi Batch S3 untuk menjalankan tindakan transcoding yang ditentukan. Untuk mendapatkan daftar objek video masukan, Anda dapat membuat laporan Inventaris S3 untuk bucket sumber S3 Anda (misalnya,*amzn-s3-demo-source-bucket*).

Sublangkah

- Membuat dan mengonfigurasi bucket untuk laporan Inventaris S3 untuk video masukan
- Konfigurasikan Inventaris Amazon S3 untuk bucket sumber video S3 Anda
- Periksa laporan inventaris untuk bucket sumber video S3 Anda

Membuat dan mengonfigurasi bucket untuk laporan Inventaris S3 untuk video masukan

Untuk menyimpan laporan Inventaris S3 yang mencantumkan objek bucket sumber S3, buat bucket tujuan Inventaris S3, lalu konfigurasikan kebijakan bucket untuk bucket untuk menulis file inventaris ke bucket sumber S3.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Pilih Buat bucket.
4. Untuk nama Bucket, masukkan nama untuk bucket Anda (misalnya,*amzn-s3-demo-destination-bucket2*).
5. Untuk Wilayah AWS, pilih di Wilayah AWS mana Anda ingin ember berada.

Bucket tujuan inventaris harus Wilayah AWS sama dengan bucket sumber tempat Anda menyiapkan Inventaris S3. Bucket tujuan inventaris bisa berbeda Akun AWS.

6. Di pengaturan Blokir Akses Publik untuk bucket ini, pertahankan pengaturan default (Blokir semua akses publik diaktifkan).
7. Untuk pengaturan yang tersisa, pertahankan default.
8. Pilih Buat bucket.
9. Dalam daftar Bucket, pilih nama bucket yang baru saja Anda buat (misalnya,*amzn-s3-demo-destination-bucket2*).
10. Untuk memberikan izin Amazon S3 untuk menulis data laporan inventaris ke bucket tujuan Inventaris S3, pilih tab Izin.
11. Gulir ke bawah ke bagian Kebijakan Bucket, lalu pilih Edit. Halaman kebijakan Bucket terbuka.
12. Untuk memberikan izin untuk Inventaris S3, di bidang Kebijakan, tempel kebijakan bucket berikut.

    Ganti tiga nilai contoh dengan nilai-nilai berikut:

    - Nama bucket yang Anda buat untuk menyimpan laporan inventaris (misalnya,*amzn-s3-demo-destination-bucket2*).
    - Nama bucket sumber yang menyimpan video input (misalnya,*amzn-s3-demo-source-bucket*).
    - Akun AWS ID yang Anda gunakan untuk membuat bucket sumber video S3 (misalnya,*111122223333*.

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Sid":"InventoryAndAnalyticsExamplePolicy",
      "Effect":"Allow",
      "Principal": {"Service": "s3.amazonaws.com"},
      "Action":"s3:PutObject",
      "Resource":["arn:aws:s3:::amzn-s3-demo-destination-bucket2/*"],
      "Condition": {
          "ArnLike": {
              "aws:SourceArn": "arn:aws:s3:::amzn-s3-demo-source-bucket"
           },
```

```
            "StringEquals": {
                "aws:SourceAccount": "111122223333",
                "s3:x-amz-acl": "bucket-owner-full-control"
             }
          }
       }
    ]
}
```

13. Pilih Simpan perubahan.

Konfigurasikan Inventaris Amazon S3 untuk bucket sumber video S3 Anda

Untuk membuat daftar file datar objek video dan metadata, Anda harus mengonfigurasi S3 Inventory untuk bucket sumber video S3 Anda. Laporan inventaris terjadwal ini dapat menyertakan semua objek dalam bucket atau objek yang dikelompokkan berdasarkan awalan bersama. Dalam tutorial ini, laporan S3 Inventory mencakup semua objek video di bucket sumber S3 Anda.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Untuk mengonfigurasi laporan Inventaris S3 dari video masukan di bucket sumber S3, dalam daftar Bucket, pilih nama bucket sumber S3 (misalnya,). *amzn-s3-demo-source-bucket*
4. Pilih tab Manajemen.
5. Gulir ke bawah ke bagian Konfigurasi inventaris, dan pilih Buat konfigurasi inventaris.
6. Untuk nama konfigurasi Inventaris, masukkan nama (misalnya,**tutorial-inventory-config**).
7. Di bawah lingkup Inventaris, pilih Versi saat ini hanya untuk versi Objek dan simpan pengaturan cakupan Inventaris lainnya disetel ke default untuk tutorial ini.
8. Di bagian Laporkan detail, untuk bucket Tujuan, pilih Akun ini.
9. Untuk Tujuan, pilih Jelajahi S3, dan pilih bucket tujuan yang Anda buat sebelumnya untuk menyimpan laporan inventaris (misalnya,*amzn-s3-demo-destination-bucket2*). Kemudian, pilih Pilih jalur.

   Bucket tujuan inventaris harus Wilayah AWS sama dengan bucket sumber tempat Anda menyiapkan Inventaris S3. Bucket tujuan inventaris bisa berbeda Akun AWS.

Di bagian bucket Destination, izin bucket Destination ditambahkan ke kebijakan bucket tujuan inventaris, yang memungkinkan Amazon S3 menempatkan data di bucket tujuan inventaris. Untuk informasi selengkapnya, lihat Membuat kebijakan bucket tujuan.

10. Untuk Frekuensi, pilih Harian.
11. Untuk Format output, pilih CSV.
12. Untuk Status, pilih Aktifkan.
13. Di bagian enkripsi sisi server, pilih Nonaktifkan untuk tutorial ini.

    Untuk informasi selengkapnya, lihat Mengonfigurasi inventaris dengan menggunakan konsol S3 dan Memberikan izin kepada Amazon S3 untuk menggunakan CMK untuk enkripsi.
14. Di bidang tambahan - bagian opsional, pilih Ukuran, Terakhir dimodifikasi, dan kelas Penyimpanan.
15. Pilih Buat.

Untuk informasi selengkapnya, lihat Mengonfigurasi inventaris dengan menggunakan konsol S3.

Periksa laporan inventaris untuk bucket sumber video S3 Anda

Saat laporan inventaris diterbitkan, file manifes dikirim ke bucket tujuan Inventaris S3.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Dalam daftar Bucket, pilih nama bucket sumber video (misalnya,*amzn-s3-demo-source-bucket*).
4. Pilih Manajemen.
5. Untuk melihat apakah laporan Inventaris S3 Anda sudah siap sehingga Anda dapat membuat pekerjaan Operasi Batch S3 di Langkah 7, di bawah konfigurasi Inventaris, periksa apakah tombol Buat pekerjaan dari manifes diaktifkan.

> **Note**
>
> Diperlukan waktu hingga 48 jam untuk mengirimkan laporan inventaris pertama. Jika tombol Buat pekerjaan dari manifes dinonaktifkan, laporan inventaris pertama belum terkirim. Tunggu hingga laporan inventaris pertama dikirimkan dan tombol Buat

pekerjaan dari manifes diaktifkan sebelum Anda membuat pekerjaan Operasi Batch S3 di Langkah 7.

6. Untuk memeriksa laporan Inventaris S3 (`manifest.json`), di kolom Tujuan, pilih nama bucket tujuan inventaris yang Anda buat sebelumnya untuk menyimpan laporan inventaris (misalnya,*amzn-s3-demo-destination-bucket2*).
7. Pada tab Objects, pilih folder yang ada dengan nama bucket sumber S3 Anda (misalnya,*amzn-s3-demo-source-bucket*). Kemudian pilih nama yang Anda masukkan dalam nama konfigurasi Inventaris saat Anda membuat konfigurasi inventaris sebelumnya (misalnya,**`tutorial-inventory-config`**).

   Anda dapat melihat daftar folder dengan tanggal pembuatan laporan sebagai namanya.
8. Untuk memeriksa laporan Inventaris S3 harian untuk tanggal tertentu, pilih folder dengan nama tanggal pembuatan yang sesuai, lalu pilih`manifest.json`.
9. Untuk memeriksa detail laporan inventaris pada tanggal tertentu, pada halaman manifest.json, pilih Unduh atau Buka.

Langkah 6: Buat peran IAM untuk Operasi Batch S3

Untuk menggunakan Operasi Batch S3 untuk melakukan transcoding batch, Anda harus terlebih dahulu membuat peran IAM untuk memberikan izin Amazon S3 untuk melakukan Operasi Batch S3.

Sublangkah

- Membuat kebijakan IAM untuk Operasi Batch S3
- Buat peran IAM Operasi Batch S3 dan lampirkan kebijakan izin

Membuat kebijakan IAM untuk Operasi Batch S3

Anda harus membuat kebijakan IAM yang memberikan izin Operasi Batch S3 untuk membaca manifes masukan, menjalankan fungsi Lambda, dan menulis laporan penyelesaian pekerjaan Operasi Batch S3.

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi di sebelah kiri, pilih Kebijakan.
3. Pilih Buat kebijakan.

4. Pilih tab JSON.
5. Di bidang teks JSON, tempel kebijakan JSON berikut.

   Dalam kebijakan JSON, ganti empat nilai contoh dengan nilai berikut:

   - Nama bucket sumber yang menyimpan video masukan Anda (misalnya,*amzn-s3-demo-source-bucket*).
   - Nama bucket tujuan inventaris yang Anda buat di Langkah 5 untuk menyimpan `manifest.json` file (misalnya,*amzn-s3-demo-destination-bucket2*).
   - Nama bucket yang Anda buat di Langkah 1 untuk menyimpan file media keluaran (misalnya,*amzn-s3-demo-destination-bucket1*). Dalam tutorial ini, kita menempatkan laporan penyelesaian pekerjaan di bucket tujuan untuk file media output.
   - Peran ARN dari fungsi Lambda yang Anda buat di Langkah 4. Untuk menemukan dan menyalin peran ARN dari fungsi Lambda, lakukan hal berikut:
     - Di tab browser baru, buka halaman Fungsi di konsol Lambda di. https://console.aws.amazon.com/lambda/home#/functions
     - Dalam daftar Fungsi, pilih nama fungsi Lambda yang Anda buat di Langkah 4 (misalnya,**`tutorial-lambda-convert`**).
     - Pilih Salin ARN.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3Get",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-source-bucket/*",
                "arn:aws:s3:::amzn-s3-demo-destination-bucket2/*"
            ]
        },
        {
            "Sid": "S3PutJobCompletionReport",
            "Effect": "Allow",
```

```
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket1/*"
        },
        {
            "Sid": "S3BatchOperationsInvokeLambda",
            "Effect": "Allow",
            "Action": [
                "lambda:InvokeFunction"
            ],
            "Resource": [
                "arn:aws:lambda:us-west-2:111122223333:function:tutorial-lambda-
convert"
            ]
        }
    ]
}
```

6. Pilih Berikutnya: Tanda.
7. Pilih Berikutnya: Tinjau.
8. Di bidang Nama, masukkan **`tutorial-s3batch-policy`**.
9. Pilih Buat kebijakan.

Buat peran IAM Operasi Batch S3 dan lampirkan kebijakan izin

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi kiri, pilih Peran, lalu pilih Buat peran.
3. Pilih jenis Layanan AWSperan, lalu pilih layanan S3.
4. Di bawah Pilih kasus penggunaan Anda, pilih Operasi Batch S3.
5. Pilih Berikutnya.
6. Di bawah Lampirkan izin, masukkan nama kebijakan IAM yang Anda buat sebelumnya (misalnya,**`tutorial-s3batch-policy`**) di kotak pencarian untuk memfilter daftar kebijakan. Pilih kotak centang di samping nama kebijakan (misalnya,**`tutorial-s3batch-policy`**).
7. Pilih Berikutnya.
8. Untuk Nama peran, masukkan **`tutorial-s3batch-role`**.
9. Pilih Buat peran.

Setelah Anda membuat peran IAM untuk Operasi Batch S3, kebijakan kepercayaan berikut secara otomatis dilampirkan ke peran tersebut. Kebijakan kepercayaan ini memungkinkan prinsipal layanan Operasi Batch S3 untuk mengambil peran IAM.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"batchoperations.s3.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

Langkah 7: Buat dan jalankan pekerjaan Operasi Batch S3

Untuk membuat tugas Operasi Batch S3 untuk memproses video input di bucket sumber S3, Anda harus menentukan parameter untuk pekerjaan khusus ini.

> **Note**
>
> Sebelum Anda mulai membuat pekerjaan Operasi Batch S3, pastikan tombol Buat pekerjaan dari manifes diaktifkan. Untuk informasi selengkapnya, lihat Periksa laporan inventaris untuk bucket sumber video S3 Anda. Jika tombol Buat pekerjaan dari manifes dinonaktifkan, laporan inventaris pertama belum terkirim dan Anda harus menunggu hingga tombol diaktifkan. Setelah mengonfigurasi Inventaris Amazon S3 untuk bucket sumber S3 di Langkah 5, diperlukan waktu hingga 48 jam untuk mengirimkan laporan inventaris pertama.

Sublangkah

- Membuat tugas Operasi Batch S3
- Jalankan tugas Operasi Batch S3 untuk menginvokasi fungsi Lambda Anda
- (Opsional) Memeriksa laporan penyelesaian Anda
- (Opsional) Memantau setiap invokasi Lambda di konsol Lambda

- (Opsional) Pantau setiap pekerjaan MediaConvert transcoding video di konsol MediaConvert

Membuat tugas Operasi Batch S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Operasi Batch.
3. Pilih Buat tugas.
4. Untuk Wilayah AWS, pilih Wilayah tempat Anda ingin membuat tugas Anda.

   Dalam tutorial ini, untuk menggunakan tugas Operasi Batch S3 untuk menginvokasi fungsi Lambda, Anda harus membuat tugas di Wilayah yang sama dengan bucket sumber video S3 tempat objek yang direferensikan dalam manifes tersebut berada.
5. Di bagian Manifes, lakukan hal berikut ini:

   a. Untuk Format manifes, pilih Laporan Inventaris S3 (manifest.json).

   b. Untuk Objek manifes, pilih Telusuri S3 untuk menemukan bucket yang Anda buat di Langkah 5 untuk menyimpan laporan inventaris (misalnya, *amzn-s3-demo-destination-bucket2*). Pada halaman Objek manifes, navigasikan nama objek hingga Anda menemukan file `manifest.json` untuk tanggal tertentu. File ini mencantumkan informasi tentang semua video yang transkode batch-nya ingin Anda lakukan. Ketika Anda telah menemukan file `manifest.json` yang ingin Anda gunakan, pilih tombol opsi di sebelahnya. Kemudian, pilih Pilih jalur.

   c. (Opsional) Untuk ID versi objek manifes-opsional, masukkan ID versi untuk objek manifes jika Anda ingin menggunakan versi selain yang terbaru.
6. Pilih Berikutnya.
7. Untuk menggunakan fungsi Lambda untuk mentranskode semua objek yang terdaftar dalam file `manifest.json` yang dipilih, di bawah Jenis operasi, pilih Menginvokasi fungsi AWS Lambda .
8. Di bagian Menginvokasi fungsi Lambda, lakukan hal berikut ini:

   a. Pilih Pilih dari fungsi di akun Anda.

   b. Untuk Fungsi Lambda, pilih nama fungsi Lambda yang telah Anda buat di Langkah 4 (misalnya, **`tutorial-lambda-convert`**).

   c. Untuk Versi fungsi Lambda, pertahankan nilai default $LATEST.
9. Pilih Berikutnya. Halaman Konfigurasi opsi tambahan terbuka.

10. Di bagian Opsi tambahan, pertahankan pengaturan default.

    Untuk informasi selengkapnya tentang opsi ini, lihat Elemen permintaan pekerjaan Operasi Batch.

11. Di bagian Laporan penyelesaian, untuk Jalur ke tujuan laporan penyelesaian, pilih Telusuri S3. Temukan bucket yang telah Anda buat untuk file media output di Langkah 1 (misalnya, *amzn-s3-demo-destination-bucket1*). Pilih tombol opsi di sebelah nama bucket tersebut. Kemudian, pilih Pilih jalur.

    Untuk pengaturan Laporan penyelesaian yang tersisa, pertahankan defaultnya. Untuk informasi selengkapnya tentang pengaturan laporan penyelesaian, lihat Elemen permintaan pekerjaan Operasi Batch. Laporan penyelesaian menyimpan catatan detail tugas dan operasi yang dilakukan.

12. Di bagian Izin, pilih Pilih dari peran IAM yang ada. Untuk Peran IAM, pilih peran IAM untuk tugas Operasi Batch S3 yang Anda buat di Langkah 6 (misalnya, **`tutorial-s3batch-role`**).
13. Pilih Berikutnya.
14. Pada halaman Ulasan, tinjau pengaturannya. Lalu, pilih Buat tugas.

    Setelah S3 selesai membaca manifes tugas Operasi Batch S3 Anda, itu akan menetapkan Status tugas menjadi Menunggu konfirmasi Anda untuk menjalankan. Untuk melihat pembaruan status tugas, segarkan halaman. Anda tidak dapat menjalankan tugas Anda sampai statusnya Menunggu konfirmasi Anda untuk menjalankan.

Jalankan tugas Operasi Batch S3 untuk menginvokasi fungsi Lambda Anda

Jalankan tugas Operasi Batch Anda untuk menginvokasi fungsi Lambda untuk transkode video. Jika tugas Anda gagal, Anda dapat memeriksa laporan penyelesaian Anda untuk mengidentifikasi penyebabnya.

Untuk menjalankan tugas Operasi Batch S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Operasi Batch.
3. Dalam daftar Tugas, pilih ID tugas di baris pertama, yang merupakan tugas Operasi Batch S3 yang telah Anda buat sebelumnya.
4. Pilih Jalankan tugas.

5. Tinjau kembali parameter tugas Anda, dan konfirmasikan bahwa nilai untuk Total objek yang tercantum dalam temuan sama dengan jumlah objek dalam manifes tersebut. Lalu, pilih Jalankan tugas.

   Halaman tugas Operasi Batch S3 Anda terbuka.

6. Setelah tugas mulai berjalan, pada halaman tugas Anda, di bawah Status, periksa kemajuan tugas Operasi Batch S3 Anda, seperti Status, % Selesai, Total yang berhasil (laju), Total yang gagal (laju), Tanggal dihentikan, dan Alasan terminasi.

   Saat tugas Operasi Batch S3 selesai, lihat data di halaman tugas Anda untuk mengonfirmasi bahwa tugas selesai seperti yang diharapkan.

   Jika lebih dari 50 persen operasi objek tugas Operasi Batch S3 gagal setelah lebih dari 1.000 pengoperasian yang telah dicoba, tugas tersebut secara otomatis gagal. Untuk memeriksa laporan penyelesaian Anda guna mengidentifikasi penyebab kegagalan, gunakan prosedur opsional berikut ini.

(Opsional) Memeriksa laporan penyelesaian Anda

Anda dapat menggunakan laporan penyelesaian untuk menentukan objek mana yang gagal, serta penyebab kegagalannya.

Untuk memeriksa laporan penyelesaian Anda terkait detail mengenai objek yang gagal

1. Pada halaman tugas Operasi Batch S3 Anda, gulir ke bawah ke bagian Laporan penyelesaian, dan pilih tautan di bawah Tujuan laporan penyelesaian.

   Halaman bucket tujuan output S3 terbuka.

2. Pada tab Objek, pilih folder yang memiliki nama yang diakhiri dengan ID tugas Operasi Batch S3 yang telah Anda buat sebelumnya.
3. Pilih hasil/.
4. Centang kotak di samping file `.csv`.
5. Untuk melihat laporan tugas, pilih Buka atau Unduh.

(Opsional) Memantau setiap invokasi Lambda di konsol Lambda

Setelah tugas Operasi Batch S3 mulai berjalan, tugas akan memanggil fungsi Lambda untuk setiap objek video input. S3 menulis log dari setiap pemanggilan CloudWatch Lambda ke Log. Anda dapat menggunakan dasbor pemantauan konsol Lambda untuk memantau fungsi Lambda Anda.

1. Buka AWS Lambda konsol di https://console.aws.amazon.com/lambda/.
2. Di panel navigasi kiri, pilih Fungsi.
3. Dalam daftar Fungsi, pilih nama fungsi Lambda yang Anda buat di Langkah 4 (misalnya, **`tutorial-lambda-convert`**).
4. Pilih tab Monitor.
5. Di bawah Metrik, lihat metrik runtime untuk fungsi Lambda Anda.
6. Di bawah Log, lihat data log untuk setiap pemanggilan Lambda melalui Wawasan Log. CloudWatch

> **Note**
>
> Saat Anda menggunakan Operasi Batch S3 dengan fungsi Lambda, fungsi Lambda dipanggil pada setiap objek. Jika tugas Operasi Batch S3 Anda berukuran besar, tugas ini dapat menginvokasi beberapa fungsi Lambda secara bersamaan, yang menyebabkan lonjakan konkurensi Lambda.
> Masing-masing Akun AWS memiliki kuota konkurensi Lambda per Wilayah. Untuk informasi selengkapnya, lihat AWS Lambda Penskalaan Fungsi dalam AWS Lambda Panduan Pengembang. Praktik terbaik untuk menggunakan fungsi Lambda dengan Operasi Batch S3 adalah dengan menetapkan batas konkurensi pada fungsi Lambda itu sendiri. Menetapkan batas konkurensi membuat tugas tidak menghabiskan sebagian besar konkurensi Lambda Anda, dan berpotensi throttling fungsi lain di akun Anda. Untuk informasi selengkapnya, lihat Mengelola konkurensi cadangan Lambda di AWS Lambda Panduan Pengembang.

(Opsional) Pantau setiap pekerjaan MediaConvert transcoding video di konsol MediaConvert

MediaConvert Pekerjaan melakukan pekerjaan transcoding file media. Saat pekerjaan Operasi Batch S3 Anda memanggil fungsi Lambda Anda untuk setiap video, setiap pemanggilan fungsi Lambda membuat pekerjaan transcoding untuk setiap video input. MediaConvert

1. Masuk ke AWS Management Console dan buka MediaConvert konsol di https://console.aws.amazon.com/mediaconvert/.
2. Jika halaman MediaConvert pengantar muncul, pilih Memulai.
3. Dari daftar Tugas, lihat setiap baris untuk memantau tugas transkode bagi setiap video input.
4. Identifikasi baris tugas yang ingin Anda periksa, dan pilih tautan ID tugas untuk membuka halaman detail tugas.
5. Pada halaman ringkasan Job, di bawah Output, pilih tautan untuk output HLS, MP4, atau Thumbnail, tergantung pada apa yang didukung oleh browser Anda, untuk membuka bucket tujuan S3 untuk file media keluaran.
6. Di folder yang sesuai (HLS, MP4, atau Thumbnail) bucket tujuan keluaran S3 Anda, pilih nama objek file media keluaran.

   Halaman detail objek terbuka.
7. Pada halaman detail objek, di bawah Ikhtisar objek, pilih tautan di bawah URL objek untuk menonton file media output yang telah ditranskode.

Langkah 8: Memeriksa file media output dari bucket tujuan S3 Anda

Untuk memeriksa file media output dari bucket tujuan S3 Anda

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Dalam daftar Bucket, pilih nama bucket tujuan S3 untuk file media output yang Anda buat di Langkah 1 (misalnya, *amzn-s3-demo-destination-bucket1*).
4. Pada tab Objek, setiap video input memiliki folder yang memiliki nama video input. Setiap folder berisi file media output transkode untuk video input.

   Untuk memeriksa file media output untuk video input, lakukan hal berikut ini:

   a. Pilih folder dengan nama video input yang ingin Anda periksa.
   b. Pilih folder Default/.
   c. Pilih folder untuk format transkode (HLS, MP4, atau thumbnail dalam tutorial ini).
   d. Pilih nama file media output.

e. Untuk melihat file yang ditranskode, pada halaman detail objek, pilih tautan di bawah URL Objek.

   File media output dalam format HLS dibagi menjadi segmen-segmen yang pendek. Untuk memutar video ini, sematkan URL objek file `.m3u8` di pemutar yang kompatibel.

Langkah 9: Membersihkan

Jika Anda mentranskode video menggunakan Operasi Batch S3, Lambda, MediaConvert dan hanya sebagai latihan pembelajaran, hapus AWS sumber daya yang Anda alokasikan sehingga Anda tidak lagi menambah biaya.

Sublangkah

- Hapus konfigurasi Inventaris S3 untuk bucket sumber S3 Anda
- Hapus fungsi Lambda
- Hapus grup CloudWatch log
- Menghapus peran IAM bersama dengan kebijakan inline untuk peran IAM
- Hapus kebijakan IAM yang dikelola pelanggan
- Kosongkan bucket S3
- Menghapus bucket S3

Hapus konfigurasi Inventaris S3 untuk bucket sumber S3 Anda

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Dalam daftar Bucket, pilih nama bucket sumber Anda (misalnya, *`amzn-s3-demo-source-bucket`*).
4. Pilih tab Manajemen.
5. Di bagian Konfigurasi inventaris, pilih tombol opsi di sebelah konfigurasi inventaris yang Anda buat di Langkah 5 (misalnya, **`tutorial-inventory-config`**).
6. Pilih Hapus, lalu pilih Konfirmasi.

Hapus fungsi Lambda

1. Buka AWS Lambda konsol di https://console.aws.amazon.com/lambda/.
2. Di panel navigasi kiri, pilih Fungsi.
3. Centang kotak di samping fungsi yang Anda buat di Langkah 4 (misalnya, **`tutorial-lambda-convert`**).
4. Pilih Tindakan, lalu pilih Hapus.
5. Di kotak dialog Hapus fungsi, pilih Hapus.

Hapus grup CloudWatch log

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi sebelah kiri, pilih Log, lalu pilih Grup log.
3. Pilih kotak centang di samping grup log yang memiliki nama yang diakhiri dengan fungsi Lambda yang Anda buat di Langkah 4 (misalnya, **`tutorial-lambda-convert`**).
4. Pilih Tindakan, lalu pilih Hapus grup log.
5. Di kotak dialog Hapus grup log, pilih Hapus.

Menghapus peran IAM bersama dengan kebijakan inline untuk peran IAM

Untuk menghapus peran IAM yang Anda buat di Langkah 2, Langkah 3, dan Langkah 6, lakukan hal berikut ini:

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi sebelah kiri, pilih Peran, lalu centang kotak di samping nama peran yang ingin Anda hapus.
3. Pilih Hapus di bagian atas halaman.
4. Di dalam kotak dialog konfirmasi, masukkan respons yang diperlukan di dalam kolom input teks berdasarkan prompt, lalu pilih Hapus.

Hapus kebijakan IAM yang dikelola pelanggan

Untuk menghapus kebijakan IAM yang dikelola pelanggan yang Anda buat di Langkah 6, lakukan hal berikut ini:

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi di sebelah kiri, pilih Kebijakan.
3. Pilih tombol opsi di samping kebijakan yang Anda buat di Langkah 6 (misalnya, **`tutorial-s3batch-policy`**). Anda dapat menggunakan kotak pencarian untuk memfilter daftar kebijakan.
4. Pilih Tindakan, lalu pilih Hapus.
5. Konfirmasi bahwa Anda ingin menghapus kebijakan ini dengan memasukkan namanya di kolom teks, lalu pilih Hapus.

Kosongkan bucket S3

Untuk mengosongkan bucket S3 yang Anda buat di Prasyarat, Langkah 1, dan Langkah 5, lakukan hal berikut ini:

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Di daftar Bucket, pilih tombol opsi di samping nama bucket yang ingin Anda kosongkan, lalu pilih Kosongkan.
4. Di halaman Bucket kosong, konfirmasikan bahwa Anda ingin mengosongkan bucket dengan mengetik **`permanently delete`** ke dalam bidang teks, lalu pilih Kosongkan.

Menghapus bucket S3

Untuk menghapus bucket S3 yang Anda buat di Prasyarat, Langkah 1, dan Langkah 5, lakukan hal berikut ini:

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Di dalam daftar Bucket, pilih tombol opsi di samping nama bucket yang ingin Anda hapus.
4. Pilih Hapus.
5. Di halaman Hapus bucket, konfirmasikan bahwa Anda ingin menghapus bucket dengan memasukkan nama bucket ke dalam bidang teks, lalu pilih Hapus bucket.

Langkah selanjutnya

Setelah menyelesaikan tutorial ini, Anda dapat menjelajahi lebih lanjut kasus penggunaan relevan lainnya:

- Anda dapat menggunakan Amazon CloudFront untuk melakukan streaming file media yang ditranskode ke pemirsa di seluruh dunia. Untuk informasi selengkapnya, lihat Tutorial: Hosting video streaming sesuai permintaan dengan Amazon S3, Amazon, dan CloudFront Amazon Route 53.
- Anda dapat mentranskode video saat Anda mengunggahnya ke bucket sumber S3. Untuk melakukannya, Anda dapat mengonfigurasi pemicu peristiwa Amazon S3 yang secara otomatis memanggil fungsi Lambda untuk mentranskode objek baru di S3. MediaConvert Untuk informasi selengkapnya, lihat Tutorial: Menggunakan pemicu Amazon S3 untuk menginvokasi fungsi Lambda di AWS Lambda Panduan Pengembang.

Pemecahan Masalah Operasi Batch

Topik berikut mencantumkan kesalahan umum untuk membantu Anda memecahkan masalah yang mungkin Anda temui saat bekerja dengan Operasi Batch Amazon S3.

Untuk memecahkan masalah dengan Replikasi Batch S3, lihat. the section called “Kesalahan Replikasi Batch”

Kesalahan Umum

- Laporan pekerjaan tidak terkirim saat ada masalah izin atau mode retensi Kunci Objek S3 diaktifkan
- Operasi Batch gagal objek dengan kesalahan 400 InvalidRequest: Tugas gagal karena hilang VersionId
- Buat kegagalan pekerjaan dengan opsi tag pekerjaan diaktifkan
- Akses Ditolak untuk membaca manifes

Laporan pekerjaan tidak terkirim saat ada masalah izin atau mode retensi Kunci Objek S3 diaktifkan

Kesalahan berikut terjadi jika izin yang diperlukan tidak ada atau mode retensi Kunci Objek (baik mode tata kelola atau mode kepatuhan) diaktifkan di bucket tujuan.

Kesalahan: Alasan kegagalan. Laporan pekerjaan tidak dapat ditulis ke bucket laporan Anda. Silakan periksa izin Anda.

Peran AWS Identity and Access Management (IAM) dan kebijakan kepercayaan harus dikonfigurasi untuk mengizinkan Operasi Batch `s3:PutObject` izin ke PUT objek dalam bucket tempat laporan akan dikirimkan. Jika izin yang diperlukan ini tidak ada, kegagalan pengiriman laporan pekerjaan akan terjadi.

Saat mode retensi diaktifkan, bucket dilindungi write-once-read-many (WORM). Kunci Objek dengan mode retensi yang diaktifkan pada bucket tujuan tidak didukung, sehingga upaya pengiriman laporan penyelesaian pekerjaan gagal. Untuk memperbaiki masalah ini, pilih bucket tujuan untuk laporan penyelesaian pekerjaan yang tidak mengaktifkan mode retensi Kunci Objek.

Operasi Batch gagal objek dengan kesalahan 400 InvalidRequest: Tugas gagal karena hilang VersionId

Contoh kesalahan berikut terjadi jika pekerjaan Operasi Batch melakukan tindakan pada objek dalam bucket Penentuan Versi dan menemukan objek dalam manifes dengan bidang ID versi kosong.

Kesalahan: *`bucket_name,prefix/file_name`*, gagal, 400InvalidRequest, Tugas gagal karena hilang VersionId

Kesalahan ini terjadi karena bidang ID versi dalam manifes adalah string kosong, bukan string `null` literal.

Operasi Batch akan gagal untuk objek atau banyak objek tertentu, tetapi tidak seluruh pekerjaan. Masalah ini terjadi jika format manifes dikonfigurasi untuk menggunakan versi IDs selama operasi. Pekerjaan non-versi tidak mengalami masalah ini karena mereka hanya beroperasi pada versi terbaru dari setiap objek dan mengabaikan versi IDs dalam manifes.

Untuk memperbaiki masalah ini, ubah versi kosong IDs menjadi `null` string. Untuk informasi selengkapnya, lihat the section called “Mengonversi string ID versi kosong menjadi string null”.

Buat kegagalan pekerjaan dengan opsi tag pekerjaan diaktifkan

Tanpa izin `s3:PutJobTagging`, membuat pekerjaan Operasi Batch dengan opsi tag pekerjaan diaktifkan menyebabkan kesalahan `403 access denied`.

Untuk membuat pekerjaan Operasi Batch dengan opsi tag pekerjaan diaktifkan, pengguna AWS Identity and Access Management (IAM) yang membuat pekerjaan Operasi Batch harus memiliki `s3:PutJobTagging` izin selain `s3:CreateJob` izin.

Untuk informasi lebih lanjut tentang izin yang diperlukan untuk Operasi Batch, lihatthe section called “Memberikan izin”.

Akses Ditolak untuk membaca manifes

Jika Operasi Batch tidak dapat membaca file manifes saat Anda mencoba membuat pekerjaan Operasi Batch, kesalahan berikut dapat terjadi.

AWS CLI

Alasan kegagalan Membaca manifes dilarang: AccessDenied

Konsol Amazon S3

Peringatan: Tidak bisa mendapatkan objek manifes ETag. Tentukan objek yang berbeda untuk melanjutkan.

Untuk mengatasi masalah ini, lakukan hal berikut:

- Verifikasi bahwa peran IAM untuk Akun AWS yang Anda gunakan untuk membuat pekerjaan Operasi Batch memiliki `s3:GetObject` izin. Peran IAM akun harus memiliki `s3:GetObject` izin untuk mengizinkan Operasi Batch membaca file manifes.

  Untuk informasi lebih lanjut tentang izin yang diperlukan untuk Operasi Batch, lihat the section called “Memberikan izin”.
- Periksa metadata objek manifes untuk ketidakcocokan akses apa pun dengan Kepemilikan Objek S3. Untuk informasi selengkapnya tentang Kepemilikan Objek S3, lihat the section called “Mengontrol kepemilikan objek”.
- Periksa apakah kunci AWS Key Management Service (AWS KMS) digunakan untuk mengenkripsi file manifes.

  Operasi Batch mendukung laporan inventaris CSV yang dienkripsi AWS KMS. Namun, Operasi Batch tidak mendukung file manifes CSV yang dienkripsi AWS KMS. Untuk informasi selengkapnya, lihat Mengonfigurasi Inventaris Amazon S3 dan Menentukan manifes.

Kueri data di tempat dengan Amazon S3 Select

> **Important**
>
> Amazon S3 Select tidak lagi tersedia untuk pelanggan baru. Pelanggan Amazon S3 Select yang sudah ada dapat terus menggunakan fitur seperti biasa. Pelajari selengkapnya

Dengan Amazon S3 Select, Anda dapat menggunakan pernyataan bahasa kueri terstruktur (SQL) untuk memfilter konten objek Amazon S3 dan mengambil hanya subset data yang Anda butuhkan. Dengan menggunakan Amazon S3 Select untuk memfilter data ini, Anda dapat mengurangi jumlah data yang ditransfer oleh Amazon S3, sehingga mengurangi biaya dan latensi untuk pengambilan data ini.

Amazon S3 Select hanya memungkinkan Anda untuk menanyakan satu objek pada satu waktu. Ia bekerja pada objek yang disimpan dalam CSV, JSON, atau Apache Parquet format. Ia juga bekerja dengan objek yang dikompresi dengan GZIP atau BZIP2 (hanya untuk objek CSV dan JSON), dan objek terenkripsi sisi server. Anda dapat menentukan format hasil sebagai CSV atau JSON, dan Anda dapat menentukan pembatasan untuk catatan di dalam hasilnya.

Anda dapat mengajukan pemindahan ekspresi SQL ke Amazon S3. Amazon S3 Select mendukung squbset SQL. Untuk informasi selengkapnya tentang elemen SQL yang didukung oleh Amazon S3 Select, lihat Referensi SQL untuk Amazon S3 Select.

Anda dapat melakukan kueri SQL dengan menggunakan konsol Amazon S3,AWS CLI(), AWS Command Line Interface operasi REST API, `SelectObjectContent` atau. AWS SDKs

> **Note**
>
> Konsol Amazon S3 membatasi jumlah data yang ditampilkan hingga 40 MB. Untuk mengambil lebih banyak data, gunakan AWS CLI atau API.

Persyaratan dan batasan

Berikut adalah persyaratan untuk menggunakan Amazon S3 Select:

- Anda harus memiliki izin `s3:GetObject` untuk objek yang Anda minta.

- Jika objek yang Anda tanyakan dienkripsi dengan enkripsi di sisi server dengan kunci yang disediakan pelanggan (SSE-C), Anda harus menggunakan `https`, dan Anda harus menyediakan kunci enkripsi saat pengajuan.

Batasan berikut berlaku saat menggunakan Amazon S3 Select:

- S3 Select dapat query hanya satu objek per permintaan.
- Panjang maksimum ekspresi SQL adalah 256 KB.
- Panjang maksimum catatan dalam input atau hasil adalah 1 MB.
- Amazon S3 Select hanya dapat mengeluarkan data yang disimpan dengan format output JSON.
- Anda tidak dapat menanyakan objek yang disimpan di kelas penyimpanan S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive, atau Reduced Redundancy Storage (RRS). Anda juga tidak dapat menanyakan objek yang disimpan di tingkat Akses Arsip Tingkat Cerdas S3 atau tingkat Akses Arsip Dalam Tingkat Cerdas S3. Untuk informasi selengkapnya tentang kelas penyimpanan, lihat Memahami dan mengelola kelas penyimpanan Amazon S3.

Batasan tambahan berlaku saat menggunakan Amazon S3 Select dengan Parquet objek:

- Amazon S3 Select hanya mendukung kompresi berbentuk kolom yang menggunakan GZIP atau Snappy. Amazon S3 Select tidak mendukung kompresi seluruh objek untuk objek. Parquet
- Amazon S3 Select tidak mendukung output Parquet. Anda harus menentukan format output sebagai CSV atau JSON.
- Ukuran maksimum untuk grup baris yang tidak dikompres adalah 512 MB.
- Anda harus menggunakan jenis data yang ditentukan di dalam skema objek.
- Pemilihan pada bidang berulang hanya akan menampilkan nilai terakhir.

Membuat permintaan

Saat Anda membuat sebuah permintaan, Anda harus menyediakan perincian objek yang sedang diminta dengan menggunakan objek `InputSerialization`. Anda dapat menyediakan perincian tentang cara menampilkan hasil dengan menggunakan objek `OutputSerialization`. Anda juga dapat menyertakan ekspresi SQL yang digunakan oleh Amazon S3 untuk memfilter permintaan.

Untuk informasi lebih lanjut tentang pembuatan permintaan Amazon S3 Select, lihat SelectObjectContent di Referensi API Amazon Simple Storage Service. Anda juga dapat melihat salah satu contoh kode SDK di bagian berikut.

Permintaan yang menggunakan rentang pemindaian

Dengan Amazon S3 Select, Anda dapat memindai subset sebuah objek dengan menentukan rentang byte ke kueri. Kemampuan ini memungkinkan Anda melakukan pemindaian pada keseluruhan objek secara paralel dengan membagi pekerjaan menjadi beberapa permintaan Amazon S3 Select yang terpisah untuk serangkaian rentang pemindaian non-tumpang tindih.

Rentang pemindaian tidak perlu diselaraskan dengan batasan catatan. Permintaan rentang pemindaian Amazon S3 Select dijalankan di seluruh rentang byte yang Anda tentukan. Catatan yang dimulai dengan rentang pemindaian yang ditentukan tetapi melampaui rentang pemindaian tersebut akan diproses oleh kueri. Contoh berikut menampilkan objek Amazon S3 yang berisi serangkaian catatan dengan format CSV yang dibatasi baris:

```
A,B
C,D
D,E
E,F
G,H
I,J
```

Misalkan Anda menggunakan parameter `ScanRange` Amazon S3 Select dan Start pada (Byte) 1 dan End pada (Byte) 4. Jadi, rentang pemindaian akan diawali di `","` dan pemindaian dilakukan hingga akhir catatan yang dimulai di `C`. Permintaan rentang pemindaian Anda akan menampilkan hasil `C, D` karena hasil tersebut adalah akhir catatannya.

Amazon S3 Pilih dukungan permintaan rentang pemindaianParquet, CSV (tanpa pembatas yang dikutip), atau objek JSON (hanya dalam mode). `LINES` Objek CSV dan JSON harus dalam kondisi tidak dikompres. Untuk objek CSV dan JSON berbasis baris, ketika rentang pemindaian ditentukan sebagai bagian dari permintaan Amazon S3 Select, semua catatan yang dimulai di dalam rentang pemindaian akan diproses. Untuk objek Parquet, semua grup baris yang dimulai di dalam rentang pemindaian yang diminta akan diproses.

Permintaan rentang pemindaian Amazon S3 Select tersedia untuk digunakan dengan AWS CLI, Amazon S3 API, dan. AWS SDKs Anda dapat menggunakan parameter `ScanRange` di dalam permintaan Amazon S3 Select untuk fitur ini. Untuk informasi lebih lanjut, lihat SelectObjectContent di dalam Referensi API Amazon Simple Storage Service.

Kesalahan

Amazon S3 Select akan menampilkan kode kesalahan dan pesan kesalahan terkait saat ditemukannya masalah ketika suatu kueri sedang dijalankan. Untuk daftar kode dan deskripsi kesalahan, lihat bagian Daftar Kode Kesalahan SELECT Object Content dari halaman Respons Kesalahan di dalam Referensi API Amazon Simple Storage Service.

Untuk informasi lebih lanjut tentang Amazon S3 Select, lihat topik-topik berikut.

Topik

- Contoh menggunakan Amazon S3 Select pada objek
- Referensi SQL untuk Amazon S3 Select

Contoh menggunakan Amazon S3 Select pada objek

> **Important**
>
> Amazon S3 Select tidak lagi tersedia untuk pelanggan baru. Pelanggan Amazon S3 Select yang sudah ada dapat terus menggunakan fitur seperti biasa. Pelajari selengkapnya

Anda dapat menggunakan S3 Select untuk memilih konten dari satu objek dengan menggunakan konsol Amazon S3, REST API, dan. AWS SDKs

Untuk informasi lebih lanjut tentang fungsi SQL yang didukung untuk S3 Select, lihat Fungsi SQL.

Menggunakan konsol S3

Untuk memilih konten dari objek di konsol Amazon S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Pilih bucket yang berisi objek yang ingin Anda pilih kontennya, lalu pilih nama objeknya.
4. Pilih Tindakan objek, dan pilih Kueri dengan S3 Select.
5. Konfigurasikan Pengaturan input, berdasarkan format data input Anda.
6. Konfigurasikan Pengaturan output, berdasarkan format output yang ingin Anda terima.

7. Untuk mengekstrak catatan dari objek yang dipilih, di bawah kueri SQL, masukkan perintah SELECT SQL. Untuk informasi lebih lanjut tentang cara menulis perintah SQL, lihat Referensi SQL untuk Amazon S3 Select.
8. Setelah memasukkan kueri SQL, pilih Jalankan kueri SQL. Kemudian, di bawah Hasil kueri, Anda akan melihat hasil kueri SQL Anda.

Menggunakan API REST

Anda dapat menggunakan AWS SDKs untuk memilih konten dari objek. Namun, jika aplikasi Anda memerlukannya, Anda dapat mengirimkan permintaan REST secara langsung. Untuk informasi lebih lanjut tentang permintaan dan format respons, kunjungi SelectObjectContent.

Menggunakan AWS SDKs

Anda dapat menggunakan Amazon S3 Select untuk memilih beberapa konten objek dengan menggunakan metode ini`selectObjectContent`. Jika metode ini berhasil, Amazon S3 Select akan menampilkan ekspresi SQL.

Java

Kode Java berikut ini menampilkan nilai kolom pertama untuk setiap catatan yang disimpan di dalam objek yang mengandung data yang disimpan dengan format CSV. Kode ini juga meminta agar pesan `Progress` dan `Stats` dapat ditampilkan. Anda harus menyediakan nama bucket yang valid dan objek yang mengandung data dengan format CSV.

Untuk petunjuk cara membuat dan menguji sampel yang berfungsi, lihat Memulai di Panduan AWS SDK untuk Java Pengembang.

```
package com.amazonaws;

import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.CSVInput;
import com.amazonaws.services.s3.model.CSVOutput;
import com.amazonaws.services.s3.model.CompressionType;
import com.amazonaws.services.s3.model.ExpressionType;
import com.amazonaws.services.s3.model.InputSerialization;
import com.amazonaws.services.s3.model.OutputSerialization;
import com.amazonaws.services.s3.model.SelectObjectContentEvent;
import com.amazonaws.services.s3.model.SelectObjectContentEventVisitor;
import com.amazonaws.services.s3.model.SelectObjectContentRequest;
```

```
import com.amazonaws.services.s3.model.SelectObjectContentResult;

import java.io.File;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.concurrent.atomic.AtomicBoolean;

import static com.amazonaws.util.IOUtils.copy;

/**
 * This example shows how to query data from S3Select and consume the response in
 the form of an
 * InputStream of records and write it to a file.
 */

public class RecordInputStreamExample {

    private static final String BUCKET_NAME = "${my-s3-bucket}";
    private static final String CSV_OBJECT_KEY = "${my-csv-object-key}";
    private static final String S3_SELECT_RESULTS_PATH = "${my-s3-select-results-
path}";
    private static final String QUERY = "select s._1 from S3Object s";

    public static void main(String[] args) throws Exception {
        final AmazonS3 s3Client = AmazonS3ClientBuilder.defaultClient();

        SelectObjectContentRequest request = generateBaseCSVRequest(BUCKET_NAME,
 CSV_OBJECT_KEY, QUERY);
        final AtomicBoolean isResultComplete = new AtomicBoolean(false);

        try (OutputStream fileOutputStream = new FileOutputStream(new File
 (S3_SELECT_RESULTS_PATH));
             SelectObjectContentResult result =
 s3Client.selectObjectContent(request)) {
            InputStream resultInputStream =
 result.getPayload().getRecordsInputStream(
                    new SelectObjectContentEventVisitor() {
                        @Override
                        public void visit(SelectObjectContentEvent.StatsEvent event)
                        {
                            System.out.println(
                                    "Received Stats, Bytes Scanned: " +
 event.getDetails().getBytesScanned()
```

```
                                                +  " Bytes Processed: " +
event.getDetails().getBytesProcessed());
                        }

                        /*
                         * An End Event informs that the request has finished
successfully.
                         */
                        @Override
                        public void visit(SelectObjectContentEvent.EndEvent event)
                        {
                            isResultComplete.set(true);
                            System.out.println("Received End Event. Result is
complete.");
                        }
                    }
            );

            copy(resultInputStream, fileOutputStream);
        }

        /*
         * The End Event indicates all matching records have been transmitted.
         * If the End Event is not received, the results may be incomplete.
         */
        if (!isResultComplete.get()) {
            throw new Exception("S3 Select request was incomplete as End Event was
not received.");
        }
    }

    private static SelectObjectContentRequest generateBaseCSVRequest(String bucket,
String key, String query) {
        SelectObjectContentRequest request = new SelectObjectContentRequest();
        request.setBucketName(bucket);
        request.setKey(key);
        request.setExpression(query);
        request.setExpressionType(ExpressionType.SQL);

        InputSerialization inputSerialization = new InputSerialization();
        inputSerialization.setCsv(new CSVInput());
        inputSerialization.setCompressionType(CompressionType.NONE);
        request.setInputSerialization(inputSerialization);
```

```
        OutputSerialization outputSerialization = new OutputSerialization();
        outputSerialization.setCsv(new CSVOutput());
        request.setOutputSerialization(outputSerialization);

        return request;
    }
}
```

JavaScript

Untuk JavaScript contoh yang menggunakan operasi AWS SDK untuk JavaScript with S3 `SelectObjectContent` API untuk memilih catatan dari file JSON dan CSV yang disimpan di Amazon S3, lihat posting blog Memperkenalkan dukungan untuk Amazon S3 Select di file. AWS SDK untuk JavaScript

Python

Untuk contoh Python tentang penggunaan kueri SQL untuk melakukan pencarian lewat data yang dimuat ke Amazon S3 sebagai file nilai terpisah-koma (CSV) dengan menggunakan S3 Select, lihat postingan blog Meminta data tanpa server atau basis data dengan menggunakan Amazon S3 Select.

Referensi SQL untuk Amazon S3 Select

> **Important**
>
> Amazon S3 Select tidak lagi tersedia untuk pelanggan baru. Pelanggan Amazon S3 Select yang sudah ada dapat terus menggunakan fitur seperti biasa. Pelajari selengkapnya

Referensi ini berisi deskripsi elemen bahasa kueri terstruktur (SQL) yang didukung oleh Amazon S3 Select.

Topik

- Perintah SELECT
- Jenis data
- Operator
- Kata kunci terpesan
- Fungsi SQL

Perintah SELECT

> **Important**
>
> Amazon S3 Select tidak lagi tersedia untuk pelanggan baru. Pelanggan Amazon S3 Select yang sudah ada dapat terus menggunakan fitur seperti biasa. Pelajari selengkapnya

Amazon S3 Select hanya mendukung perintah SQL `SELECT`. Klausul standar ANSI berikut didukung untuk `SELECT`:

- Daftar `SELECT`
- Klausul `FROM`
- Klausul `WHERE`
- Klausul `LIMIT`

> **Note**
>
> Kueri Amazon S3 Select saat ini tidak mendukung subkueri atau penggabungan.

Daftar SELECT

Daftar `SELECT` mencantumkan kolom, fungsi, dan ekspresi yang ingin Anda tampilkan dari kueri. Daftar ini mewakili output kueri.

```
SELECT *
SELECT projection1 AS column_alias_1, projection2 AS column_alias_2
```

Bentuk pertama `SELECT` dengan `*` (tanda bintang) mengembalikan setiap baris yang melewati klausul `WHERE`, apa adanya. Bentuk kedua `SELECT` membuat baris dengan ekspresi skalar output ***projection1*** dan ***projection2*** yang ditentukan pengguna untuk setiap kolom.

Klausul FROM

Amazon S3 Select mendukung bentuk klausul `FROM` berikut:

```
FROM table_name
```

```
FROM table_name alias
FROM table_name AS alias
```

Dalam setiap bentuk klausul FROM, table_name adalah S3Object yang sedang dikuerikan. Pengguna yang berasal dari basis data relasional tradisional dapat menganggap hal ini sebagai skema basis data yang berisi beberapa tampilan di atas tabel.

Mengikuti SQL standar, klausul FROM membuat baris yang difilter dalam klausul WHERE dan diproyeksikan dalam daftar SELECT.

Untuk objek JSON yang disimpan di Amazon S3 Select, Anda juga dapat menggunakan bentuk klausul FROM berikut:

```
FROM S3Object[*].path
FROM S3Object[*].path alias
FROM S3Object[*].path AS alias
```

Menggunakan bentuk klausul FROM, Anda dapat memilih dari array atau objek di dalam objek JSON. Anda dapat menentukan path menggunakan salah satu bentuk berikut:

- Berdasarkan nama (dalam objek): .*name* atau ['*name*']
- Berdasarkan indeks (dalam array): [*index*]
- Berdasarkan karakter wildcard (dalam objek): .*
- Berdasarkan karakter wildcard (dalam array): [*]

> **Note**
>
> - Bentuk klausul FROM ini hanya berfungsi dengan objek JSON.
> - Karakter wildcard selalu mengeluarkan setidaknya satu catatan. Apabila tidak ada catatan yang cocok, Amazon S3 Select mengeluarkan nilai MISSING. Selama serialisasi output (setelah kueri selesai berjalan), Amazon S3 Select menggantikan nilai MISSING dengan catatan kosong.
> - Fungsi agregat (AVG, COUNT, MAX, MIN, dan SUM) melewati nilai MISSING.
> - Apabila Anda tidak menyediakan alias saat menggunakan karakter wildcard, Anda dapat merujuk baris menggunakan elemen terakhir di jalur tersebut. Misalnya, Anda dapat memilih semua harga dari daftar pembukuan menggunakan kueri SELECT price FROM S3Object[*].books[*].price. Apabila jalur diakhiri karakter wildcard alih-

alih nama, maka Anda dapat menggunakan nilai _1 tersebut untuk merujuk ke baris. Misalnya, alih-alih `SELECT price FROM S3Object[*].books[*].price`, Anda dapat menggunakan kueri `SELECT _1.price FROM S3Object[*].books[*]`.

- Amazon S3 Select selalu memperlakukan dokumen JSON sebagai array nilai tingkat root. Dengan demikian, sekalipun objek JSON yang Anda kueri hanya memiliki satu elemen root, klausul `FROM` harus dimulai dengan `S3Object[*]`. Namun, karena alasan kompatibilitas, Amazon S3 Select memungkinkan Anda menghilangkan karakter wildcard jika Anda tidak menyertakan jalur. Dengan demikian, klausul lengkap `FROM S3Object` setara dengan `FROM S3Object[*] as S3Object`. Apabila Anda menyertakan jalur, Anda juga harus menggunakan karakter wildcard. Jadi, `FROM S3Object` dan `FROM S3Object[*].`*`path`* keduanya adalah klausul yang valid, tetapi tidak dengan `FROM S3Object.`*`path`*.

Example

Contoh:

Contoh 1

Contoh ini menampilkan hasil menggunakan set data dan kueri berikut:

```
{ "Rules": [ {"id": "1"}, {"expr": "y > x"}, {"id": "2", "expr": "z = DEBUG"} ]}
{ "created": "June 27", "modified": "July 6" }
```

```
SELECT id FROM S3Object[*].Rules[*].id
```

```
{"id":"1"}
{}
{"id":"2"}
{}
```

Amazon S3 Select menghasilkan setiap hasil karena alasan berikut:

- `{"id":"id-1"}`—`S3Object[0].Rules[0].id` menghasilkan kecocokan.
- `{}`—`S3Object[0].Rules[1].id` tidak cocok dengan catatan, jadi Amazon S3 Select mengeluarkan `MISSING`, yang kemudian diubah menjadi catatan kosong selama serialisasi output dan dikembalikan.

- `{"id":"id-2"}`–`S3Object[0].Rules[2].id` menghasilkan kecocokan.
- `{}`–`S3Object[1]` tidak cocok dengan `Rules`, jadi Amazon S3 Select mengeluarkan `MISSING`, yang kemudian diubah menjadi catatan kosong selama serialisasi output dan dikembalikan.

Apabila Anda tidak ingin Amazon S3 Select mengembalikan catatan kosong saat tidak menemukan kecocokan, Anda dapat menguji nilai `MISSING`. Kueri berikut mengembalikan hasil yang sama seperti kueri sebelumnya, tetapi nilai yang kosong dihilangkan:

```
SELECT id FROM S3Object[*].Rules[*].id WHERE id IS NOT MISSING
```

```
{"id":"1"}
{"id":"2"}
```

Contoh 2

Contoh ini menampilkan hasil menggunakan set data dan kueri berikut:

```
{ "created": "936864000", "dir_name": "important_docs", "files": [ { "name": "." },
 { "name": ".." }, { "name": ".aws" }, { "name": "downloads" } ], "owner": "Amazon
 S3" }
{ "created": "936864000", "dir_name": "other_docs", "files": [ { "name": "." },
 { "name": ".." }, { "name": "my stuff" }, { "name": "backup" } ], "owner": "User" }
```

```
SELECT d.dir_name, d.files FROM S3Object[*] d
```

```
{"dir_name":"important_docs","files":[{"name":"."},{"name":".."},{"name":".aws"},
{"name":"downloads"}]}
{"dir_name":"other_docs","files":[{"name":"."},{"name":".."},{"name":"my stuff"},
{"name":"backup"}]}
```

```
SELECT _1.dir_name, _1.owner FROM S3Object[*]
```

```
{"dir_name":"important_docs","owner":"Amazon S3"}
{"dir_name":"other_docs","owner":"User"}
```

Klausul WHERE

Klausul `WHERE` mengikuti sintaks ini:

```
WHERE condition
```

Klausul WHERE memfilter baris berdasarkan *condition*. Ketentuan adalah ekspresi yang memiliki hasil Boolean. Hanya baris yang syaratnya dievaluasi sebagai TRUE yang dikembalikan di hasilnya.

Klausul LIMIT

Klausul LIMIT mengikuti sintaks ini:

```
LIMIT number
```

Klausul LIMIT membatasi jumlah catatan yang Anda ingin kuerinya dikembalikan berdasarkan *number*.

Akses atribut

Klausul SELECT dan WHERE dapat merujuk ke data catatan menggunakan salah satu metode dalam bagian berikut, tergantung pada jika file yang dicari dalam format CSV atau JSON.

CSV

- Nomor Kolom–Anda dapat merujuk ke kolom Nth pada baris dengan nama kolom _*N*, dengan *N* adalah posisi kolom. Hitungan posisi dimulai pada 1. Misalnya, kolom pertama diberi nama _1 dan kolom kedua diberi nama _2.

  Anda dapat merujuk ke kolom sebagai _*N* atau *alias*._*N*. Misalnya, _2 dan myAlias._2 adalah cara yang valid untuk merujuk ke kolom dalam daftar SELECT dan klausul WHERE.
- Header Kolom–Untuk objek dalam format CSV yang memiliki baris header, header tersedia untuk daftar SELECT dan klausul WHERE. Secara khusus, seperti dalam SQL tradisional, dalam ekspresi klausul SELECT dan WHERE, Anda dapat melihat kolom dengan *alias*.*column_name* atau *column_name*.

JSON

- Dokumen–Anda dapat mengakses bidang dokumen JSON sebagai *alias*.*name*. Anda juga dapat mengakses bidang bersarang, misalnya, *alias*.*name1*.*name2*.*name3*.
- Daftar–Anda dapat mengakses elemen dalam daftar JSON menggunakan indeks berbasis nol dengan operator []. Misalnya, Anda dapat mengakses elemen kedua dari daftar sebagai

alias[1]. Anda dapat menggabungkan elemen daftar yang diakses dengan bidang, misalnya, *alias*.*name1*.*name2*[1].*name3*.

- Contoh: Pertimbangkan objek JSON ini sebagai sampel set data:

```
{"name": "Susan Smith",
"org": "engineering",
"projects":
    [
     {"project_name":"project1", "completed":false},
     {"project_name":"project2", "completed":true}
    ]
}
```

Contoh 1

Kueri berikut mengembalikan hasil ini:

```
Select s.name from S3Object s
```

```
{"name":"Susan Smith"}
```

Contoh 2

Kueri berikut mengembalikan hasil ini:

```
Select s.projects[0].project_name from S3Object s
```

```
{"project_name":"project1"}
```

Sensitivitas huruf besar/kecil pada nama header dan atribut

Dengan Amazon S3 Select, Anda dapat menggunakan tanda petik dua untuk menunjukkan header kolom (untuk objek CSV) dan atribut (untuk objek JSON) yang peka terhadap huruf besar/kecil. Tanpa tanda petik dua, header dan atribut objek tidak sensitif terhadap huruf besar/kecil. Kesalahan akan muncul dalam kasus ambiguitas.

Contoh berikut adalah 1) objek Amazon S3 dalam format CSV dengan header bidang yang ditentukan, dan dengan `FileHeaderInfo` diatur ke `"Use"` untuk permintaan kueri; atau 2) objek Amazon S3 dalam format JSON dengan atribut yang ditentukan.

Contoh 1: Objek yang dikuerikan memiliki header atau atribut `NAME`.

- Ekspresi berikut berhasil mengembalikan nilai dari objek. Karena tidak ada tanda petik, kueri tidak peka huruf besar/kecil.

```
SELECT s.name from S3Object s
```

- Ekspresi berikut menghasilkan kesalahan 400 `MissingHeaderName`. Karena terdapat tanda petik, kueri peka huruf besar/kecil.

```
SELECT s."name" from S3Object s
```

Contoh 2: Objek Amazon S3 yang dikuerikan memiliki satu header atau atribut dengan `NAME` header atau atribut lainnya dengan `name`.

- Ekspresi berikut menghasilkan kesalahan 400 `AmbiguousFieldName`. Karena tidak ada tanda petik, kueri tidak sensitif terhadap huruf besar/kecil, tetapi terdapat dua kecocokan, jadi kesalahan dilemparkan.

```
SELECT s.name from S3Object s
```

- Ekspresi berikut berhasil mengembalikan nilai dari objek. Karena terdapat tanda petik, kueri peka huruf besar/kecil, sehingga tidak ada ambiguitas.

```
SELECT s."NAME" from S3Object s
```

Menggunakan kata kunci terpesan sebagai istilah yang ditetapkan pengguna

Amazon S3 Select memiliki serangkaian kata kunci terpesan yang diperlukan untuk menjalankan ekspresi SQL yang digunakan untuk mengkueri konten objek. Kata kunci terpesan meliputi nama fungsi, jenis data, operator, dan sebagainya. Dalam beberapa kasus, istilah yang ditentukan pengguna, seperti header kolom (untuk file CSV) atau atribut (untuk objek JSON), dapat bertentangan dengan kata kunci terpesan. Pada saat hal ini terjadi, Anda harus menggunakan tanda

petik dua untuk menunjukkan bahwa Anda sengaja menggunakan istilah khusus pengguna yang bertentangan dengan kata kunci terpesan. Apabila tidak, kesalahan penguraian 400 akan terjadi.

Untuk mengetahui daftar lengkap kata kunci terpesan, lihat Kata kunci terpesan.

Contoh berikut adalah 1) objek Amazon S3 dalam format CSV dengan header bidang yang ditentukan, dengan `FileHeaderInfo` diatur ke `"Use"` untuk permintaan kueri, atau 2) objek Amazon S3 dalam format JSON dengan atribut yang ditentukan.

Contoh: Objek yang dikuerikan memiliki header atau atribut bernama `CAST`, yang merupakan kata kunci terpesan.

- Ekspresi berikut berhasil mengembalikan nilai dari objek. Karena tanda petik digunakan dalam kueri, S3 Select menggunakan header atau atribut yang ditentukan pengguna.

```
SELECT s."CAST" from S3Object s
```

- Ekspresi berikut menghasilkan kesalahan penguraian 400. Karena tidak ada tanda petik yang digunakan dalam kueri, `CAST` bertentangan dengan kata kunci terpesan.

```
SELECT s.CAST from S3Object s
```

Ekspresi skalar

Dalam klausul `WHERE` dan daftar `SELECT`, Anda dapat memiliki ekspresi skalar SQL, yaitu ekspresi yang mengembalikan nilai skalar. Bentuknya adalah sebagai berikut:

- ***literal***

  SQL literal.

- ***column_reference***

  Referensi ke kolom berupa *`column_name`* atau *`alias`*`.`*`column_name`*.

- ***unary_op expression***

  Dalam hal ini, ***unary_op*** adalah operator numerik SQL.

- ***expression binary_op expression***

  Dalam hal ini, ***binary_op*** adalah operator biner SQL.

- ***func_name***

  Dalam hal ini, ***func_name*** adalah nama fungsi skalar yang diminta.
- ***expression*** `[ NOT ]` `BETWEEN` ***expression*** `AND` ***expression***
- ***expression*** `LIKE` ***expression*** `[ ESCAPE` ***expression*** `]`

Jenis data

> **Important**
>
> Amazon S3 Select tidak lagi tersedia untuk pelanggan baru. Pelanggan Amazon S3 Select yang sudah ada dapat terus menggunakan fitur seperti biasa. Pelajari selengkapnya

Amazon S3 Select mendukung beberapa jenis data primitif.

Konversi jenis data

Aturan umumnya adalah mengikuti fungsi `CAST` jika ditentukan. Apabila `CAST` tidak ditentukan, semua data input diperlakukan sebagai string. Dalam hal ini, Anda harus memasukkan data input Anda ke dalam jenis data yang relevan jika perlu.

Untuk informasi selengkapnya tentang fungsi `CAST`, lihat CAST.

Jenis data yang didukung

Amazon S3 Select mendukung rangkaian jenis data primitif berikut.

Nama	Deskripsi	Contoh
`bool`	Nilai Boolean, baik `TRUE` ataupun `FALSE`.	`FALSE`
`int`, `integer`	Bilangan bulat bertanda 8-byte dalam rentang -9.223.37 2.036.854.775.808 hingga 9.223.372.036.854.775.807.	`100000`
`string`	String panjang UTF8 variabel yang dikodekan. Batas default adalah 1 karakter. Batas karakter maksimum adalah 2.147.483.647.	`'xyz'`

Nama	Deskripsi	Contoh
`float`	Angka floating point 8-byte.	`CAST(0.456 AS FLOAT)`
`decimal`, `numeric`	Angka dasar-10, dengan presisi maksimum 38 (yaitu, jumlah maksimal digit signifikan), dan dengan skala dalam rentang -2^{31} hingga $2^{31}-1$ (yaitu, eksponen dasar-10). **Note** Amazon S3 Select mengabaikan skala dan presisi saat Anda memberikan keduanya secara bersamaan.	`123.456`
`timestamp`	Timestamp mewakili momen waktu tertentu, selalu menyertakan offset lokal, dan mampu memiliki presisi arbitrer. Dalam format teks, timestamp mengikuti catatan W3C tentang format tanggal dan waktu, tetapi harus mengakhir inya dengan timestamp T setidaknya presisi sehari penuh. Beberapa detik fraksional diperbolehkan, dengan setidaknya satu digit presisi, dan maksimum tidak terbatas. Offset waktu lokal dapat dinyatakan sebagai offset jam:menit dari UTC, atau sebagai `Z` literal untuk menunjukkan waktu lokal UTC. Offset waktu lokal diperlukan pada timestamp dengan waktu dan tidak diizinkan pada nilai tanggal.	`CAST('2007-04-05T14:30Z' AS TIMESTAMP)`

Jenis Parquet yang didukung

Amazon S3 Select mendukung jenis Parquet berikut.

- `DATE`
- `DECIMAL`
- `ENUM`

- `INT(8)`
- `INT(16)`
- `INT(32)`
- `INT(64)`
- `LIST`

> **Note**
>
> Untuk output jenis `LIST` Parquet, Amazon S3 Select hanya mendukung format JSON. Namun, jika kueri membatasi data ke nilai sederhana, jenis `LIST` Parquet juga dapat dikuerikan dalam format CSV.

- `STRING`
- Presisi `TIMESTAMP` yang didukung (`MILLIS`/`MICROS`/`NANOS`)

> **Note**
>
> Timestamp yang disimpan sebagai `INT(96)` tidak didukung.
> Karena rentang jenis `INT(64)`, timestamp yang menggunakan unit `NANOS` hanya dapat mewakili nilai antara `1677-09-21 00:12:43` dan `2262-04-11 23:47:16`. Nilai di luar rentang ini tidak dapat diwakili dengan unit `NANOS`.

Pemetaan jenis Parquet ke jenis data yang didukung di Amazon S3 Select

Jenis Parquet	Jenis data yang didukung
`DATE`	`timestamp`
`DECIMAL`	`decimal`, `numeric`
`ENUM`	`string`
`INT(8)`	`int`, `integer`

Jenis Parquet	Jenis data yang didukung
INT(16)	int, integer
INT(32)	int, integer
INT(64)	decimal, numeric
LIST	Setiap jenis Parquet dalam daftar dipetakan ke jenis data yang sesuai.
STRING	string
TIMESTAMP	timestamp

Operator

> **Important**
>
> Amazon S3 Select tidak lagi tersedia untuk pelanggan baru. Pelanggan Amazon S3 Select yang sudah ada dapat terus menggunakan fitur seperti biasa. Pelajari selengkapnya

Amazon S3 Select mendukung operator berikut.

Operator logis

- AND
- NOT
- OR

Operator perbandingan

- <
- >

- `<=`
- `>=`
- `=`
- `<>`
- `!=`
- `BETWEEN`
- `IN`–Sebagai contoh: `IN ('a', 'b', 'c')`

Operator pencocokan pola

- `LIKE`
- `_` (Cocok untuk setiap karakter)
- `%` (Cocok untuk urutan karakter apa pun)

Operator numerik

- `IS NULL`
- `IS NOT NULL`

Operator matematika

Penambahan, pengurangan, perkalian, pembagian, dan modulo didukung, sebagai berikut:

- `+`
- `-`
- `*`
- `/`
- `%`

Prasyarat operator

Tabel berikut menunjukkan prasyarat operator dalam urutan menurun.

Operator atau elemen	Asosiatif	Diperlukan
-	kanan	numerik minus
*, /, %	kiri	perkalian, pembagian, modulo
+, -	kiri	penambahan, pengurangan
IN		atur keanggotaan
BETWEEN		pengurungan rentang
LIKE		pencocokan pola string
<>		kurang dari, lebih dari
=	kanan	kesetaraan, penugasan
NOT	kanan	negasi logis
AND	kiri	konjungsi logis
OR	kiri	disjungsi logis

Kata kunci terpesan

> **Important**
>
> Amazon S3 Select tidak lagi tersedia untuk pelanggan baru. Pelanggan Amazon S3 Select yang sudah ada dapat terus menggunakan fitur seperti biasa. Pelajari selengkapnya

Berikut adalah daftar kata kunci terpesan untuk Amazon S3 Select. Kata kunci ini mencakup nama fungsi, jenis data, operator, dan sebagainya, yang diperlukan untuk menjalankan ekspresi SQL yang digunakan untuk melakukan kueri konten objek.

```
absolute
action
add
all
allocate
alter
and
any
are
as
asc
assertion
at
authorization
avg
bag
begin
between
bit
bit_length
blob
bool
boolean
both
by
cascade
cascaded
case
cast
catalog
```

```
char
char_length
character
character_length
check
clob
close
coalesce
collate
collation
column
commit
connect
connection
constraint
constraints
continue
convert
corresponding
count
create
cross
current
current_date
current_time
current_timestamp
current_user
cursor
date
day
deallocate
dec
decimal
declare
default
deferrable
deferred
delete
desc
describe
descriptor
diagnostics
disconnect
distinct
```

```
domain
double
drop
else
end
end-exec
escape
except
exception
exec
execute
exists
external
extract
false
fetch
first
float
for
foreign
found
from
full
get
global
go
goto
grant
group
having
hour
identity
immediate
in
indicator
initially
inner
input
insensitive
insert
int
integer
intersect
interval
```

```
into
is
isolation
join
key
language
last
leading
left
level
like
limit
list
local
lower
match
max
min
minute
missing
module
month
names
national
natural
nchar
next
no
not
null
nullif
numeric
octet_length
of
on
only
open
option
or
order
outer
output
overlaps
pad
```

```
partial
pivot
position
precision
prepare
preserve
primary
prior
privileges
procedure
public
read
real
references
relative
restrict
revoke
right
rollback
rows
schema
scroll
second
section
select
session
session_user
set
sexp
size
smallint
some
space
sql
sqlcode
sqlerror
sqlstate
string
struct
substring
sum
symbol
system_user
table
```

```
temporary
then
time
timestamp
timezone_hour
timezone_minute
to
trailing
transaction
translate
translation
trim
true
tuple
union
unique
unknown
unpivot
update
upper
usage
user
using
value
values
varchar
varying
view
when
whenever
where
with
work
write
year
zone
```

Fungsi SQL

> **Important**
>
> Amazon S3 Select tidak lagi tersedia untuk pelanggan baru. Pelanggan Amazon S3 Select yang sudah ada dapat terus menggunakan fitur seperti biasa. Pelajari selengkapnya

Amazon S3 Select mendukung fungsi SQL berikut.

Topik

- Fungsi agregat
- Fungsi kondisional
- Fungsi konversi
- Fungsi tanggal
- Fungsi string

Fungsi agregat

> **Important**
>
> Amazon S3 Select tidak lagi tersedia untuk pelanggan baru. Pelanggan Amazon S3 Select yang sudah ada dapat terus menggunakan fitur seperti biasa. Pelajari selengkapnya

Amazon S3 Select mendukung fungsi agregat berikut.

Fungsi	Jenis Argumen	Jenis pengembalian
`AVG(`*`expressio`* *`n`* `)`	`INT`, `FLOAT`, `DECIMAL`	`DECIMAL` untuk argumen `INT`, `FLOAT` untuk argumen floating point; jika tidak, sama

Fungsi	Jenis Argumen	Jenis pengembalian
		dengan jenis data argumen.
`COUNT`	-	`INT`
`MAX(`*expression*`)`	`INT`, `DECIMAL`	Sama seperti jenis argumen.
`MIN(`*expression*`)`	`INT`, `DECIMAL`	Sama seperti jenis argumen.
`SUM(`*expression*`)`	`INT`, `FLOAT`, `DOUBLE`, `DECIMAL`	`INT` untuk argumen `INT`, `FLOAT` untuk argumen floating point; jika tidak, sama dengan jenis data argumen.

Contoh SUM

Untuk menggabungkan ukuran objek total suatu folder dalam laporan Inventaris S3, gunakan ekspresi `SUM`.

Laporan Inventaris S3 berikut adalah file CSV yang dikompresi dengan GZIP. Terdapat tiga kolom.

- Kolom pertama adalah nama bucket S3 (*`DOC-EXAMPLE-BUCKET`*) yang digunakan untuk laporan Inventaris S3.
- Kolom kedua adalah nama kunci objek yang secara unik mengidentifikasi objek dalam bucket.

  Nilai *`example-folder/`* di baris pertama adalah untuk folder *`example-folder`*. Di Amazon S3, saat Anda membuat folder di bucket, S3 membuat objek 0-byte dengan kunci yang diatur ke nama folder yang Anda berikan.

Nilai *example-folder/object1* di baris kedua adalah untuk objek *object1* di folder *example-folder*.

Nilai *example-folder/object2* di baris ketiga adalah untuk objek *object2* di folder *example-folder*.

Untuk informasi selengkapnya tentang folder S3, lihat Mengatur objek di konsol Amazon S3 dengan menggunakan folder.

- Kolom ketiga adalah ukuran objek dalam byte.

```
"DOC-EXAMPLE-BUCKET","example-folder/","0"
"DOC-EXAMPLE-BUCKET","example-folder/object1","2011267"
"DOC-EXAMPLE-BUCKET","example-folder/object2","1570024"
```

Untuk menggunakan ekspresi SUM guna menghitung ukuran total folder *example-folder*, jalankan kueri SQL dengan Amazon S3 Select.

```
SELECT SUM(CAST(_3 as INT)) FROM s3object s WHERE _2 LIKE 'example-folder/%' AND _2 != 
 'example-folder/';
```

Hasil Kueri:

```
3581291
```

Fungsi kondisional

> **Important**
>
> Amazon S3 Select tidak lagi tersedia untuk pelanggan baru. Pelanggan Amazon S3 Select yang sudah ada dapat terus menggunakan fitur seperti biasa. Pelajari selengkapnya

Amazon S3 Select mendukung fungsi kondisional berikut.

Topik

- CASE
- COALESCE

- NULLIF

CASE

Ekspresi CASE adalah ekspresi kondisional yang serupa dengan pernyataan `if/then/else` yang ditemukan dalam bahasa lain. CASEdigunakan untuk menentukan hasil jika terdapat beberapa kondisi. Terdapat dua jenis ekspresi CASE: sederhana dan dicari.

Dalam ekspresi CASE sederhana, ekspresi dibandingkan dengan nilai. Ketika kecocokan ditemukan, tindakan yang ditentukan dalam klausul THEN diterapkan. Apabila kecocokan tidak ditemukan, tindakan dalam klausul ELSE diterapkan.

Dalam ekspresi CASE yang dicari, masing-masing CASE dievaluasi berdasarkan ekspresi Boolean, dan pernyataan CASE mengembalikan kecocokan pertama CASE. Jika tidak ada kecocokan CASE yang ditemukan di antara klausul WHEN, tindakan dalam klausul ELSE dikembalikan.

Sintaks

> **Note**
>
> Saat ini, Amazon S3 Select tidak mendukung ORDER BY atau kueri yang berisi baris baru. Pastikan Anda menggunakan kueri tanpa jeda baris.

Berikut ini adalah pernyataan CASE sederhana yang digunakan untuk mencocokkan kondisi:

```
CASE expression WHEN value THEN result [WHEN...] [ELSE result] END
```

Berikut ini adalah pernyataan CASE yang dicari yang digunakan untuk mengevaluasi setiap kondisi:

```
CASE WHEN boolean condition THEN result [WHEN ...] [ELSE result] END
```

Contoh

> **Note**
>
> Apabila Anda menggunakan konsol Amazon S3 untuk menjalankan contoh berikut dan file CSV berisi baris header, pilih Kecualikan baris pertama data CSV.

Contoh 1: Gunakan ekspresi CASE sederhana untuk mengganti `New York City` dengan `Big Apple` dalam kueri. Ganti semua nama kota lainnya dengan `other`.

```
SELECT venuecity, CASE venuecity WHEN 'New York City' THEN 'Big Apple' ELSE 'other' END
 FROM S3Object;
```

Hasil kueri:

```
venuecity        |   case
-----------------+-----------
Los Angeles      | other
New York City    | Big Apple
San Francisco    | other
Baltimore        | other
...
```

Contoh 2: Gunakan ekspresi CASE yang dicari untuk menetapkan nomor grup berdasarkan nilai `pricepaid` penjualan tiket individu:

```
SELECT pricepaid, CASE WHEN CAST(pricepaid as FLOAT) < 10000 THEN 'group 1' WHEN
 CAST(pricepaid as FLOAT) > 10000 THEN 'group 2' ELSE 'group 3' END FROM S3Object;
```

Hasil kueri:

```
pricepaid |  case
-----------+---------
12624.00 | group 2
10000.00 | group 3
10000.00 | group 3
9996.00 | group 1
9988.00 | group 1
...
```

COALESCE

`COALESCE` mengevaluasi argumen secara berurutan dan mengembalikan nilai non-tidak diketahui pertama, yaitu, nilai non-null atau non-hilang yang pertama. Fungsi ini tidak memperbanyak nilai null dan hilang.

Sintaks

```
COALESCE ( expression, expression, ... )
```

Parameter

expression

Ekspresi target tempat fungsi beroperasi.

Contoh

```
COALESCE(1)               -- 1
COALESCE(null)            -- null
COALESCE(null, null)      -- null
COALESCE(missing)         -- null
COALESCE(missing, missing) -- null
COALESCE(1, null)         -- 1
COALESCE(null, null, 1)   -- 1
COALESCE(null, 'string')  -- 'string'
COALESCE(missing, 1)      -- 1
```

NULLIF

Dengan dua ekspresi, `NULLIF` mengembalikan `NULL` jika kedua ekspresi tersebut mengevaluasi nilai yang sama; jika tidak, `NULLIF` mengembalikan hasil evaluasi ke ekspresi yang pertama.

Sintaks

```
NULLIF ( expression1, expression2 )
```

Parameter

expression1, *expression2*

Ekspresi target tempat fungsi beroperasi.

Contoh

```
NULLIF(1, 1)            -- null
NULLIF(1, 2)            -- 1
```

```
NULLIF(1.0, 1)           -- null
NULLIF(1, '1')           -- 1
NULLIF([1], [1])         -- null
NULLIF(1, NULL)          -- 1
NULLIF(NULL, 1)          -- null
NULLIF(null, null)       -- null
NULLIF(missing, null)    -- null
NULLIF(missing, missing) -- null
```

Fungsi konversi

> **Important**
>
> Amazon S3 Select tidak lagi tersedia untuk pelanggan baru. Pelanggan Amazon S3 Select yang sudah ada dapat terus menggunakan fitur seperti biasa. Pelajari selengkapnya

Amazon S3 Select mendukung fungsi konversi berikut.

Topik

- CAST

CAST

Fungsi CAST mengonversi suatu entitas, seperti ekspresi yang mengevaluasi ke nilai tunggal, dari satu jenis ke lainnya.

Sintaks

```
CAST ( expression AS data_type )
```

Parameter

expression

Kombinasi satu atau beberapa nilai, operator, dan fungsi SQL yang mengevaluasi nilai.

data_type

Jenis data target, seperti INT, untuk menghadirkan ekspresi. Untuk daftar jenis data yang didukung, lihat Jenis data.

Contoh

```
CAST('2007-04-05T14:30Z' AS TIMESTAMP)
CAST(0.456 AS FLOAT)
```

Fungsi tanggal

> **Important**
>
> Amazon S3 Select tidak lagi tersedia untuk pelanggan baru. Pelanggan Amazon S3 Select yang sudah ada dapat terus menggunakan fitur seperti biasa. Pelajari selengkapnya

Amazon S3 Select mendukung fungsi tanggal berikut.

Topik

- DATE_ADD
- DATE_DIFF
- EXTRACT
- TO_STRING
- TO_TIMESTAMP
- UTCNOW

DATE_ADD

Dengan bagian tanggal, kuantitas, dan timestamp, DATE_ADD mengembalikan timestamp yang diperbarui dengan mengubah bagian tanggal berdasarkan kuantitas.

Sintaks

```
DATE_ADD( date_part, quantity, timestamp )
```

Parameter

date_part

Menentukan bagian dari tanggal untuk diubah. Ini dapat berupa salah satu dari berikut ini:

- tahun

- bulan
- hari
- jam
- menit
- detik

quantity

Nilai yang diterapkan pada timestamp yang diperbarui. Nilai positif untuk *quantity* ditambahkan ke date_part timestamp, dan pengurangan nilai negatif.

timestamp

Timestamp target tempat fungsi beroperasi.

Contoh

```
DATE_ADD(year, 5, `2010-01-01T`)                -- 2015-01-01 (equivalent to
 2015-01-01T)
DATE_ADD(month, 1, `2010T`)                     -- 2010-02T (result will add precision
 as necessary)
DATE_ADD(month, 13, `2010T`)                    -- 2011-02T
DATE_ADD(day, -1, `2017-01-10T`)                -- 2017-01-09 (equivalent to
 2017-01-09T)
DATE_ADD(hour, 1, `2017T`)                      -- 2017-01-01T01:00-00:00
DATE_ADD(hour, 1, `2017-01-02T03:04Z`)          -- 2017-01-02T04:04Z
DATE_ADD(minute, 1, `2017-01-02T03:04:05.006Z`) -- 2017-01-02T03:05:05.006Z
DATE_ADD(second, 1, `2017-01-02T03:04:05.006Z`) -- 2017-01-02T03:04:06.006Z
```

DATE_DIFF

Dengan bagian tanggal dan dua timestamp yang valid, DATE_DIFF mengembalikan selisih bagian tanggal. Nilai yang dikembalikan adalah bilangan bulat negatif ketika nilai *date_part* dari *timestamp1* lebih dari nilai *date_part* dari *timestamp2*. Nilai yang dikembalikan adalah bilangan bulat positif ketika nilai *date_part* dari *timestamp1* kurang dari nilai *date_part* dari *timestamp2*.

Sintaks

```
DATE_DIFF( date_part, timestamp1, timestamp2 )
```

Parameter

date_part

Menentukan bagian mana dari timestamp untuk dibandingkan. Untuk definisi `date_part`, lihat DATE_ADD.

timestamp1

Timestamp pertama untuk dibandingkan.

timestamp2

Timestamp kedua untuk dibandingkan.

Contoh

```
DATE_DIFF(year, `2010-01-01T`, `2011-01-01T`)           -- 1
DATE_DIFF(year, `2010T`, `2010-05T`)                    -- 4 (2010T is equivalent to
 2010-01-01T00:00:00.000Z)
DATE_DIFF(month, `2010T`, `2011T`)                      -- 12
DATE_DIFF(month, `2011T`, `2010T`)                      -- -12
DATE_DIFF(day, `2010-01-01T23:00`, `2010-01-02T01:00`) -- 0 (need to be at least 24h
 apart to be 1 day apart)
```

EXTRACT

Dengan bagian tanggal dan timestamp, EXTRACT mengembalikan nilai bagian tanggal timestamp.

Sintaks

```
EXTRACT( date_part FROM timestamp )
```

Parameter

date_part

Menentukan bagian dari timestamp untuk diekstrak. Ini dapat berupa salah satu dari berikut ini:

- YEAR
- MONTH
- DAY

- HOUR
- MINUTE
- SECOND
- TIMEZONE_HOUR
- TIMEZONE_MINUTE

timestamp

Timestamp target tempat fungsi beroperasi.

Contoh

```
EXTRACT(YEAR FROM `2010-01-01T`)                              -- 2010
EXTRACT(MONTH FROM `2010T`)                                   -- 1 (equivalent to
 2010-01-01T00:00:00.000Z)
EXTRACT(MONTH FROM `2010-10T`)                                -- 10
EXTRACT(HOUR FROM `2017-01-02T03:04:05+07:08`)                -- 3
EXTRACT(MINUTE FROM `2017-01-02T03:04:05+07:08`)              -- 4
EXTRACT(TIMEZONE_HOUR FROM `2017-01-02T03:04:05+07:08`)       -- 7
EXTRACT(TIMEZONE_MINUTE FROM `2017-01-02T03:04:05+07:08`)     -- 8
```

TO_STRING

Dengan timestamp dan pola format, TO_STRING mengembalikan representasi string timestamp dalam format yang diberikan.

Sintaks

```
TO_STRING ( timestamp time_format_pattern )
```

Parameter

timestamp

Timestamp target tempat fungsi beroperasi.

time_format_pattern

String yang memiliki interpretasi karakter khusus berikut:

format	Contoh	Deskripsi
yy	69	2 digit tahun
y	1969	4 digit tahun
yyyy	1969	4 digit tahun yang ditambahkan nol
M	1	Bulan dalam tahun
MM	01	Bulan dalam tahun yang ditambahkan nol
MMM	Jan	Singkatan nama bulan dalam tahun
MMMM	January	Nama bulan lengkap dalam tahun
MMMMM	J	Huruf pertama bulan dalam tahun (CATATAN: Format ini tidak valid untuk digunakan dengan fungsi TO_TIMESTAMP .)

format	Contoh	Deskripsi
d	2	Hari dalam bulan (1-31)
dd	02	Hari pada bulan yang ditambahkan nol (01-31)
a	AM	AM atau PM pada hari
h	3	Jam dalam hari (1-12)
hh	03	Jam dalam hari yang ditambahkan nol (01-12)
H	3	Jam dalam hari (0-23)
HH	03	Jam dalam hari yang ditambahkan nol (00-23)
m	4	Menit dalam jam (0-59)
mm	04	Menit dalam jam yang ditambahkan nol (00-59)
s	5	Detik dalam menit (0-59)

format	Contoh	Deskripsi
ss	05	Detik dalam menit yang ditambahkan nol (00-59)
S	0	Fraksi detik (presisi: 0,1, rentang: 0,0-0,9)
SS	6	Fraksi detik (presisi: 0,01, rentang: 0,0-0,99)
SSS	60	Fraksi detik (presisi: 0,001, rentang: 0,0-0,999)
…	…	…
SSSSSSSSS	60000000	Fraksi detik (presisi maksimum: 1 nanodetik, rentang: 0,0-0,999 999999)
n	60000000	Nano dari detik

format	Contoh	Deskripsi
X	+07 atau Z	Offset dalam jam, atau Z jika offset adalah 0
XX atau XXXX	+0700 atau Z	Offset dalam jam dan menit, atau Z jika offset adalah 0
XXX atau XXXXX	+07:00 atau Z	Offset dalam jam dan menit, atau Z jika offset adalah 0
x	7	Offset dalam jam
xx atau xxxx	700	Offset dalam jam dan menit
xxx atau xxxxx	+07:00	Offset dalam jam dan menit

Contoh

```
TO_STRING(`1969-07-20T20:18Z`,  'MMMM d, y')                    -- "July 20, 1969"
TO_STRING(`1969-07-20T20:18Z`, 'MMM d, yyyy')                   -- "Jul 20, 1969"
TO_STRING(`1969-07-20T20:18Z`, 'M-d-yy')                        -- "7-20-69"
TO_STRING(`1969-07-20T20:18Z`, 'MM-d-y')                        -- "07-20-1969"
TO_STRING(`1969-07-20T20:18Z`, 'MMMM d, y h:m a')               -- "July 20, 1969 8:18
 PM"
TO_STRING(`1969-07-20T20:18Z`, 'y-MM-dd''T''H:m:ssX')           --
 "1969-07-20T20:18:00Z"
TO_STRING(`1969-07-20T20:18+08:00Z`, 'y-MM-dd''T''H:m:ssX')     --
 "1969-07-20T20:18:00Z"
```

```
TO_STRING(`1969-07-20T20:18+08:00`, 'y-MM-dd''T''H:m:ssXXXX')   --
 "1969-07-20T20:18:00+0800"
TO_STRING(`1969-07-20T20:18+08:00`, 'y-MM-dd''T''H:m:ssXXXXX')  --
 "1969-07-20T20:18:00+08:00"
```

TO_TIMESTAMP

Dengan string, `TO_TIMESTAMP` mengonversi menjadi timestamp. `TO_TIMESTAMP` adalah operasi terbalik `TO_STRING`.

Sintaks

```
TO_TIMESTAMP ( string )
```

Parameter

`string`

String target tempat fungsi beroperasi.

Contoh

```
TO_TIMESTAMP('2007T')                          -- `2007T`
TO_TIMESTAMP('2007-02-23T12:14:33.079-08:00') -- `2007-02-23T12:14:33.079-08:00`
```

UTCNOW

`UTCNOW` mengembalikan waktu saat ini dalam UTC sebagai timestamp.

Sintaks

```
UTCNOW()
```

Parameter

`UTCNOW` tidak menggunakan parameter.

Contoh

```
UTCNOW() -- 2017-10-13T16:02:11.123Z
```

Fungsi string

> **Important**
>
> Amazon S3 Select tidak lagi tersedia untuk pelanggan baru. Pelanggan Amazon S3 Select yang sudah ada dapat terus menggunakan fitur seperti biasa. Pelajari selengkapnya

Amazon S3 Select mendukung fungsi string berikut.

Topik

- CHAR_LENGTH, CHARACTER_LENGTH
- LOWER
- SUBSTRING
- TRIM
- UPPER

CHAR_LENGTH, CHARACTER_LENGTH

`CHAR_LENGTH` (atau`CHARACTER_LENGTH`) menghitung jumlah karakter dalam string yang ditentukan.

> **Note**
>
> `CHAR_LENGTH` dan `CHARACTER_LENGTH` adalah sinonim.

Sintaks

```
CHAR_LENGTH ( string )
```

Parameter

`string`

String target tempat fungsi beroperasi.

Contoh

```
CHAR_LENGTH('')          -- 0
CHAR_LENGTH('abcdefg')   -- 7
```

LOWER

Dengan string, LOWER mengubah semua karakter huruf besar menjadi karakter huruf kecil. Setiap karakter yang tidak dalam huruf besar tidak berubah.

Sintaks

```
LOWER ( string )
```

Parameter

string

String target tempat fungsi beroperasi.

Contoh

```
LOWER('AbCdEfG!@#$') -- 'abcdefg!@#$'
```

SUBSTRING

Dengan string, indeks awal, dan secara opsional panjang, SUBSTRINGmengembalikan substring dari indeks awal hingga ujung string, atau hingga panjang yang disediakan.

> **Note**
>
> Karakter pertama dari string input memiliki posisi indeks 1.
>
> - Jika `start` < 1, tanpa panjang yang ditentukan, posisi indeks diatur ke 1.
> - Jika `start` < 1, dengan panjang yang ditentukan, posisi indeks diatur ke `start + length -1`.
> - Jika `start + length -1` < 0, string kosong dikembalikan.
> - Jika `start + length -1` > = 0, substring dimulai pada posisi indeks 1 dengan panjang `start + length - 1` dikembalikan.

Sintaks

```
SUBSTRING( string FROM start [ FOR length ] )
```

Parameter

string

String target tempat fungsi beroperasi.

start

Posisi awal string.

length

Panjang substring untuk dikembalikan. Apabila tidak ada, lanjutkan ke akhir string.

Contoh

```
SUBSTRING("123456789", 0)      -- "123456789"
SUBSTRING("123456789", 1)      -- "123456789"
SUBSTRING("123456789", 2)      -- "23456789"
SUBSTRING("123456789", -4)     -- "123456789"
SUBSTRING("123456789", 0, 999) -- "123456789"
SUBSTRING("123456789", 1, 5)   -- "12345"
```

TRIM

Memangkas karakter di depan atau di belakang string. Karakter default yang harus dihapus adalah spasi (' ').

Sintaks

```
TRIM ( [[LEADING | TRAILING | BOTH remove_chars] FROM] string )
```

Parameter

string

String target tempat fungsi beroperasi.

`LEADING | TRAILING | BOTH`

Parameter ini menunjukkan jika akan memangkas karakter di depan atau di belakang, atau karakter di depan dan di belakang.

`remove_chars`

Set karakter untuk dihapus. *`remove_chars`* dapat berupa string dengan panjang > 1. Fungsi ini mengembalikan string dengan karakter apa pun dari *`remove_chars`* yang ditemukan di awal atau akhir string yang dihapus.

Contoh

```
TRIM('       foobar         ')              -- 'foobar'
TRIM('      \tfoobar\t         ')           -- '\tfoobar\t'
TRIM(LEADING FROM '       foobar         ')  -- 'foobar         '
TRIM(TRAILING FROM '       foobar         ') -- '       foobar'
TRIM(BOTH FROM '       foobar         ')     -- 'foobar'
TRIM(BOTH '12' FROM '1112211foobar22211122') -- 'foobar'
```

UPPER

Dengan string, UPPER mengubah semua karakter huruf kecil menjadi karakter huruf besar. Setiap karakter yang tidak dalam huruf kecil tidak berubah.

Sintaks

```
UPPER ( string )
```

Parameter

`string`

String target tempat fungsi beroperasi.

Contoh

```
UPPER('AbCdEfG!@#$') -- 'ABCDEFG!@#$'
```

Bekerja dengan bucket direktori

Bucket direktori mengatur data secara hierarkis ke dalam direktori, alih-alih struktur penyimpanan datar dari bucket tujuan umum. Tidak ada batas prefiks untuk bucket direktori dan direktori individu dapat menskalakan secara horizontal.

Anda dapat membuat hingga 100 bucket direktori di masing-masing Akun AWS, tanpa batasan jumlah objek yang dapat Anda simpan dalam ember. Kuota bucket Anda diterapkan ke setiap Wilayah di wilayah Anda Akun AWS. Jika aplikasi Anda memerlukan peningkatan batas ini, hubungi Dukungan.

> **Important**
>
> Bucket direktori di Availability Zones yang tidak memiliki aktivitas permintaan untuk periode minimal 90 hari transisi ke status tidak aktif. Saat dalam keadaan tidak aktif, bucket direktori sementara tidak dapat diakses untuk dibaca dan ditulis. Bucket yang tidak aktif menyimpan semua penyimpanan, metadata objek, dan metadata bucket. Biaya penyimpanan yang ada berlaku untuk ember yang tidak aktif. Jika Anda membuat permintaan akses ke bucket yang tidak aktif, bucket akan beralih ke status aktif, biasanya dalam beberapa menit. Selama periode transisi ini, membaca dan menulis mengembalikan kode `503 (Service Unavailable)` kesalahan HTTP. Ini tidak berlaku untuk bucket di Local Zones.

Ada beberapa jenis bucket Amazon S3. Sebelum membuat bucket, pastikan Anda memilih jenis bucket yang paling sesuai dengan aplikasi dan persyaratan kinerja Anda. Untuk informasi lebih lanjut tentang berbagai jenis bucket dan kasus penggunaan yang sesuai untuk masing-masing, lihatBucket.

Topik berikut menyediakan informasi tentang bucket direktori. Untuk informasi selengkapnya tentang bucket tujuan umum, lihat Ikhtisar ember tujuan umum.

Untuk informasi selengkapnya tentang bucket direktori, lihat topik berikut.

- Nama bucket direktori
- Direktori
- Nama kunci
- Manajemen akses

Nama bucket direktori

Nama bucket direktori terdiri dari nama dasar yang Anda berikan dan akhiran yang berisi ID Zona (Availability Zone atau Local Zone) tempat bucket Anda berada. Nama bucket direktori harus mengikuti format ini dan mematuhi aturan penamaan bucket direktori:

```
bucket-base-name--zone-id--x-s3
```

Misalnya, nama bucket direktori berikut berisi ID Zona Ketersediaan `usw2-az1`:

```
bucket-base-name--usw2-az1--x-s3
```

Untuk informasi selengkapnya, lihat Aturan penamaan bucket direktori.

Direktori

Bucket direktori mengatur data secara hierarkis ke dalam direktori, alih-alih struktur penyortiran datar dari bucket tujuan umum.

Dengan namespace hierarkis, pembatas dalam kunci objek menjadi penting. Satu-satunya pembatas yang didukung adalah garis miring (`/`). Direktori ditentukan oleh batas pembatas. Misalnya, kunci objek `dir1/dir2/file1.txt` menghasilkan direktori `dir1/` dan `dir2/` secara otomatis dibuat, dan objek `file1.txt` ditambahkan ke direktori `/dir2` di jalur `dir1/dir2/file1.txt`.

Model pengindeksan bucket direktori mengembalikan hasil yang tidak disortir untuk operasi API `ListObjectsV2`. Jika Anda perlu membatasi hasil Anda ke subbagian bucket Anda, Anda dapat menentukan jalur subdirektori dalam parameter `prefix`, misalnya, `prefix=dir1/`.

Nama kunci

Untuk bucket direktori, subdirektori yang umum untuk beberapa kunci objek dibuat dengan kunci objek pertama. Kunci objek tambahan untuk subdirektori yang sama menggunakan subdirektori yang dibuat sebelumnya. Model ini memberi Anda fleksibilitas dalam memilih kunci objek yang paling cocok untuk aplikasi, dengan dukungan yang sama untuk direktori yang jarang dan padat.

Manajemen akses

Bucket direktori memiliki semua pengaturan Blokir Akses Publik S3 yang diaktifkan secara default di tingkat bucket. Kepemilikan Objek S3 disetel ke pemilik bucket yang diberlakukan dan daftar kontrol akses (ACLs) dinonaktifkan. Pengaturan ini tidak dapat dimodifikasi.

Secara default, pengguna tidak memiliki izin untuk bucket direktori. Untuk memberikan izin akses bagi bucket direktori, Anda dapat menggunakan IAM untuk membuat pengguna, grup, atau peran dan melampirkan izin ke identitas tersebut. Untuk informasi selengkapnya, lihat Mengotorisasi operasi API endpoint Regional dengan IAM.

Anda juga dapat mengontrol akses ke bucket direktori melalui titik akses. Titik akses menyederhanakan pengelolaan akses data dalam skala besar untuk set data bersama di Amazon S3. Titik akses adalah nama host unik yang Anda buat untuk menegakkan izin dan kontrol jaringan yang berbeda untuk semua permintaan yang dibuat melalui titik akses. Untuk informasi selengkapnya, lihat Mengelola akses ke kumpulan data bersama dalam bucket direktori dengan titik akses.

Kuota ember direktori

Kuota, juga disebut sebagai batas, adalah jumlah maksimum sumber daya layanan atau operasi untuk Anda Akun AWS. Berikut ini adalah kuota untuk bucket direktori. Untuk informasi selengkapnya tentang kuota di Amazon S3, lihat kuota Amazon S3.

Nama	Default	Dapat disesuaikan	Deskripsi
Bucket direktori	Setiap Akun: 100	Ya	Jumlah bucket direktori Amazon S3 yang dapat Anda buat di akun.
Baca TPS per ember direktori	Setiap bucket direktori : hingga 200.000 TPS baca	Untuk meminta peningkatan kuota, hubungi Support.	Jumlah GET/HEAD permintaan per detik per bucket direktori.
Tulis TPS per ember direktori	Setiap bucket direktori : hingga 100.000 menulis TPS	Untuk meminta peningkatan kuota, hubungi Support.	Jumlah PUT/DELETE permintaan per detik per bucket direktori.

Membuat dan menggunakan bucket direktori

Untuk informasi selengkapnya tentang bekerja dengan bucket direktori, lihat topik berikut.

- Kasus penggunaan untuk ember direktori
- Perbedaan untuk ember direktori
- Jaringan untuk ember direktori
- Aturan penamaan bucket direktori
- Melihat properti bucket direktori
- Mengelola kebijakan bucket direktori
- Mengosongkan bucket direktori
- Menghapus bucket direktori
- Mendaftar bucket direktori
- Menentukan apakah Anda dapat mengakses bucket direktori
- Bekerja dengan objek di dalam bucket direktori
- Keamanan untuk ember direktori
- Mengelola akses ke kumpulan data bersama dalam bucket direktori dengan titik akses
- Mengoptimalkan kinerja bucket direktori
- Berkembang dengan ember direktori
- Menggunakan tag dengan ember direktori S3

Kasus penggunaan untuk ember direktori

Bucket direktori mendukung pembuatan bucket di jenis lokasi bucket berikut: Availability Zone atau Local Zone.

Untuk kasus penggunaan latensi rendah, Anda dapat membuat bucket direktori dalam satu Availability Zone untuk menyimpan data. Bucket direktori di Availability Zones mendukung kelas penyimpanan S3 Express One Zone. Kelas penyimpanan S3 Express One Zone direkomendasikan jika aplikasi Anda sensitif terhadap kinerja dan mendapat manfaat dari milidetik PUT dan latensi satu digit. GET Untuk mempelajari selengkapnya tentang membuat bucket direktori di Availability Zones, lihatBeban kerja kinerja tinggi.

Untuk kasus penggunaan residensi data, Anda dapat membuat bucket direktori dalam satu Zona Lokal AWS Khusus (DLZ) untuk menyimpan data. Bucket direktori di Local Zones mendukung kelas penyimpanan S3 One Zone-Infrequent Access (S3 One Zone-IA; Z-IA). Untuk mempelajari selengkapnya tentang membuat bucket direktori di Local Zones, lihatBeban kerja residensi data.

Topik

- Beban kerja kinerja tinggi
- Beban kerja residensi data

Beban kerja kinerja tinggi

S3 Express One Zone

Anda dapat menggunakan Amazon S3 Express One Zone untuk beban kerja berkinerja tinggi. S3 Express One Zone adalah kelas penyimpanan S3 pertama di mana Anda dapat memilih satu Availability Zone dengan opsi untuk menempatkan penyimpanan objek Anda bersama dengan sumber daya komputasi Anda yang memberikan kecepatan akses setinggi mungkin. Objek di S3 Express One Zone disimpan dalam bucket direktori yang terletak di Availability Zones. Untuk informasi selengkapnya tentang bucket direktori, lihat Bucket direktori.

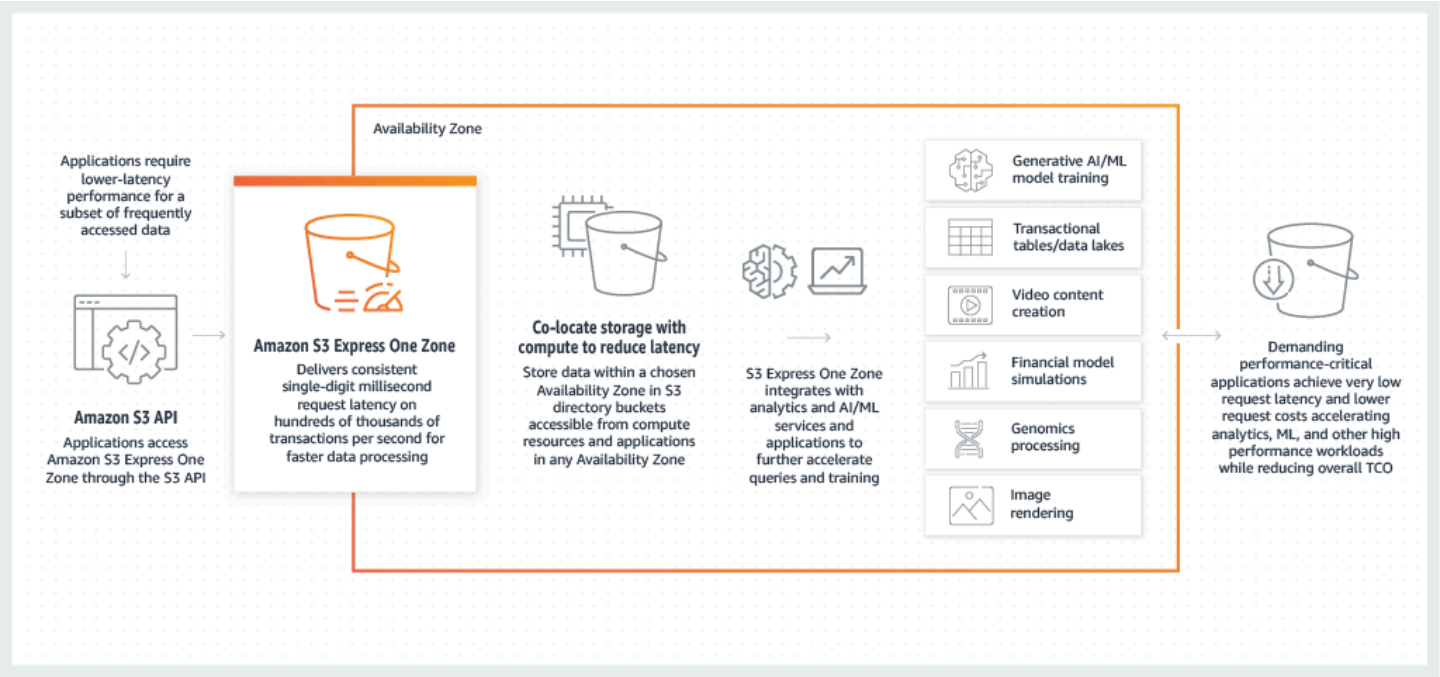
Amazon S3 Express One Zone adalah kelas penyimpanan Amazon S3 tunggal berkinerja tinggi yang dibuat khusus untuk menghadirkan akses data milidetik satu digit yang konsisten untuk aplikasi Anda yang paling sensitif terhadap latensi. S3 Express One Zone adalah kelas penyimpanan cloud-object latensi terendah yang tersedia saat ini, dengan kecepatan akses data hingga 10x lebih cepat dan dengan biaya permintaan 50 persen lebih rendah dari Standar S3. Aplikasi dapat memperoleh manfaat segera dari permintaan yang diselesaikan hingga urutan besarnya lebih cepat. S3 Express One Zone memberikan elastisitas kinerja yang serupa dengan kelas penyimpanan S3 lainnya. S3 Express One Zone digunakan untuk beban kerja atau aplikasi kritis kinerja yang membutuhkan latensi milidetik satu digit yang konsisten.

Seperti kelas penyimpanan Amazon S3 lainnya, Anda tidak perlu merencanakan atau menyediakan persyaratan kapasitas atau throughput terlebih dahulu. Anda dapat menskalakan penyimpanan Anda ke atas atau ke bawah, berdasarkan kebutuhan, dan mengakses data Anda melalui Amazon S3 API.

Kelas penyimpanan Amazon S3 Express One Zone dirancang untuk ketersediaan 99,95 persen dalam satu Availability Zone dan didukung oleh Perjanjian Tingkat Layanan Amazon S3. Dengan S3 Express One Zone, data Anda disimpan secara berlebihan di beberapa perangkat dalam satu Zona

Ketersediaan. S3 Express One Zone dirancang untuk menangani kegagalan perangkat bersamaan dengan cepat mendeteksi dan memperbaiki setiap redundansi yang hilang. Apabila perangkat yang ada mengalami kegagalan, S3 Express One Zone secara otomatis mengalihkan permintaan ke perangkat baru dalam Zona Ketersediaan. Redundansi ini membantu memastikan akses tanpa gangguan ke data Anda dalam Zona Ketersediaan.

S3 Express One Zone sangat ideal untuk aplikasi apa pun yang penting untuk meminimalkan latensi yang diperlukan untuk mengakses objek. Aplikasi semacam itu dapat berupa alur kerja interaktif manusia, seperti pengeditan video, di mana para profesional kreatif membutuhkan akses responsif ke konten dari antarmuka pengguna mereka. S3 Express One Zone juga menguntungkan beban kerja analitik dan machine learning yang memiliki persyaratan responsif serupa dari data mereka, terutama beban kerja dengan banyak akses yang lebih kecil atau sejumlah besar akses acak. S3 Express One Zone dapat digunakan bersama yang lain Layanan AWS untuk mendukung beban kerja analitik dan kecerdasan buatan dan pembelajaran mesin (AI/ML), seperti Amazon EMR, Amazon AI, dan Amazon Athena SageMaker .



Untuk bucket direktori yang menggunakan kelas penyimpanan S3 Express One Zone, data disimpan di beberapa perangkat dalam satu Availability Zone tetapi tidak menyimpan data secara berlebihan di seluruh Availability Zone. Saat membuat bucket direktori untuk menggunakan kelas penyimpanan S3 Express One Zone, sebaiknya tentukan Wilayah AWS dan Availability Zone yang lokal untuk Amazon, Amazon Elastic Kubernetes Service EC2, atau Amazon Elastic Container Service (Amazon ECS) menghitung instans untuk mengoptimalkan kinerja.

Saat menggunakan S3 Express One Zone, Anda dapat berinteraksi dengan bucket direktori Anda di cloud pribadi virtual (VPC) dengan menggunakan titik akhir VPC gateway. Dengan titik akhir gateway, Anda dapat mengakses bucket direktori S3 Express One Zone dari VPC Anda tanpa gateway internet atau perangkat NAT untuk VPC Anda, dan tanpa biaya tambahan.

Anda dapat menggunakan banyak operasi dan fitur API Amazon S3 yang sama dengan bucket direktori yang Anda gunakan dengan bucket tujuan umum dan kelas penyimpanan lainnya. Ini termasuk Mountpoint untuk Amazon S3, enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3), enkripsi sisi server dengan kunci () (AWS Key Management Service SSE-KMS), Operasi Batch S3, dan AWS KMS Akses Publik Blok S3. Anda dapat mengakses S3 Express One Zone dengan menggunakan konsol Amazon S3 AWS Command Line Interface ,AWS CLI() AWS SDKs, dan Amazon S3 REST API.

Untuk informasi selengkapnya tentang S3 Express One Zone, lihat topik berikut.

- Gambaran Umum
- Fitur S3 Express One Zone
- Layanan terkait
- Langkah selanjutnya

Gambaran Umum

Untuk mengoptimalkan kinerja dan mengurangi latensi, S3 Express One Zone memperkenalkan konsep baru berikut.

Zona Ketersediaan

Kelas penyimpanan Amazon S3 Express One Zone dirancang untuk ketersediaan 99,95 persen dalam satu Availability Zone dan didukung oleh Perjanjian Tingkat Layanan Amazon S3. Dengan S3 Express One Zone, data Anda disimpan secara berlebihan di beberapa perangkat dalam satu Zona Ketersediaan. S3 Express One Zone dirancang untuk menangani kegagalan perangkat bersamaan dengan cepat mendeteksi dan memperbaiki setiap redundansi yang hilang. Apabila perangkat yang ada mengalami kegagalan, S3 Express One Zone secara otomatis mengalihkan permintaan ke perangkat baru dalam Zona Ketersediaan. Redundansi ini membantu memastikan akses tanpa gangguan ke data Anda dalam Zona Ketersediaan.

Zona Ketersediaan adalah satu atau lebih pusat data diskret dengan daya redundan, jaringan, dan konektivitas dalam Wilayah AWS. Saat membuat bucket direktori, Anda memilih Availability Zone dan Wilayah AWS lokasi bucket Anda.

Zona Ketersediaan Tunggal

Saat membuat bucket direktori, Anda memilih Zona Ketersediaan dan Wilayah AWS.

Bucket direktori menggunakan kelas penyimpanan S3 Express One Zone, yang dibuat untuk digunakan oleh aplikasi yang sensitif terhadap kinerja. S3 Express One Zone adalah kelas penyimpanan S3 pertama di mana Anda dapat memilih satu Zona Ketersediaan dengan opsi untuk menempatkan bersama penyimpanan objek dengan sumber daya komputasi Anda, yang memberikan kecepatan akses setinggi mungkin.

Dengan S3 Express One Zone, data Anda disimpan secara berlebihan di beberapa perangkat dalam satu Zona Ketersediaan. S3 Express One Zone dirancang untuk ketersediaan 99,95 persen dalam satu Availability Zone dan didukung oleh Perjanjian Tingkat Layanan Amazon S3. Untuk informasi selengkapnya, lihat Zona Ketersediaan

Titik akhir dan titik akhir VPC gateway

Operasi API manajemen ember untuk bucket direktori tersedia melalui titik akhir Regional dan disebut sebagai operasi API titik akhir Regional. Contoh operasi API titik akhir Regional adalah `CreateBucket` dan `DeleteBucket`. Setelah membuat bucket direktori, Anda dapat menggunakan operasi API titik akhir Zonal untuk mengunggah dan mengelola objek di bucket direktori. Operasi API titik akhir Zona tersedia melalui titik akhir Zona. Contoh operasi API titik akhir Zonal adalah `PutObject` dan `CopyObject`.

Anda dapat mengakses S3 Express One Zone dari VPC Anda dengan menggunakan titik akhir VPC gateway. Setelah Anda membuat titik akhir gateway, Anda dapat menambahkannya sebagai target di tabel rute Anda untuk lalu lintas yang ditujukan dari VPC Anda ke S3 Express One Zone. Seperti Amazon S3, Tidak dikenakan biaya tambahan untuk menggunakan titik akhir gateway. Untuk informasi selengkapnya tentang cara mengonfigurasi titik akhir VPC gateway, lihat Jaringan untuk ember direktori

Otorisasi berbasis sesi

Dengan S3 Express One Zone, Anda mengautentikasi dan mengotorisasi permintaan melalui mekanisme berbasis sesi baru yang dioptimalkan untuk memberikan latensi terendah. Anda dapat menggunakan `CreateSession` untuk meminta kredensial sementara yang menyediakan akses latensi rendah ke bucket Anda. Kredensial sementara ini dicakup ke bucket direktori S3 tertentu. Token sesi hanya digunakan dengan operasi Zonal (tingkat objek) (dengan pengecualian). CopyObject Untuk informasi selengkapnya, lihat Mengotorisasi operasi API titik akhir Zonal dengan `CreateSession`.

Yang didukung AWS SDKs untuk S3 Express One Zone menangani pembentukan sesi dan penyegaran atas nama Anda. Untuk melindungi sesi Anda, kredensial keamanan sementara akan kedaluwarsa setelah 5 menit. Setelah Anda mengunduh dan menginstal AWS SDKs dan mengkonfigurasi izin yang diperlukan AWS Identity and Access Management (IAM), Anda dapat segera mulai menggunakan operasi API.

Fitur S3 Express One Zone

Fitur S3 berikut tersedia untuk S3 Express One Zone. Untuk mengetahui daftar lengkap operasi API yang didukung dan fitur yang tidak didukung, lihat. Perbedaan untuk ember direktori

Manajemen akses dan keamanan

Anda dapat menggunakan fitur berikut untuk mengaudit dan mengelola akses. Secara default, bucket direktori bersifat pribadi dan hanya dapat diakses oleh pengguna yang secara eksplisit diberikan akses. Tidak seperti bucket tujuan umum, yang dapat mengatur batas kendali akses pada tingkat bucket, prefiks, atau tag objek, batas kendali akses untuk bucket direktori ditetapkan hanya pada tingkat bucket. Untuk informasi selengkapnya, lihat Mengotorisasi operasi API endpoint Regional dengan IAM.

- S3 Block Public Access — Semua pengaturan Akses Publik Blok S3 diaktifkan secara default di tingkat bucket. Pengaturan default ini tidak dapat diubah.
- Kepemilikan Objek S3 (pemilik bucket diberlakukan secara default) — Daftar kontrol akses (ACLs) tidak didukung untuk bucket direktori. Bucket direktori secara otomatis menggunakan pengaturan yang diberlakukan pemilik bucket untuk Kepemilikan Objek S3. Pemilik bucket diberlakukan berarti ACLs dinonaktifkan, dan pemilik bucket secara otomatis memiliki dan memiliki kendali penuh atas setiap objek di bucket. Pengaturan default ini tidak dapat diubah.
- AWS Identity and Access Management (IAM) — IAM membantu Anda mengontrol akses ke bucket direktori Anda dengan aman. Anda dapat menggunakan IAM untuk memberikan akses ke operasi API manajemen bucket (Regional) dan operasi API manajemen objek (Zonal) melalui tindakan. `s3express:CreateSession` Untuk informasi selengkapnya, lihat Mengotorisasi operasi API endpoint Regional dengan IAM. Tidak seperti tindakan manajemen objek, tindakan manajemen bucket tidak dapat dilakukan lintas akun. Hanya pemilik bucket yang dapat melakukan tindakan tersebut.
- Kebijakan bucket–Gunakan bahasa kebijakan berbasis IAM untuk mengonfigurasi izin berbasis sumber daya untuk bucket direktori Anda. Anda juga dapat menggunakan IAM untuk mengontrol akses ke operasi `CreateSession` API, yang memungkinkan Anda menggunakan operasi API Zonal, atau manajemen objek. Anda dapat memberikan akses akun yang sama atau lintas akun

ke operasi API Zonal. Untuk informasi selengkapnya tentang izin dan kebijakan S3 Express One Zone, lihat. Mengotorisasi operasi API endpoint Regional dengan IAM

- IAM Access Analyzer for S3 — Evaluasi dan pantau kebijakan akses Anda untuk memastikan bahwa kebijakan tersebut hanya menyediakan akses yang dimaksudkan ke sumber daya S3 Anda.

Pencatatan log dan pemantauan

S3 Express One Zone menggunakan alat pencatatan dan pemantauan S3 berikut yang dapat Anda gunakan untuk memantau dan mengontrol bagaimana sumber daya Anda digunakan:

- CloudWatch Metrik Amazon — Pantau AWS sumber daya dan aplikasi Anda dengan menggunakan CloudWatch untuk mengumpulkan dan melacak metrik. S3 Express One Zone menggunakan CloudWatch namespace yang sama dengan kelas penyimpanan Amazon S3 lainnya (`AWS/S3`) dan mendukung metrik penyimpanan harian untuk bucket direktori: dan. `BucketSizeBytes` `NumberOfObjects` Untuk informasi selengkapnya, lihat Memantau metrik dengan Amazon CloudWatch.
- AWS CloudTrail log — AWS CloudTrail adalah log Layanan AWS yang membantu Anda menerapkan audit operasional dan risiko, tata kelola, dan kepatuhan Anda Akun AWS dengan mencatat tindakan yang diambil oleh pengguna, peran, atau. Layanan AWS Untuk S3 Express One Zone, CloudTrail menangkap operasi API titik akhir Regional (misalnya, `CreateBucket` dan`PutBucketPolicy`) sebagai peristiwa manajemen dan operasi API Zonal (misalnya, `GetObject` dan`PutObject`) sebagai peristiwa data. Peristiwa ini mencakup tindakan yang diambil dalam operasi AWS Management Console, AWS Command Line Interface (AWS CLI) AWS SDKs, dan AWS API. Untuk informasi selengkapnya, lihat Logging with AWS CloudTrail for S3 Express One Zone.

> **Note**
>
> Log akses server Amazon S3 tidak didukung dengan S3 Express One Zone.

Manajemen objek

Anda dapat mengelola penyimpanan objek Anda dengan menggunakan konsol Amazon S3, AWS SDKs, dan. AWS CLI Fitur-fitur berikut tersedia untuk manajemen objek dengan S3 Express One Zone:

- Operasi Batch S3 — Gunakan Operasi Batch untuk melakukan operasi massal pada objek dalam bucket direktori, misalnya, fungsi Copy dan AWS Lambda Invoke. Misalnya, Anda dapat menggunakan Operasi Batch untuk menyalin objek antara bucket direktori dan bucket tujuan umum. Dengan Operasi Batch, Anda dapat mengelola miliaran objek dalam skala besar dengan satu permintaan S3 dengan menggunakan AWS SDKs atau AWS CLI atau beberapa klik di konsol Amazon S3.
- Impor–Setelah membuat bucket direktori, Anda dapat mengisi bucket dengan objek menggunakan fitur impor di konsol Amazon S3. Impor adalah metode yang disederhanakan untuk membuat pekerjaan Operasi Batch guna menyalin objek dari bucket tujuan umum ke bucket direktori.

AWS SDKs dan pustaka klien

Anda dapat mengelola penyimpanan objek Anda dengan menggunakan pustaka AWS SDKs dan klien.

- Mountpoint untuk Amazon S3 - Mountpoint untuk Amazon S3 adalah klien file open source yang memberikan akses throughput tinggi, menurunkan biaya komputasi untuk data lake di Amazon S3. Mountpoint untuk Amazon S3 menerjemahkan panggilan API sistem file lokal ke panggilan API objek S3 seperti dan. `GET` `LIST` Ini sangat ideal untuk beban kerja data lake read-heavy yang memproses petabyte data dan membutuhkan throughput elastis tinggi yang disediakan oleh Amazon S3 untuk meningkatkan dan menurunkan di ribuan instance.
- S3A- S3A adalah antarmuka yang Hadoop kompatibel yang direkomendasikan untuk mengakses penyimpanan data di Amazon S3. S3Amenggantikan klien sistem S3N Hadoop file.
- PyTorchon AWS — PyTorch on AWS adalah kerangka pembelajaran mendalam open source yang membuatnya lebih mudah untuk mengembangkan model pembelajaran mesin dan menerapkannya ke produksi.
- AWS SDKs— Anda dapat menggunakan AWS SDKs saat mengembangkan aplikasi dengan Amazon S3. AWS SDKsSederhanakan tugas pemrograman Anda dengan membungkus API Amazon S3 REST yang mendasarinya. Untuk informasi selengkapnya tentang menggunakan AWS SDKs With S3 Express One Zone, lihatthe section called “AWS SDKs”.

Enkripsi dan perlindungan data

Objek di S3 Express One Zone secara otomatis dienkripsi oleh enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3). S3 Express One Zone juga mendukung enkripsi sisi server dengan kunci AWS Key Management Service (AWS KMS) (SSE-KMS). S3 Express One Zone tidak

mendukung enkripsi sisi server dengan kunci enkripsi yang disediakan pelanggan (SSE-C), atau enkripsi sisi server dua lapis dengan (DSSE-KMS). AWS KMS keys Untuk informasi selengkapnya, lihat Perlindungan data dan enkripsi.

S3 Express One Zone menawarkan kepada Anda opsi untuk memilih algoritma checksum yang digunakan untuk memvalidasi data Anda selama mengunggah atau mengunduh. Anda dapat memilih salah satu algoritma pemeriksaan integritas data Secure Hash Algorithms (SHA) atau Cyclic Redundancy Check (CRC) berikut:, C, SHA-1, dan SHA-256. CRC32 CRC32 MD5checksum berbasis tidak didukung dengan kelas penyimpanan S3 Express One Zone.

Untuk informasi selengkapnya, lihat Praktik terbaik checksum tambahan S3.

AWS Versi Tanda Tangan 4 (SigV4)

S3 Express One Zone menggunakan AWS Signature Versi 4 (SigV4). SigV4adalah protokol penandatanganan yang digunakan untuk mengautentikasi permintaan ke Amazon S3 melalui HTTPS. S3 Express One Zone menandatangani permintaan dengan menggunakan AWS Sigv4. Untuk informasi selengkapnya, lihat Mengautentikasi Permintaan (Versi AWS Tanda Tangan 4) di Referensi API Amazon Simple Storage Service.

Konsistensi kuat

S3 Express One Zone memberikan read-after-write konsistensi yang kuat untuk `PUT` dan `DELETE` permintaan objek di bucket direktori Anda secara keseluruhan. Wilayah AWS Untuk informasi selengkapnya, lihat Model konsistensi data Amazon S3.

Layanan terkait

Anda dapat menggunakan yang berikut ini Layanan AWS dengan kelas penyimpanan S3 Express One Zone untuk mendukung kasus penggunaan latensi rendah spesifik Anda.

- Amazon Elastic Compute Cloud (Amazon EC2) — Amazon EC2 menyediakan kapasitas komputasi yang aman dan terukur di. AWS Cloud Menggunakan Amazon EC2 mengurangi kebutuhan Anda untuk berinvestasi dalam perangkat keras di muka, sehingga Anda dapat mengembangkan dan menerapkan aplikasi lebih cepat. Anda dapat menggunakan Amazon EC2 untuk meluncurkan server virtual sebanyak atau sesedikit yang Anda butuhkan, mengonfigurasi keamanan dan jaringan, dan mengelola penyimpanan.
- AWS Lambda–Lambda adalah layanan komputasi yang memungkinkan Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Anda mengonfigurasi pengaturan pemberitahuan

di bucket, dan memberikan izin kepada Amazon S3 untuk memanggil fungsi di kebijakan izin berbasis sumber daya milik fungsi.

- Amazon Elastic Kubernetes Service (Amazon EKS) — Amazon EKS adalah layanan terkelola yang menghilangkan kebutuhan untuk menginstal, mengoperasikan, dan memelihara Kubernetes pesawat kontrol Anda sendiri. AWSKubernetesadalah sistem open source yang mengotomatiskan manajemen, penskalaan, dan penyebaran aplikasi kontainer.
- Amazon Elastic Container Service (Amazon ECS)—Amazon ECS adalah layanan orkestrasi kontainer yang dikelola sepenuhnya yang membantu Anda mendeploy, mengelola, dan menskalakan aplikasi kontainer dengan mudah.
- AWS Key Management Service (AWS KMS) - AWS Key Management Service (AWS KMS) adalah layanan AWS terkelola yang memudahkan Anda untuk membuat dan mengontrol kunci enkripsi yang digunakan untuk mengenkripsi data Anda. Kunci AWS KMS yang Anda buat di AWS KMS dilindungi oleh FIPS 140-2 modul keamanan perangkat keras tervalidasi (HSM). Untuk menggunakan atau mengelola kunci KMS Anda, Anda berinteraksi dengan AWS KMS.
- Amazon Athena–Athena adalah layanan kueri interaktif yang memudahkan analisis data di Amazon S3 dengan menggunakan SQL standar. Anda juga dapat menggunakan Athena untuk menjalankan analisis data secara interaktif dengan menggunakan Apache Spark tanpa harus merencanakan, mengonfigurasi, atau mengelola sumber daya. Ketika Anda menjalankan Apache Spark aplikasi di Athena, Anda mengirimkan Spark kode untuk diproses dan menerima hasilnya secara langsung.
- SageMakerPelatihan Amazon — Tinjau opsi untuk model pelatihan dengan AmazonSageMaker, termasuk algoritme bawaan, algoritme khusus, pustaka, dan model dari Marketplace. AWS
- AWS Glue— AWS Glue adalah layanan integrasi data tanpa server yang memudahkan pengguna analitik untuk menemukan, menyiapkan, memindahkan, dan mengintegrasikan data dari berbagai sumber. Anda dapat menggunakannya AWS Glue untuk analitik, pembelajaran mesin, dan pengembangan aplikasi. AWS Glue juga mencakup produktivitas tambahan dan perkakas operasi data untuk menulis, menjalankan pekerjaan, dan mengimplementasikan alur kerja bisnis.
- Amazon EMR - Amazon EMR adalah platform cluster terkelola yang menyederhanakan menjalankan kerangka kerja data besar, seperti Apache Hadoop danApache Spark, AWS untuk memproses dan menganalisis sejumlah besar data.
- AWS CloudTrail— AWS CloudTrail adalah AWS layanan yang membantu Anda mengaktifkan audit operasional dan risiko, tata kelola, dan kepatuhan akun Anda AWS . Tindakan yang diambil oleh pengguna, peran, atau AWS layanan dicatat sebagai peristiwa di CloudTrail. Peristiwa mencakup tindakan yang diambil di Konsol AWS Manajemen, Antarmuka Baris AWS Perintah, dan AWS SDKs dan APIs.

- AWS CloudFormation — adalah layanan yang membantu Anda memodelkan dan mengatur AWS sumber daya Anda sehingga Anda dapat menghabiskan lebih sedikit waktu mengelola sumber daya tersebut dan lebih banyak waktu untuk berfokus pada aplikasi Anda yang berjalan AWS. Anda membuat template yang menjelaskan semua AWS sumber daya yang Anda inginkan (seperti instans Amazon atau EC2 instans Amazon RDS DB), dan CloudFormation menangani penyediaan dan konfigurasi sumber daya tersebut untuk Anda. Anda tidak perlu membuat dan mengonfigurasi AWS sumber daya secara individual dan mencari tahu apa yang bergantung pada apa; CloudFormation menangani itu.

Langkah selanjutnya

Untuk informasi selengkapnya tentang bekerja dengan kelas penyimpanan dan bucket direktori S3 Express One Zone, lihat topik berikut:

- Tutorial: Memulai dengan S3 Express One Zone
- Zona Ketersediaan dan Wilayah S3 Express One Zone
- Jaringan untuk bucket direktori di Availability Zone
- Membuat bucket direktori di Availability Zone
- Titik akhir regional dan Zonal untuk bucket direktori di Availability Zone
- Mengoptimalkan kinerja S3 Express One Zone

Tutorial: Memulai dengan S3 Express One Zone

Amazon S3 Express One Zone adalah kelas penyimpanan S3 pertama tempat Anda dapat memilih satu Availability Zone dengan opsi untuk menempatkan penyimpanan objek bersama dengan sumber daya komputasi yang memberikan kecepatan akses setinggi mungkin. Data di S3 Express One Zone disimpan dalam bucket direktori yang terletak di Availability Zones. Untuk informasi selengkapnya tentang bucket direktori, lihat Bucket direktori.

S3 Express One Zone sangat ideal untuk aplikasi apa pun yang sangat penting untuk meminimalkan latensi permintaan. Aplikasi semacam itu dapat berupa alur kerja interaktif manusia, seperti pengeditan video, di mana para profesional kreatif membutuhkan akses responsif ke konten dari antarmuka pengguna mereka. S3 Express One Zone juga menguntungkan beban kerja analitik dan pembelajaran mesin yang memiliki persyaratan responsif serupa dari data mereka, terutama beban kerja dengan banyak akses yang lebih kecil atau sejumlah besar akses acak. S3 Express One Zone dapat digunakan dengan AWS layanan lain seperti Amazon EMR, Amazon Athena, AWS Glue Data

Catalog dan SageMaker Amazon Model Training untuk mendukung beban kerja analitik, kecerdasan buatan, dan pembelajaran mesin (AI/ML),. Anda dapat bekerja dengan kelas penyimpanan S3 Express One Zone dan bucket direktori dengan menggunakan konsol Amazon S3, Command Line Interface (AWS CLI) AWS SDKs AWS , dan Amazon S3 REST API. Untuk informasi selengkapnya, lihat Apa itu S3 Express One Zone? dan Bagaimana S3 Express One Zone berbeda? .
Ini adalah diagram alur kerja S3 Express One Zone.

Tujuan

Dalam tutorial ini, Anda akan belajar cara membuat titik akhir gateway, membuat dan melampirkan kebijakan IAM, membuat bucket direktori dan kemudian menggunakan tindakan Impor untuk mengisi bucket direktori Anda dengan objek yang saat ini disimpan di bucket tujuan umum Anda. Atau, Anda dapat mengunggah objek secara manual ke bucket direktori Anda.

Topik

- Prasyarat
- Langkah 1: Konfigurasikan titik akhir VPC gateway untuk mencapai bucket direktori S3 Express One Zone
- Langkah 2: Buat ember direktori S3 Express One Zone
- Langkah 3: Mengimpor data ke dalam bucket direktori S3 Express One Zone
- Langkah 4: Unggah objek secara manual ke bucket direktori S3 Express One Zone Anda
- Langkah 5: Kosongkan ember direktori S3 Express One Zone Anda
- Langkah 6: Hapus bucket direktori S3 Express One Zone Anda
- Langkah selanjutnya

Prasyarat

Sebelum Anda memulai tutorial ini, Anda harus memiliki Akun AWS yang dapat Anda masuki sebagai pengguna AWS Identity and Access Management (IAM) dengan izin yang benar.

Sublangkah

- Buat sebuah Akun AWS
- Buat pengguna IAM di Akun AWS (konsol) Anda
- Buat kebijakan IAM dan lampirkan ke pengguna atau peran IAM (konsol)

Buat sebuah Akun AWS

Untuk menyelesaikan tutorial ini, Anda memerlukan file Akun AWS. Ketika Anda mendaftar AWS, Anda Akun AWS secara otomatis mendaftar untuk semua layanan di AWS, termasuk Amazon S3. Anda hanya membayar biaya layanan yang Anda gunakan. Untuk informasi selengkapnya tentang harga, lihat harga S3.

Buat pengguna IAM di Akun AWS (konsol) Anda

AWS Identity and Access Management (IAM) adalah sebuah Layanan AWS yang membantu administrator mengontrol akses ke sumber daya dengan aman. AWS Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diotorisasi (memiliki izin) untuk mengakses objek dan menggunakan bucket direktori di S3 Express One Zone. Anda dapat menggunakan IAM tanpa biaya tambahan.

Secara default, pengguna tidak memiliki izin untuk mengakses bucket direktori dan melakukan operasi S3 Express One Zone. Untuk memberikan izin akses ke direktori bucket dan operasi S3 Express One Zone, Anda bisa menggunakan IAM untuk membuat pengguna atau peran dan melampirkan izin ke identitas tersebut. Untuk informasi selengkapnya tentang cara membuat pengguna IAM, lihat Membuat pengguna IAM (konsol) di Panduan Pengguna IAM. Untuk informasi selengkapnya tentang cara membuat peran IAM, lihat Membuat peran untuk mendelegasikan izin ke pengguna IAM di Panduan Pengguna IAM.

Untuk kesederhanaan, tutorial ini membuat dan menggunakan pengguna IAM. Setelah menyelesaikan tutorial ini, ingatlah untukHapus pengguna IAM. Untuk penggunaan produksi, kami menyarankan Anda mengikuti praktik terbaik Keamanan di IAM dalam Panduan Pengguna IAM. Praktik terbaik mengharuskan pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses AWS dengan kredensi sementara. Praktik terbaik lainnya adalah meminta beban kerja untuk menggunakan kredensil sementara dengan peran IAM untuk mengakses. AWS Untuk mempelajari selengkapnya tentang AWS IAM Identity Center cara membuat pengguna dengan kredensial sementara, lihat Memulai di AWS IAM Identity Center Panduan Pengguna.

> **Warning**
>
> Pengguna IAM memiliki kredensi jangka panjang, yang menghadirkan risiko keamanan. Untuk membantu mengurangi risiko ini, kami menyarankan agar Anda memberikan pengguna ini hanya izin yang mereka perlukan untuk melakukan tugas dan menghapus pengguna ini ketika mereka tidak lagi diperlukan.

Buat kebijakan IAM dan lampirkan ke pengguna atau peran IAM (konsol)

Secara default, pengguna tidak memiliki izin untuk bucket direktori dan operasi S3 Express One Zone. Untuk memberikan izin akses bagi bucket direktori, Anda dapat menggunakan IAM untuk membuat pengguna, grup, atau peran dan melampirkan izin ke identitas tersebut. Bucket direktori adalah satu-satunya sumber daya yang dapat Anda sertakan dalam kebijakan bucket atau kebijakan identitas IAM untuk akses S3 Express One Zone.

Untuk menggunakan operasi API titik akhir Regional (operasi tingkat ember atau bidang kontrol) dengan S3 Express One Zone, Anda menggunakan model otorisasi IAM, yang tidak melibatkan manajemen sesi. Izin diberikan untuk tindakan yang dilakukan secara individual. Untuk menggunakan operasi API titik akhir Zonal (operasi tingkat objek atau bidang data), Anda gunakan `CreateSession`untuk membuat dan mengelola sesi yang dioptimalkan untuk otorisasi permintaan data latensi rendah. Untuk mengambil dan menggunakan token sesi, Anda harus mengizinkan tindakan `s3express:CreateSession` untuk bucket direktori Anda dalam kebijakan berbasis identitas atau kebijakan bucket. Jika Anda mengakses S3 Express One Zone di konsol Amazon S3, melalui AWS Command Line Interface (AWS CLI), atau dengan menggunakan, S3 Express One Zone membuat sesi AWS SDKs atas nama Anda. Untuk informasi selengkapnya, lihat `CreateSession`otorisasi dan AWS Identity and Access Management (IAM) untuk S3 Express One Zone.

Untuk membuat kebijakan IAM dan melampirkan kebijakan ke pengguna (atau peran) IAM

1. Masuk ke Konsol AWS Manajemen dan buka Konsol Manajemen IAM.
2. Di panel navigasi, pilih Kebijakan.
3. Pilih Buat Kebijakan.
4. Pilih JSON.
5. Salin kebijakan di bawah ini ke jendela Editor kebijakan. Sebelum Anda dapat membuat bucket direktori atau menggunakan S3 Express One Zone, Anda harus memberikan izin yang diperlukan untuk peran AWS Identity and Access Management (IAM) atau pengguna Anda. Kebijakan contoh ini memungkinkan akses ke operasi `CreateSession` API (untuk digunakan dengan operasi API Zonal atau tingkat objek lainnya) dan semua operasi API titik akhir Regional (tingkat ember). Kebijakan ini memungkinkan operasi API `CreateSession` untuk digunakan dengan semua bucket direktori, tetapi operasi API titik akhir Regional hanya diizinkan untuk digunakan dengan bucket direktori yang ditentukan. Untuk menggunakan kebijakan contoh ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
{
     "Version": "2012-10-17",
     "Statement": [
         {
             "Sid": "AllowAccessRegionalEndpointAPIs",
             "Effect": "Allow",
             "Action": [
                 "s3express:DeleteBucket",
                 "s3express:DeleteBucketPolicy",
                 "s3express:CreateBucket",
                 "s3express:PutBucketPolicy",
                 "s3express:GetBucketPolicy",
                 "s3express:ListAllMyDirectoryBuckets"
             ],

             "Resource": "arn:aws:s3express:region:account_id:bucket/bucket-base-
name--zone-id--x-s3/*"
         },
         {
             "Sid": "AllowCreateSession",
             "Effect": "Allow",
             "Action": "s3express:CreateSession",
             "Resource": "*"
         }
     ]
 }
```

6. Pilih Berikutnya.
7. Sebutkan kebijakan tersebut.

> **Note**
>
> Tag bucket tidak didukung untuk S3 Express One Zone.

8. Pilih Buat kebijakan.
9. Sekarang setelah Anda membuat kebijakan IAM, Anda dapat melampirkannya ke pengguna IAM. Di panel navigasi, pilih Kebijakan.
10. Di bilah pencarian, masukkan nama kebijakan Anda.
11. Dari menu Tindakan, pilih Lampirkan.
12. Di bawah Filter menurut Jenis Entitas, pilih pengguna IAM atau Peran.

13. Di kolom pencarian, ketikkan nama pengguna atau peran yang ingin Anda gunakan.
14. Pilih Lampirkan Kebijakan.

Langkah 1: Konfigurasikan titik akhir VPC gateway untuk mencapai bucket direktori S3 Express One Zone

Anda dapat mengakses operasi API Zonal dan Regional melalui titik akhir cloud privat virtual (VPC) gateway. Titik akhir Gateway dapat memungkinkan lalu lintas mencapai S3 Express One Zone tanpa melintasi Gateway NAT. Kami sangat menyarankan untuk menggunakan titik akhir gateway karena mereka menyediakan jalur jaringan paling optimal saat bekerja dengan S3 Express One Zone. Anda dapat mengakses bucket direktori S3 Express One Zone dari VPC Anda tanpa gateway internet atau perangkat NAT untuk VPC Anda, dan tanpa biaya tambahan. Gunakan prosedur berikut untuk mengonfigurasi titik akhir gateway yang terhubung ke objek kelas penyimpanan S3 Express One Zone dan bucket direktori.

Untuk mengakses S3 Express One Zone, Anda menggunakan titik akhir Regional dan Zonal yang berbeda dari titik akhir Amazon S3 standar. Bergantung pada operasi Amazon S3 API yang Anda gunakan, titik akhir Zonal atau Regional diperlukan. Untuk daftar lengkap operasi API yang didukung menurut tipe titik akhir, lihat Operasi API yang didukung oleh S3 Express One Zone. Anda harus mengakses titik akhir Zonal dan Regional melalui titik akhir gateway cloud privat virtual (VPC).

Gunakan prosedur berikut untuk membuat titik akhir gateway yang terhubung ke objek kelas penyimpanan S3 Express One Zone dan bucket direktori.

Untuk mengonfigurasi titik akhir VPC gateway

1. Buka Konsol VPC Amazon di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi samping di bawah Virtual private cloud, pilih Endpoints.
3. Pilih Buat Titik Akhir.
4. Buat nama untuk titik akhir Anda.
5. Untuk Kategori layanan, pilih Layanan AWS.
6. Di bawah Layanan, cari menggunakan filter Type=Gateway dan kemudian pilih tombol opsi di sebelah com.amazonaws. ***region***.s3express.
7. Untuk VPC, pilih VPC yang akan digunakan untuk menciptakan titik akhir.
8. Untuk Tabel rute, pilih tabel rute yang akan digunakan oleh titik akhir. Amazon VPC secara otomatis menambahkan rute yang mengarahkan lalu lintas yang ditujukan untuk layanan ke antarmuka jaringan titik akhir.

9. Untuk Kebijakan, pilih Akses penuh untuk mengizinkan semua operasi oleh semua pengguna utama pada semua sumber daya melalui titik akhir VPC. Jika tidak, pilih Kustom untuk melampirkan kebijakan titik akhir VPC yang mengontrol izin yang dimiliki pengguna utama untuk melakukan tindakan pada sumber daya melalui titik akhir VPC.
10. Pilih Buat Titik Akhir.

Setelah membuat titik akhir gateway, Anda dapat menggunakan titik akhir API Regional dan titik akhir API Zonal untuk mengakses objek kelas penyimpanan Amazon S3 Express One Zone dan bucket direktori.

Langkah 2: Buat ember direktori S3 Express One Zone

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi di bagian atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah tempat Anda ingin membuat ember.

   > **Note**
   >
   > Untuk meminimalkan latensi dan biaya serta memenuhi persyaratan regulasi, pilih Wilayah yang dekat dengan Anda. Objek yang disimpan di Wilayah tidak pernah keluar dari Wilayah kecuali Anda secara tegas mentransfer atau mereplikasinya ke Wilayah lain. Untuk daftar Amazon S3 Wilayah AWS, lihat Layanan AWS titik akhir di. Referensi Umum Amazon Web Services

3. Di panel navigasi kiri, pilih Bucket direktori.
4. Pilih Buat bucket. Halaman Buat bucket terbuka.
5. Di bawah Konfigurasi umum, lihat Wilayah AWS tempat bucket Anda akan dibuat.

   Di bawah Jenis Bucket, pilih Direktori.

   > **Note**
   >
   > - Jika Anda memilih Region yang tidak mendukung bucket direktori, opsi jenis Bucket akan hilang, dan tipe bucket default ke bucket tujuan umum. Untuk membuat bucket direktori, Anda harus memilih Region yang didukung. Untuk daftar Wilayah yang

mendukung bucket direktori dan kelas penyimpanan Amazon S3 Express One Zone, lihat. the section called “Zona Ketersediaan dan Wilayah S3 Express One Zone”

- Setelah membuat bucket, Anda tidak dapat mengubah tipe bucket.

> **Note**
>
> Zona Ketersediaan tidak dapat diubah setelah bucket dibuat.

6. Untuk Zona Ketersediaan, pilih Zona Ketersediaan lokal untuk layanan komputasi Anda. Untuk daftar Availability Zones yang mendukung bucket direktori dan kelas penyimpanan S3 Express One Zone, lihat. the section called “Zona Ketersediaan dan Wilayah S3 Express One Zone”

   Di Availability Zone, pilih kotak centang untuk mengetahui bahwa jika terjadi pemadaman Availability Zone, data Anda mungkin tidak tersedia atau hilang.

   > **Important**
   >
   > Meskipun bucket direktori disimpan di beberapa perangkat dalam satu Availability Zone, bucket direktori tidak menyimpan data secara berlebihan di Availability Zone.

7. Untuk Nama bucket, masukkan nama untuk bucket direktori Anda.

   Aturan penamaan berikut berlaku untuk bucket direktori.

   - Jadilah unik dalam Zona yang dipilih (AWS Availability Zone atau AWS Local Zone).
   - Panjang nama harus antara 3 (menit) dan 63 (maks) karakter, termasuk akhiran.
   - Hanya terdiri dari huruf kecil, angka, tanda hubung (-).
   - Dimulai dan diakhiri dengan huruf atau angka.
   - Harus menyertakan akhiran berikut:`--`*`zone-id`*`--x-s3`.
   - Nama bucket tidak boleh dimulai dengan prefiks `xn--`.
   - Nama bucket tidak boleh dimulai dengan prefiks `sthree-`.
   - Nama bucket tidak boleh dimulai dengan prefiks `sthree-configurator`.
   - Nama bucket tidak boleh dimulai dengan prefiks `amzn-s3-demo-`.
   - Nama bucket tidak boleh diakhiri dengan sufiks `-s3alias`. Sufiks ini dicadangkan untuk nama alias titik akses. Untuk informasi selengkapnya, lihat Alias titik akses.

- Nama bucket tidak boleh diakhiri dengan sufiks `--ol-s3`. Sufiks ini dicadangkan untuk nama alias Titik Akses Lambda Objek. Untuk informasi selengkapnya, lihat Cara menggunakan alias gaya bucket untuk bucket S3 Anda Titik Akses Lambda Objek.
- Nama bucket tidak boleh diakhiri dengan sufiks `.mrap`. Akhiran ini dicadangkan untuk nama Titik Akses Multi-Wilayah. Untuk informasi selengkapnya, lihat Aturan penamaan Titik Akses Multi-Wilayah Amazon S3.

Sufiks secara otomatis ditambahkan ke nama dasar yang Anda berikan saat membuat bucket direktori menggunakan konsol. Sufiks ini mencakup ID Zona Ketersediaan dari Zona Ketersediaan yang Anda pilih.

Setelah membuat bucket, Anda tidak dapat mengubah namanya. Untuk informasi selengkapnya tentang penamaan bucket, lihat Aturan penamaan bucket tujuan umum.

> **Important**
>
> Jangan sertakan informasi sensitif, seperti nomor akun, dalam nama bucket. Nama ember terlihat di titik URLs itu ke objek di ember.

8. Di bawah Kepemilikan Objek, setelan yang diberlakukan pemilik Bucket diaktifkan secara otomatis, dan semua daftar kontrol akses (ACLs) dinonaktifkan. Untuk bucket direktori, tidak ACLs dapat diaktifkan.

   Pemilik bucket diberlakukan (default) — ACLs dinonaktifkan, dan pemilik bucket secara otomatis memiliki dan memiliki kontrol penuh atas setiap objek di bucket tujuan umum. ACLs tidak lagi memengaruhi izin akses ke data di bucket tujuan umum S3. Bucket menggunakan kebijakan secara eksklusif untuk menentukan kontrol akses.

9. Di bawah pengaturan Blokir Akses Publik untuk bucket ini, semua setelan Blokir Akses Publik untuk bucket direktori Anda diaktifkan secara otomatis. Pengaturan ini tidak dapat dimodifikasi untuk bucket direktori. Untuk informasi lebih lanjut tentang pemblokiran akses publik, lihat Melakukan blok akses publik ke penyimpanan Amazon S3 Anda.
10. Untuk mengonfigurasi enkripsi default, di bawah Jenis enkripsi, pilih salah satu dari berikut ini:

    - Enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3)
    - Enkripsi sisi server dengan AWS Key Management Service kunci (SSE-KMS)

Untuk informasi selengkapnya tentang penggunaan enkripsi di sisi server Amazon S3 guna mengenkripsi data Anda, lihat Perlindungan data dan enkripsi.

> **Important**
>
> Jika Anda menggunakan opsi SSE-KMS untuk konfigurasi enkripsi default, Anda tunduk pada kuota permintaan per detik (RPS) AWS KMS. Untuk informasi selengkapnya tentang kuota AWS KMS dan cara meminta kenaikan kuota, lihat Kuota di Panduan Pengembang AWS Key Management Service .
> Saat mengaktifkan enkripsi default, Anda mungkin perlu memperbarui kebijakan bucket. Untuk informasi selengkapnya, lihat Menggunakan enkripsi SSE-KMS untuk operasi lintas akun.

11. Jika Anda memilih enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3), di bawah Kunci Bucket, Diaktifkan akan muncul. Kunci Bucket S3 selalu diaktifkan saat Anda mengonfigurasi bucket direktori untuk menggunakan enkripsi default dengan SSE-S3. Kunci Bucket S3 selalu diaktifkan untuk `GET` dan `PUT` beroperasi di bucket direktori dan tidak dapat dinonaktifkan. Kunci Bucket S3 tidak didukung, saat Anda menyalin objek terenkripsi SSE-KMS dari bucket tujuan umum ke bucket direktori, dari bucket direktori ke bucket tujuan umum, atau di antara bucket direktori, melalui, operasi dalam Operasi Batch, atau pekerjaan CopyObject. UploadPartCopyCopyimport Dalam hal ini, Amazon S3 melakukan panggilan ke AWS KMS setiap kali permintaan salinan dibuat untuk objek yang dienkripsi KMS.

    S3 Bucket Keys menurunkan biaya enkripsi dengan mengurangi lalu lintas permintaan dari Amazon S3 ke. AWS KMS Untuk informasi selengkapnya, lihat Mengurangi biaya SSE-KMS dengan Kunci Bucket Amazon S3.

12. Jika Anda memilih enkripsi sisi server dengan AWS Key Management Service kunci (SSE-KMS), di bawah AWS KMS kunci, tentukan AWS Key Management Service kunci Anda dengan salah satu cara berikut atau buat kunci baru.

    - Untuk memilih dari daftar kunci KMS yang tersedia, pilih Pilih dari Anda AWS KMS keys, dan pilih kunci KMS Anda dari Tersedia. AWS KMS keys

      Hanya kunci terkelola pelanggan Anda yang muncul di daftar ini. The Kunci yang dikelola AWS (`aws/s3`) tidak didukung dalam bucket direktori. Untuk informasi selengkapnya tentang

CMK, lihat Kunci pelanggan dan AWS kunci di AWS Key Management Service Panduan Pengembang.

- Untuk memasukkan tombol KMS ARN atau alias, pilih Masukkan ARN, dan masukkan kunci KMS AWS KMS key ARN atau alias Anda di ARN.AWS KMS key
- Untuk membuat kunci terkelola pelanggan baru di AWS KMS konsol, pilih Buat kunci KMS.

  Untuk informasi selengkapnya tentang membuat AWS KMS key, lihat Membuat kunci di Panduan AWS Key Management Service Pengembang.

Important

- Konfigurasi SSE-KMS Anda hanya dapat mendukung 1 kunci terkelola pelanggan per bucket direktori selama masa pakai bucket. Kunci yang dikelola AWS(`aws/s3`) tidak didukung. Selain itu, setelah menentukan kunci terkelola pelanggan untuk SSE-KMS, Anda tidak dapat mengganti kunci terkelola pelanggan untuk konfigurasi SSE-KMS bucket.

  Anda dapat mengidentifikasi kunci terkelola pelanggan yang Anda tentukan untuk konfigurasi SSE-KMS bucket, dengan cara berikut:

  - Anda membuat permintaan operasi `HeadObject` API untuk menemukan nilai `x-amz-server-side-encryption-aws-kms-key-id` dalam respons Anda.

  Untuk menggunakan kunci terkelola pelanggan baru untuk data Anda, sebaiknya salin objek yang ada ke bucket direktori baru dengan kunci terkelola pelanggan baru.
- Anda hanya dapat menggunakan tombol KMS yang tersedia Wilayah AWS sama dengan bucket. Konsol Amazon S3 hanya mencantumkan kunci 100 KMS pertama di Wilayah yang sama dengan bucket. Untuk menggunakan kunci KMS yang tidak terdaftar, Anda harus memasukkan ARN kunci KMS Anda. Jika Anda ingin menggunakan kunci KMS yang dimiliki oleh akun lain, Anda harus terlebih dahulu memiliki izin untuk menggunakan kunci tersebut, dan kemudian Anda harus memasukkan ARN kunci KMS. Untuk informasi selengkapnya tentang izin lintas akun untuk kunci KMS, lihat Membuat kunci KMS yang dapat digunakan akun lain di Panduan Pengembang AWS Key Management Service . Untuk informasi selengkapnya tentang SSE-KMS, lihat Menentukan enkripsi sisi server dengan AWS KMS (SSE-KMS) untuk unggahan objek baru di bucket direktori.

- Bila Anda menggunakan enkripsi sisi server AWS KMS key untuk bucket direktori, Anda harus memilih kunci KMS enkripsi simetris. Amazon S3 hanya mendukung kunci KMS enkripsi simetris dan tidak mendukung kunci KMS asimetris. Untuk informasi selengkapnya, lihat Mengidentifikasi tombol KMS simetris dan asimetris dalam Panduan Pengembang AWS Key Management Service .

Untuk informasi selengkapnya tentang penggunaan AWS KMS dengan Amazon S3, lihat. Menggunakan enkripsi sisi server dengan AWS KMS kunci (SSE-KMS) di bucket direktori

13. Pilih Buat bucket. Setelah membuat bucket, Anda dapat menambahkan file dan folder ke bucket. Untuk informasi selengkapnya, lihat the section called “Bekerja dengan objek di dalam bucket direktori”.

Langkah 3: Mengimpor data ke dalam bucket direktori S3 Express One Zone

Untuk menyelesaikan langkah ini, Anda harus memiliki bucket tujuan umum yang berisi objek dan terletak Wilayah AWS sama dengan bucket direktori Anda.

Setelah membuat bucket direktori di Amazon S3, Anda dapat mengisi bucket baru dengan data menggunakan tindakan Impor di konsol Amazon S3. Impor menyederhanakan penyalinan data ke dalam bucket direktori dengan memungkinkan Anda memilih awalan atau bucket tujuan umum untuk Mengimpor data dari tanpa harus menentukan semua objek yang akan disalin satu per satu. Impor menggunakan Operasi Batch S3 yang menyalin objek dalam awalan yang dipilih atau bucket tujuan umum. Anda dapat memantau kemajuan pekerjaan Impor salinan melalui halaman detail pekerjaan Operasi Batch S3.

Untuk menggunakan tindakan Impor

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi di bagian atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah yang terkait dengan Availability Zone tempat bucket direktori Anda berada.
3. Di panel navigasi kiri, pilih Bucket direktori.
4. Pilih tombol opsi di sebelah nama bucket tempat Anda ingin mengimpor objek.
5. Pilih Impor.

6. Untuk Sumber, masukkan bucket bertujuan umum (atau jalur bucket termasuk awalan) yang berisi objek yang ingin Anda impor. Untuk memilih bucket bertujuan umum yang ada dari daftar, pilih Jelajahi S3.
7. Di bagian Izin, Anda dapat memilih untuk membuat peran IAM secara otomatis. Atau, Anda dapat memilih peran IAM dari daftar, atau langsung memasukkan peran IAM ARN.

   - Untuk mengizinkan Amazon S3 membuat peran IAM baru atas nama Anda, pilih Buat peran IAM baru.

     > **Note**
     >
     > Jika objek sumber Anda dienkripsi dengan enkripsi di sisi server dengan kunci AWS Key Management Service (AWS KMS) (SSE-KMS), jangan pilih opsi Buat peran IAM baru. Sebagai gantinya, tentukan peran IAM lama yang memiliki izin `kms:Decrypt`. Amazon S3 akan menggunakan izin ini untuk mendekripsi objek Anda. Selama proses impor, Amazon S3 kemudian akan mengenkripsi ulang objek tersebut dengan menggunakan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3).

   - Untuk memilih peran IAM yang ada dari daftar, pilih Pilih dari peran IAM yang ada.
   - Untuk menentukan peran IAM yang ada dengan memasukkan Amazon Resource Name (ARN), pilih Masukkan ARN peran IAM, lalu masukkan ARN di bidang yang sesuai.
8. Tinjau informasi yang ditampilkan di bagian Tujuan dan Pengaturan objek yang disalin. Jika informasi di bagian Tujuan sudah benar, pilih Impor untuk memulai tugas penyalinan.

   Konsol Amazon S3 menampilkan status tugas baru Anda di halaman Operasi Batch. Untuk informasi selengkapnya tentang tugas, pilih tombol opsi di sebelah nama tugas, lalu pada menu Tindakan, pilih Lihat detail. Untuk membuka bucket direktori tempat objek akan diimpor, pilih Lihat tujuan impor.

Langkah 4: Unggah objek secara manual ke bucket direktori S3 Express One Zone Anda

Anda juga dapat mengunggah objek secara manual ke bucket direktori Anda.

Untuk mengunggah objek secara manual

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

2. Di bilah navigasi di sudut kanan atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah yang terkait dengan Availability Zone tempat bucket direktori Anda berada.
3. Di panel navigasi kiri, pilih Bucket direktori.
4. Pilih nama bucket yang ingin Anda unggah folder atau file Anda.

> **Note**
>
> Jika Anda memilih bucket direktori yang sama dengan yang Anda gunakan pada langkah sebelumnya dari tutorial ini, bucket direktori Anda akan berisi objek yang diunggah dari alat Impor. Perhatikan bahwa objek-objek ini sekarang disimpan di kelas penyimpanan S3 Express One Zone.

5. Dalam daftar Objek, pilih Unggah.
6. Pada halaman Unggah, lakukan salah satu hal berikut:
   - Seret dan lepas file dan folder ke area unggahan bertitik.
   - Pilih Tambahkan file atau Tambah folder, pilih file atau folder yang akan diunggah, lalu pilih Buka atau Unggah.
7. Di bawah Checksum, pilih fungsi Checksum yang ingin Anda gunakan.

> **Note**
>
> Kami merekomendasikan penggunaan CRC32 dan CRC32 C untuk kinerja terbaik dengan kelas penyimpanan S3 Express One Zone. Untuk informasi selengkapnya, lihat praktik terbaik checksum tambahan S3.

   (Opsional) Jika Anda mengunggah satu objek yang berukuran kurang dari 16 MB, Anda juga dapat menentukan nilai checksum yang telah dihitung sebelumnya. Saat Anda memberikan nilai yang telah dihitung sebelumnya, Amazon S3 membandingkannya dengan nilai yang dihitung dengan menggunakan fungsi checksum yang dipilih. Jika nilai tidak cocok, unggahan tidak akan dimulai.
8. Opsi di bagian Izin dan Properti secara otomatis diatur ke pengaturan default dan tidak dapat dimodifikasi. Blokir Akses Publik diaktifkan secara otomatis, dan Versi S3 dan Kunci Objek S3 tidak dapat diaktifkan untuk bucket direktori.

(Opsional) Jika Anda ingin menambahkan metadata dalam pasangan nilai kunci ke objek Anda, perluas bagian Properti, lalu di bagian Metadata, pilih Tambahkan metadata.

9. Untuk mengunggah file dan folder yang terdaftar, pilih Unggah.

   Amazon S3 mengunggah objek dan folder Anda. Setelah unggahan selesai, Anda melihat pesan sukses di halaman Unggahan: status.

   Anda telah berhasil membuat bucket direktori dan mengunggah objek ke bucket Anda.

Langkah 5: Kosongkan ember direktori S3 Express One Zone Anda

Anda dapat mengosongkan bucket direktori Amazon S3 Anda dengan menggunakan konsol Amazon S3.

Untuk mengosongkan ember direktori

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi di sudut kanan atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah yang terkait dengan Availability Zone tempat bucket direktori Anda berada.
3. Di panel navigasi kiri, pilih Bucket direktori.
4. Pilih tombol opsi di sebelah nama ember yang ingin Anda kosongkan, lalu pilih Kosong.
5. Di halaman Bucket kosong, konfirmasikan bahwa Anda ingin mengosongkan bucket dengan mengetik **`permanently delete`** ke dalam bidang teks, lalu pilih Kosongkan.
6. Pantau perkembangan proses pengosongan bucket di halaman Empty bucket: status.

Langkah 6: Hapus bucket direktori S3 Express One Zone Anda

Setelah mengosongkan bucket direktori dan membatalkan semua unggahan multipart yang sedang berlangsung, Anda dapat menghapus bucket dengan menggunakan konsol Amazon S3.

Untuk menghapus bucket direktori

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

2. Di bilah navigasi di sudut kanan atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah yang terkait dengan Availability Zone tempat bucket direktori Anda berada.
3. Di panel navigasi kiri, pilih Bucket direktori.
4. Dalam daftar bucket Direktori, pilih tombol opsi di sebelah bucket yang ingin Anda hapus.
5. Pilih Hapus.
6. Pada halaman Delete bucket, masukkan nama bucket di kolom teks untuk mengonfirmasi penghapusan bucket Anda.

> **Important**
>
> Menghapus bucket direktori tidak dapat dibatalkan.

7. Untuk menghapus bucket direktori, pilih Hapus bucket.

Langkah selanjutnya

Dalam tutorial ini, Anda telah belajar cara membuat bucket direktori dan menggunakan kelas penyimpanan S3 Express One Zone. Setelah menyelesaikan tutorial ini, Anda dapat menjelajahi AWS layanan terkait untuk digunakan dengan kelas penyimpanan S3 Express One Zone.

Anda dapat menggunakan yang berikut ini Layanan AWS dengan kelas penyimpanan S3 Express One Zone untuk mendukung kasus penggunaan latensi rendah spesifik Anda.

- Amazon Elastic Compute Cloud (Amazon EC2) — Amazon EC2 menyediakan kapasitas komputasi yang aman dan terukur di. AWS Cloud Menggunakan Amazon EC2 mengurangi kebutuhan Anda untuk berinvestasi dalam perangkat keras di muka, sehingga Anda dapat mengembangkan dan menerapkan aplikasi lebih cepat. Anda dapat menggunakan Amazon EC2 untuk meluncurkan server virtual sebanyak atau sesedikit yang Anda butuhkan, mengonfigurasi keamanan dan jaringan, dan mengelola penyimpanan.
- AWS Lambda–Lambda adalah layanan komputasi yang memungkinkan Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Anda mengonfigurasi pengaturan pemberitahuan di bucket, dan memberikan izin kepada Amazon S3 untuk memanggil fungsi di kebijakan izin berbasis sumber daya milik fungsi.
- Amazon Elastic Kubernetes Service (Amazon EKS) — Amazon EKS adalah layanan terkelola yang menghilangkan kebutuhan untuk menginstal, mengoperasikan, dan memelihara

Kubernetes pesawat kontrol Anda sendiri. AWSKubernetesadalah sistem sumber terbuka yang mengotomatiskan manajemen, penskalaan, dan penyebaran aplikasi kontainer.

- Amazon Elastic Container Service (Amazon ECS)—Amazon ECS adalah layanan orkestrasi kontainer yang dikelola sepenuhnya yang membantu Anda mendeploy, mengelola, dan menskalakan aplikasi kontainer dengan mudah.
- Amazon EMR - Amazon EMR adalah platform cluster terkelola yang menyederhanakan menjalankan kerangka kerja data besar, seperti Apache Hadoop dan Apache Spark seterusnya AWS untuk memproses dan menganalisis sejumlah besar data.
- Amazon Athena–Athena adalah layanan kueri interaktif yang memudahkan analisis data di Amazon S3 dengan menggunakan SQL standar. Anda juga dapat menggunakan Athena untuk menjalankan analisis data secara interaktif dengan menggunakan Apache Spark tanpa harus merencanakan, mengonfigurasi, atau mengelola sumber daya. Ketika Anda menjalankan Apache Spark aplikasi di Athena, Anda mengirimkan Spark kode untuk diproses dan menerima hasilnya secara langsung.
- AWS Glue Data Catalog — AWS Glue adalah layanan integrasi data tanpa server yang memudahkan pengguna analitik untuk menemukan, menyiapkan, memindahkan, dan mengintegrasikan data dari berbagai sumber. Anda dapat menggunakannya AWS Glue untuk analitik, pembelajaran mesin, dan pengembangan aplikasi. AWS Glue Data Catalog adalah repositori terpusat yang menyimpan metadata tentang kumpulan data organisasi Anda. Ini bertindak sebagai indeks lokasi, skema, dan metrik run-time dari sumber data Anda.
- Pelatihan Model SageMaker Runtime Amazon — Amazon SageMaker Runtime adalah layanan pembelajaran mesin yang dikelola sepenuhnya. Dengan SageMaker Runtime, ilmuwan dan pengembang data dapat dengan cepat dan mudah membangun dan melatih model pembelajaran mesin, dan kemudian langsung menerapkannya ke lingkungan host yang siap produksi.

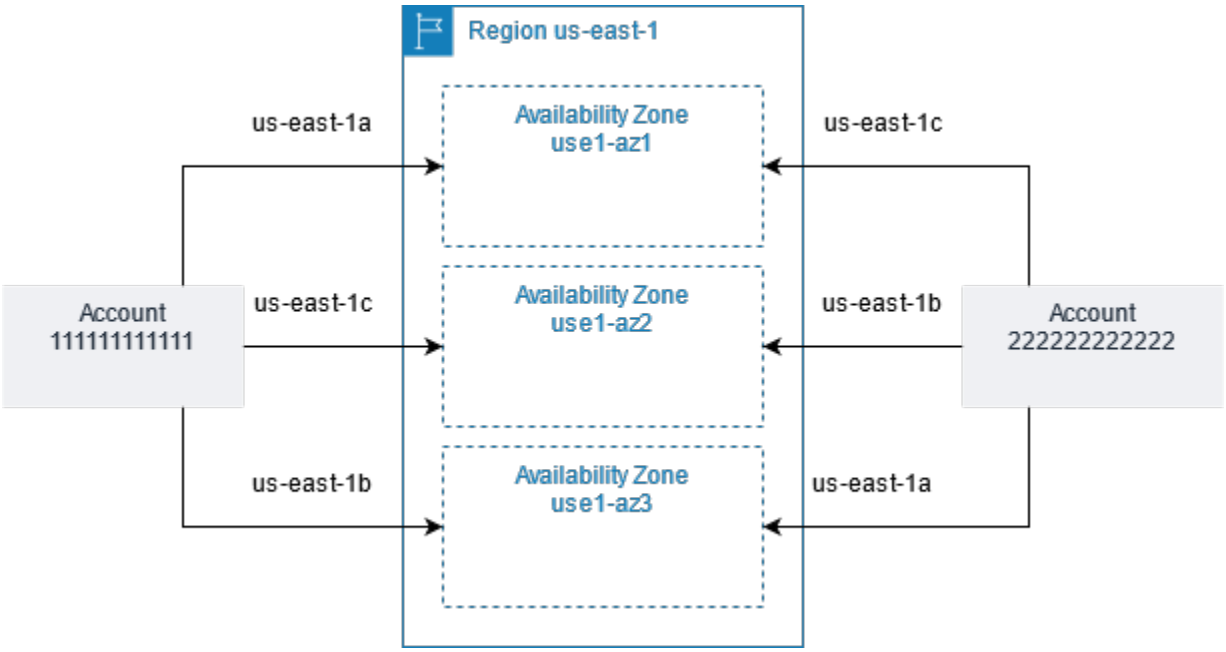
Untuk informasi lebih lanjut tentang S3 Express One Zone, lihat Apa itu S3 Express One Zone? dan Bagaimana S3 Express One Zone berbeda? .

Zona Ketersediaan dan Wilayah S3 Express One Zone

Zona Ketersediaan adalah satu atau lebih pusat data diskret dengan daya redundan, jaringan, dan konektivitas dalam Wilayah AWS. Untuk mengoptimalkan pengambilan latensi rendah, objek di kelas penyimpanan Amazon S3 Express One Zone disimpan secara berlebihan di bucket direktori S3 dalam satu Zona Ketersediaan yang bersifat lokal untuk beban kerja komputasi Anda. Saat membuat bucket direktori, Anda memilih Availability Zone dan Wilayah AWS lokasi bucket Anda.

AWS memetakan Availability Zone fisik secara acak ke nama Availability Zone untuk masing-masing Akun AWS nama. Pendekatan ini membantu mendistribusikan sumber daya di seluruh Availability Zone di Wilayah AWS, alih-alih sumber daya yang kemungkinan terkonsentrasi di Availability Zone pertama untuk setiap Wilayah. Akibatnya, Availability Zone `us-east-1a` untuk Anda Akun AWS mungkin tidak mewakili lokasi fisik yang sama dengan lokasi `us-east-1a` yang berbeda Akun AWS. Untuk informasi selengkapnya, lihat Wilayah dan Zona Ketersediaan di Panduan EC2 Pengguna Amazon.

Untuk mengoordinasikan Zona Ketersediaan di seluruh akun, Anda harus menggunakan ID AZ, yang merupakan pengenal unik dan konsisten untuk Zona Ketersediaan. Misalnya, `use1-az1` adalah ID AZ untuk `us-east-1` Wilayah dan memiliki lokasi fisik yang sama di setiap wilayah Akun AWS. Ilustrasi berikut menunjukkan bagaimana AZ IDs sama untuk setiap akun, meskipun nama Availability Zone mungkin dipetakan secara berbeda untuk setiap akun.



Dengan S3 Express One Zone, data Anda disimpan secara berlebihan di beberapa perangkat dalam satu Zona Ketersediaan. S3 Express One Zone dirancang untuk ketersediaan 99,95 persen dalam satu Availability Zone dan didukung oleh Perjanjian Tingkat Layanan Amazon S3. Untuk informasi selengkapnya, lihat Zona Ketersediaan

Tabel berikut menunjukkan Wilayah dan Availability Zone yang didukung S3 Express One Zone.

Nama wilayah	Kode Wilayah	ID Zona Ketersediaan				
Timur AS (N. Virginia)	us-east-1	use1-az4				
		use1-az5				
		use1-az6				
AS Timur (Ohio)	us-east-2	use2-az1				
		use2-az2				
AS Barat (Oregon)	us-west-2	usw2-az1				
		usw2-az3				
		usw2-az4				
Asia Pasifik (Mumbai)	ap-south-1	aps1-az1				
		aps1-az3				
Asia Pasifik (Tokyo)	ap-northeast-1	apne1-az1				
		apne1-az4				
Eropa (Irlandia)	eu-west-1	euw1-az1				
		euw1-az3				
Eropa (Stockholm)	eu-north-1	eun1-az1				
		eun1-az2				
		eun1-az3				

Jaringan untuk bucket direktori di Availability Zone

Topik berikut menjelaskan persyaratan jaringan untuk mengakses S3 Express One Zone dengan menggunakan titik akhir VPC gateway.

Topik

- Titik akhir untuk bucket direktori di Availability Zones
- Mengkonfigurasi titik akhir gateway VPC

Titik akhir untuk bucket direktori di Availability Zones

Tabel berikut menunjukkan titik akhir API Regional dan Zonal yang tersedia untuk setiap Wilayah dan Zona Ketersediaan.

Mengkonfigurasi titik akhir gateway VPC

Gunakan prosedur berikut untuk membuat titik akhir gateway yang terhubung ke objek kelas penyimpanan Amazon S3 Express One Zone dan bucket direktori.

Untuk mengonfigurasi titik akhir VPC gateway

1. Buka Konsol VPC Amazon.
2. Di panel navigasi, pilih Titik Akhir.
3. Pilih Buat Titik Akhir.
4. Buat nama untuk titik akhir Anda.
5. Untuk Kategori layanan, pilih Layanan AWS.
6. Untuk Layanan, tambahkan filter Type=Gateway dan kemudian pilih tombol opsi di sebelah com.amazonaws. ***region***.s3express.
7. Untuk VPC, pilih VPC yang akan digunakan untuk menciptakan titik akhir.
8. Untuk Tabel rute, pilih tabel rute yang akan digunakan oleh titik akhir. Amazon VPC secara otomatis menambahkan rute yang mengarahkan lalu lintas yang ditujukan untuk layanan ke antarmuka jaringan titik akhir.
9. Untuk Kebijakan, pilih Akses penuh untuk mengizinkan semua operasi oleh semua pengguna utama pada semua sumber daya melalui titik akhir VPC. Jika tidak, pilih Kustom untuk melampirkan kebijakan titik akhir VPC yang mengontrol izin yang dimiliki pengguna utama untuk melakukan tindakan pada sumber daya melalui titik akhir VPC.
10. (Opsional) Untuk menambahkan tanda, pilih Tambahkan tanda baru dan masukkan kunci dan nilai tanda.
11. Pilih Buat Titik Akhir.

Setelah membuat titik akhir gateway, Anda dapat menggunakan titik akhir API Regional dan titik akhir API Zonal untuk mengakses objek kelas penyimpanan Amazon S3 Express One Zone dan bucket direktori.

Membuat bucket direktori di Availability Zone

Untuk mulai menggunakan kelas penyimpanan Amazon S3 Express One Zone, Anda membuat bucket direktori. Kelas penyimpanan S3 Express One Zone hanya dapat digunakan dengan bucket direktori. Kelas penyimpanan S3 Express One Zone mendukung kasus penggunaan latensi rendah dan menyediakan pemrosesan data yang lebih cepat dalam satu Zona Ketersediaan. Jika aplikasi Anda sensitif terhadap kinerja dan mendapat manfaat dari latensi PUT dan GET satu digit milidetik, sebaiknya buat bucket direktori sehingga Anda dapat menggunakan kelas penyimpanan S3 Express One Zone.

Terdapat dua tipe bucket Amazon S3, bucket bertujuan umum dan bucket direktori. Anda harus memilih tipe bucket yang paling sesuai dengan aplikasi dan persyaratan kinerja Anda. Bucket bertujuan umum adalah tipe bucket S3 asli. Bucket tujuan umum direkomendasikan untuk sebagian besar kasus penggunaan dan pola akses dan memungkinkan objek disimpan di semua kelas penyimpanan, kecuali S3 Express One Zone. Untuk informasi selengkapnya tentang bucket tujuan umum, lihat Ikhtisar ember tujuan umum.

Bucket direktori menggunakan kelas penyimpanan S3 Express One Zone, yang dirancang agar digunakan untuk beban kerja atau aplikasi yang kritis kinerja yang memerlukan latensi milidetik satu digit yang konsisten. S3 Express One Zone adalah kelas penyimpanan S3 pertama di mana Anda dapat memilih satu Zona Ketersediaan dengan opsi untuk menempatkan bersama penyimpanan objek dengan sumber daya komputasi Anda, yang memberikan kecepatan akses setinggi mungkin. Saat membuat bucket direktori, Anda dapat secara opsional menentukan Wilayah AWS dan Availability Zone yang lokal untuk Amazon, Amazon Elastic Kubernetes Service EC2, atau Amazon Elastic Container Service (Amazon ECS) menghitung instans untuk mengoptimalkan kinerja.

Dengan S3 Express One Zone, data Anda disimpan secara berlebihan di beberapa perangkat dalam satu Zona Ketersediaan. S3 Express One Zone dirancang untuk ketersediaan 99,95 persen dalam satu Availability Zone dan didukung oleh Perjanjian Tingkat Layanan Amazon S3. Untuk informasi selengkapnya, lihat Zona Ketersediaan

Bucket direktori mengatur data secara hierarkis ke dalam direktori, sebagai lawan dari struktur penyimpanan datar dari bucket tujuan umum. Tidak ada batas prefiks untuk bucket direktori dan direktori individu dapat menskalakan secara horizontal.

Untuk informasi selengkapnya tentang bucket direktori, lihat Bekerja dengan bucket direktori.

Nama bucket direktori

Nama bucket direktori harus mengikuti format ini dan mematuhi aturan penamaan bucket direktori:

```
bucket-base-name--zone-id--x-s3
```

Misalnya, nama bucket direktori berikut berisi ID Zona Ketersediaan `usw2-az1`:

```
bucket-base-name--usw2-az1--x-s3
```

Untuk informasi selengkapnya tentang aturan penamaan bucket direktori, lihat Aturan penamaan bucket direktori.

Menggunakan konsol S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi di bagian atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah tempat Anda ingin membuat ember.

   > **Note**
   >
   > Untuk meminimalkan latensi dan biaya serta memenuhi persyaratan regulasi, pilih Wilayah yang dekat dengan Anda. Objek yang disimpan di Wilayah tidak pernah keluar dari Wilayah kecuali Anda secara tegas mentransfer atau mereplikasinya ke Wilayah lain. Untuk daftar Amazon S3 Wilayah AWS, lihat Layanan AWS titik akhir di. Referensi Umum Amazon Web Services

3. Di panel navigasi kiri, pilih Bucket direktori.
4. Pilih Buat bucket. Halaman Buat bucket terbuka.
5. Di bawah Konfigurasi umum, lihat Wilayah AWS tempat bucket Anda akan dibuat.

   Di bawah Jenis Bucket, pilih Direktori.

   > **Note**
   >
   > - Jika Anda memilih Region yang tidak mendukung bucket direktori, opsi jenis Bucket akan hilang, dan tipe bucket default ke bucket tujuan umum. Untuk membuat bucket

direktori, Anda harus memilih Region yang didukung. Untuk daftar Wilayah yang mendukung bucket direktori dan kelas penyimpanan Amazon S3 Express One Zone, lihat. the section called “Zona Ketersediaan dan Wilayah S3 Express One Zone”

- Setelah membuat bucket, Anda tidak dapat mengubah tipe bucket.

> **Note**
>
> Zona Ketersediaan tidak dapat diubah setelah bucket dibuat.

6. Untuk Zona Ketersediaan, pilih Zona Ketersediaan lokal untuk layanan komputasi Anda. Untuk daftar Availability Zones yang mendukung bucket direktori dan kelas penyimpanan S3 Express One Zone, lihat. the section called “Zona Ketersediaan dan Wilayah S3 Express One Zone”

   Di Availability Zone, pilih kotak centang untuk mengetahui bahwa jika terjadi pemadaman Availability Zone, data Anda mungkin tidak tersedia atau hilang.

   > **Important**
   >
   > Meskipun bucket direktori disimpan di beberapa perangkat dalam satu Availability Zone, bucket direktori tidak menyimpan data secara berlebihan di Availability Zone.

7. Untuk Nama bucket, masukkan nama untuk bucket direktori Anda.

   Aturan penamaan berikut berlaku untuk bucket direktori.

   - Jadilah unik dalam Zona yang dipilih (AWS Availability Zone atau AWS Local Zone).
   - Panjang nama harus antara 3 (menit) dan 63 (maks) karakter, termasuk akhiran.
   - Hanya terdiri dari huruf kecil, angka, tanda hubung (-).
   - Dimulai dan diakhiri dengan huruf atau angka.
   - Harus menyertakan akhiran berikut:`--`*`zone-id`*`--x-s3`.
   - Nama bucket tidak boleh dimulai dengan prefiks `xn--`.
   - Nama bucket tidak boleh dimulai dengan prefiks `sthree-`.
   - Nama bucket tidak boleh dimulai dengan prefiks `sthree-configurator`.
   - Nama bucket tidak boleh dimulai dengan prefiks `amzn-s3-demo-`.

- Nama bucket tidak boleh diakhiri dengan sufiks `-s3alias`. Sufiks ini dicadangkan untuk nama alias titik akses. Untuk informasi selengkapnya, lihat Alias titik akses.
- Nama bucket tidak boleh diakhiri dengan sufiks `--ol-s3`. Sufiks ini dicadangkan untuk nama alias Titik Akses Lambda Objek. Untuk informasi selengkapnya, lihat Cara menggunakan alias gaya bucket untuk bucket S3 Anda Titik Akses Lambda Objek.
- Nama bucket tidak boleh diakhiri dengan sufiks `.mrap`. Akhiran ini dicadangkan untuk nama Titik Akses Multi-Wilayah. Untuk informasi selengkapnya, lihat Aturan penamaan Titik Akses Multi-Wilayah Amazon S3.

Sufiks secara otomatis ditambahkan ke nama dasar yang Anda berikan saat membuat bucket direktori menggunakan konsol. Sufiks ini mencakup ID Zona Ketersediaan dari Zona Ketersediaan yang Anda pilih.

Setelah membuat bucket, Anda tidak dapat mengubah namanya. Untuk informasi selengkapnya tentang penamaan bucket, lihat Aturan penamaan bucket tujuan umum.

> **Important**
>
> Jangan sertakan informasi sensitif, seperti nomor akun, dalam nama bucket. Nama ember terlihat di titik URLs itu ke objek di ember.

8. Di bawah Kepemilikan Objek, setelan yang diberlakukan pemilik Bucket diaktifkan secara otomatis, dan semua daftar kontrol akses (ACLs) dinonaktifkan. Untuk bucket direktori, tidak ACLs dapat diaktifkan.

   Pemilik bucket diberlakukan (default) — ACLs dinonaktifkan, dan pemilik bucket secara otomatis memiliki dan memiliki kontrol penuh atas setiap objek di bucket tujuan umum. ACLs tidak lagi memengaruhi izin akses ke data di bucket tujuan umum S3. Bucket menggunakan kebijakan secara eksklusif untuk menentukan kontrol akses.

9. Di bawah pengaturan Blokir Akses Publik untuk bucket ini, semua setelan Blokir Akses Publik untuk bucket direktori Anda diaktifkan secara otomatis. Pengaturan ini tidak dapat dimodifikasi untuk bucket direktori. Untuk informasi lebih lanjut tentang pemblokiran akses publik, lihat Melakukan blok akses publik ke penyimpanan Amazon S3 Anda.
10. Untuk mengonfigurasi enkripsi default, di bawah Jenis enkripsi, pilih salah satu dari berikut ini:

    - Enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3)

- Enkripsi sisi server dengan AWS Key Management Service kunci (SSE-KMS)

Untuk informasi selengkapnya tentang penggunaan enkripsi di sisi server Amazon S3 guna mengenkripsi data Anda, lihat Perlindungan data dan enkripsi.

> **Important**
>
> Jika Anda menggunakan opsi SSE-KMS untuk konfigurasi enkripsi default, Anda tunduk pada kuota permintaan per detik (RPS) AWS KMS. Untuk informasi selengkapnya tentang kuota AWS KMS dan cara meminta kenaikan kuota, lihat Kuota di Panduan Pengembang AWS Key Management Service .
> Saat mengaktifkan enkripsi default, Anda mungkin perlu memperbarui kebijakan bucket. Untuk informasi selengkapnya, lihat Menggunakan enkripsi SSE-KMS untuk operasi lintas akun.

11. Jika Anda memilih enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3), di bawah Kunci Bucket, Diaktifkan akan muncul. Kunci Bucket S3 selalu diaktifkan saat Anda mengonfigurasi bucket direktori untuk menggunakan enkripsi default dengan SSE-S3. Kunci Bucket S3 selalu diaktifkan untuk `GET` dan `PUT` beroperasi di bucket direktori dan tidak dapat dinonaktifkan. Kunci Bucket S3 tidak didukung, saat Anda menyalin objek terenkripsi SSE-KMS dari bucket tujuan umum ke bucket direktori, dari bucket direktori ke bucket tujuan umum, atau di antara bucket direktori, melalui, operasi dalam Operasi Batch, atau pekerjaan CopyObject. UploadPartCopyCopyimport Dalam hal ini, Amazon S3 melakukan panggilan ke AWS KMS setiap kali permintaan salinan dibuat untuk objek yang dienkripsi KMS.

    S3 Bucket Keys menurunkan biaya enkripsi dengan mengurangi lalu lintas permintaan dari Amazon S3 ke. AWS KMS Untuk informasi selengkapnya, lihat Mengurangi biaya SSE-KMS dengan Kunci Bucket Amazon S3.

12. Jika Anda memilih enkripsi sisi server dengan AWS Key Management Service kunci (SSE-KMS), di bawah AWS KMS kunci, tentukan AWS Key Management Service kunci Anda dengan salah satu cara berikut atau buat kunci baru.

    - Untuk memilih dari daftar kunci KMS yang tersedia, pilih Pilih dari Anda AWS KMS keys, dan pilih kunci KMS Anda dari Tersedia. AWS KMS keys

      Hanya kunci terkelola pelanggan Anda yang muncul di daftar ini. The Kunci yang dikelola AWS (`aws/s3`) tidak didukung dalam bucket direktori. Untuk informasi selengkapnya tentang

CMK, lihat Kunci pelanggan dan AWS kunci di AWS Key Management Service Panduan Pengembang.

- Untuk memasukkan tombol KMS ARN atau alias, pilih Masukkan ARN, dan masukkan kunci KMS AWS KMS key ARN atau alias Anda di ARN.AWS KMS key
- Untuk membuat kunci terkelola pelanggan baru di AWS KMS konsol, pilih Buat kunci KMS.

  Untuk informasi selengkapnya tentang membuat AWS KMS key, lihat Membuat kunci di Panduan AWS Key Management Service Pengembang.

> **Important**
>
> - Konfigurasi SSE-KMS Anda hanya dapat mendukung 1 kunci terkelola pelanggan per bucket direktori selama masa pakai bucket. Kunci yang dikelola AWS(`aws/s3`) tidak didukung. Selain itu, setelah menentukan kunci terkelola pelanggan untuk SSE-KMS, Anda tidak dapat mengganti kunci terkelola pelanggan untuk konfigurasi SSE-KMS bucket.
>
>   Anda dapat mengidentifikasi kunci terkelola pelanggan yang Anda tentukan untuk konfigurasi SSE-KMS bucket, dengan cara berikut:
>
>   - Anda membuat permintaan operasi `HeadObject` API untuk menemukan nilai `x-amz-server-side-encryption-aws-kms-key-id` dalam respons Anda.
>
>   Untuk menggunakan kunci terkelola pelanggan baru untuk data Anda, sebaiknya salin objek yang ada ke bucket direktori baru dengan kunci terkelola pelanggan baru.
> - Anda hanya dapat menggunakan tombol KMS yang tersedia Wilayah AWS sama dengan bucket. Konsol Amazon S3 hanya mencantumkan kunci 100 KMS pertama di Wilayah yang sama dengan bucket. Untuk menggunakan kunci KMS yang tidak terdaftar, Anda harus memasukkan ARN kunci KMS Anda. Jika Anda ingin menggunakan kunci KMS yang dimiliki oleh akun lain, Anda harus terlebih dahulu memiliki izin untuk menggunakan kunci tersebut, dan kemudian Anda harus memasukkan ARN kunci KMS. Untuk informasi selengkapnya tentang izin lintas akun untuk kunci KMS, lihat Membuat kunci KMS yang dapat digunakan akun lain di Panduan Pengembang AWS Key Management Service . Untuk informasi selengkapnya tentang SSE-KMS, lihat Menentukan enkripsi sisi server dengan AWS KMS (SSE-KMS) untuk unggahan objek baru di bucket direktori.

- Bila Anda menggunakan enkripsi sisi server AWS KMS key untuk bucket direktori, Anda harus memilih kunci KMS enkripsi simetris. Amazon S3 hanya mendukung kunci KMS enkripsi simetris dan tidak mendukung kunci KMS asimetris. Untuk informasi selengkapnya, lihat Mengidentifikasi tombol KMS simetris dan asimetris dalam Panduan Pengembang AWS Key Management Service .

Untuk informasi selengkapnya tentang penggunaan AWS KMS dengan Amazon S3, lihat. Menggunakan enkripsi sisi server dengan AWS KMS kunci (SSE-KMS) di bucket direktori

13. Pilih Buat bucket. Setelah membuat bucket, Anda dapat menambahkan file dan folder ke bucket. Untuk informasi selengkapnya, lihat the section called “Bekerja dengan objek di dalam bucket direktori”.

Menggunakan AWS SDKs

SDK for Go

Contoh ini menunjukkan cara membuat bucket direktori dengan menggunakan file AWS SDK untuk Go.

Example

```
var bucket = "..."

func runCreateBucket(c *s3.Client) {
    resp, err := c.CreateBucket(context.Background(), &s3.CreateBucketInput{
        Bucket: &bucket,
        CreateBucketConfiguration: &types.CreateBucketConfiguration{
            Location: &types.LocationInfo{
                Name: aws.String("usw2-az1"),
                Type: types.LocationTypeAvailabilityZone,
            },
            Bucket: &types.BucketInfo{
                DataRedundancy: types.DataRedundancySingleAvailabilityZone,
                Type:           types.BucketTypeDirectory,
            },
        },
    })
    var terr *types.BucketAlreadyOwnedByYou
    if errors.As(err, &terr) {
```

```
        fmt.Printf("BucketAlreadyOwnedByYou: %s\n", aws.ToString(terr.Message))
        fmt.Printf("noop...\n")
        return
    }
    if err != nil {
        log.Fatal(err)
    }

    fmt.Printf("bucket created at %s\n", aws.ToString(resp.Location))
}
```

SDK for Java 2.x

Contoh ini menunjukkan cara membuat bucket direktori dengan menggunakan file AWS SDK for Java 2.x.

Example

```
public static void createBucket(S3Client s3Client, String bucketName) {

    //Bucket name format is {base-bucket-name}--{az-id}--x-s3
    //example: doc-example-bucket--usw2-az1--x-s3 is a valid name for a directory
 bucket created in
    //Region us-west-2, Availability Zone 2

    CreateBucketConfiguration bucketConfiguration =
 CreateBucketConfiguration.builder()
             .location(LocationInfo.builder()
                     .type(LocationType.AVAILABILITY_ZONE)
                     .name("usw2-az1").build()) //this must match the Region and
 Availability Zone in your bucket name
             .bucket(BucketInfo.builder()
                    .type(BucketType.DIRECTORY)
                    .dataRedundancy(DataRedundancy.SINGLE_AVAILABILITY_ZONE)
                    .build()).build();
    try {

             CreateBucketRequest bucketRequest =
 CreateBucketRequest.builder().bucket(bucketName).createBucketConfiguration(bucketConfigurat
             CreateBucketResponse response = s3Client.createBucket(bucketRequest);
             System.out.println(response);
    }

    catch (S3Exception e) {
```

```
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

AWS SDK untuk JavaScript

Contoh ini menunjukkan cara membuat bucket direktori dengan menggunakan file AWS SDK untuk JavaScript.

Example

```
// file.mjs, run with Node.js v16 or higher
// To use with the preview build, place this in a folder
// inside the preview build directory, such as /aws-sdk-js-v3/workspace/

import { S3 } from "@aws-sdk/client-s3";

const region = "us-east-1";
const zone = "use1-az4";
const suffix = `${zone}--x-s3`;

const s3 = new S3({ region });

const bucketName = `...--${suffix}`;

const createResponse = await s3.createBucket(
    { Bucket: bucketName,
      CreateBucketConfiguration: {Location: {Type: "AvailabilityZone", Name: zone},
      Bucket: { Type: "Directory", DataRedundancy: "SingleAvailabilityZone" }}
    }
   );
```

SDK untuk .NET

Contoh ini menunjukkan cara membuat bucket direktori dengan menggunakan file SDK untuk .NET.

Example

```
using (var amazonS3Client = new AmazonS3Client())
```

```
{
    var putBucketResponse = await amazonS3Client.PutBucketAsync(new PutBucketRequest
    {

       BucketName = "DOC-EXAMPLE-BUCKET--usw2-az1--x-s3",
       PutBucketConfiguration = new PutBucketConfiguration
       {
         BucketInfo = new BucketInfo { DataRedundancy =
 DataRedundancy.SingleAvailabilityZone, Type = BucketType.Directory },
         Location = new LocationInfo { Name = "usw2-az1", Type =
 LocationType.AvailabilityZone }
       }
     }).ConfigureAwait(false);
}
```

SDK for PHP

Contoh ini menunjukkan cara membuat bucket direktori dengan menggunakan file AWS SDK for PHP.

Example

```
require 'vendor/autoload.php';

$s3Client = new S3Client([

    'region'      => 'us-east-1',
]);


$result = $s3Client->createBucket([
    'Bucket' => 'doc-example-bucket--use1-az4--x-s3',
    'CreateBucketConfiguration' => [
        'Location' => ['Name'=> 'use1-az4', 'Type'=> 'AvailabilityZone'],
        'Bucket' => ["DataRedundancy" => "SingleAvailabilityZone" ,"Type" =>
 "Directory"]   ],
]);
```

SDK for Python

Contoh ini menunjukkan cara membuat bucket direktori dengan menggunakan file AWS SDK untuk Python (Boto3).

Example

```
import logging
import boto3
from botocore.exceptions import ClientError

def create_bucket(s3_client, bucket_name, availability_zone):
    '''
    Create a directory bucket in a specified Availability Zone

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket to create; for example, 'doc-example-bucket--usw2-
az1--x-s3'
    :param availability_zone: String; Availability Zone ID to create the bucket in,
 for example, 'usw2-az1'
    :return: True if bucket is created, else False
    '''

    try:
        bucket_config = {
                'Location': {
                    'Type': 'AvailabilityZone',
                    'Name': availability_zone
                },
                'Bucket': {
                    'Type': 'Directory',
                    'DataRedundancy': 'SingleAvailabilityZone'
                }
            }
        s3_client.create_bucket(
            Bucket = bucket_name,
            CreateBucketConfiguration = bucket_config
        )
    except ClientError as e:
        logging.error(e)
        return False
    return True


if __name__ == '__main__':
    bucket_name = 'BUCKET_NAME'
    region = 'us-west-2'
    availability_zone = 'usw2-az1'
    s3_client = boto3.client('s3', region_name = region)
```

```
    create_bucket(s3_client, bucket_name, availability_zone)
```

SDK for Ruby

Contoh ini menunjukkan cara membuat bucket direktori dengan menggunakan file AWS SDK untuk Ruby.

Example

```
s3 = Aws::S3::Client.new(region:'us-west-2')
s3.create_bucket(
  bucket: "bucket_base_name--az_id--x-s3",
  create_bucket_configuration: {
    location: { name: 'usw2-az1', type: 'AvailabilityZone' },
    bucket: { data_redundancy: 'SingleAvailabilityZone', type: 'Directory' }
  }
)
```

Menggunakan AWS CLI

Contoh ini menunjukkan cara membuat bucket direktori dengan menggunakan file AWS CLI. Untuk menggunakan perintah ganti *user input placeholders* dengan informasi Anda sendiri.

Saat membuat bucket direktori, Anda harus memberikan detail konfigurasi dan menggunakan konvensi penamaan berikut: *bucket-base-name*--*zone-id*--x-s3

```
aws s3api create-bucket
--bucket bucket-base-name--zone-id--x-s3
--create-bucket-configuration 'Location={Type=AvailabilityZone,Name=usw2-
az1},Bucket={DataRedundancy=SingleAvailabilityZone,Type=Directory}'
--region us-west-2
```

Untuk informasi lebih lanjut, lihat create-bucket di. AWS Command Line Interface

Titik akhir regional dan Zonal untuk bucket direktori di Availability Zone

Untuk mengakses objek dan bucket direktori yang disimpan di S3 Express One Zone, Anda menggunakan titik akhir VPC gateway. Bucket direktori menggunakan titik akhir API Regional dan Zonal. Bergantung pada operasi Amazon S3 API yang Anda gunakan, titik akhir Regional atau Zonal diperlukan. Tidak dikenakan biaya tambahan untuk menggunakan titik akhir gateway.

Operasi API tingkat bucket (atau bidang kontrol) tersedia melalui titik akhir Regional dan disebut sebagai operasi API titik akhir Regional. Contoh operasi API titik akhir Regional adalah `CreateBucket` dan `DeleteBucket`.

Saat membuat bucket direktori yang disimpan di S3 Express One Zone, Anda memilih Availability Zone tempat bucket Anda akan berada. Anda dapat menggunakan operasi API titik akhir Zonal untuk mengunggah dan mengelola objek di bucket direktori Anda.

Operasi API tingkat objek (atau bidang data) tersedia melalui titik akhir Zonal dan disebut sebagai operasi API titik akhir Zonal. Contoh operasi API titik akhir Zonal adalah `CreateSession` dan `PutObject`.

Mengoptimalkan kinerja S3 Express One Zone

Amazon S3 Express One Zone adalah kelas penyimpanan S3 Zona Ketersediaan (AZ) tunggal berkinerja tinggi yang dibuat khusus untuk menghadirkan akses data milidetik satu digit yang konsisten untuk aplikasi Anda yang paling sensitif terhadap latensi. S3 Express One Zone adalah kelas penyimpanan S3 pertama yang memberi Anda opsi untuk mencari sumber daya penyimpanan objek dan AWS komputasi berkinerja tinggi, seperti Amazon Elastic Compute Cloud, Amazon Elastic Kubernetes Service, dan Amazon Elastic Container Service, dalam satu Availability Zone. Lokasi bersama penyimpanan dan sumber daya komputasi Anda mengoptimalkan kinerja dan biaya komputasi serta memberikan peningkatan kecepatan pemrosesan data.

S3 Express One Zone memberikan elastisitas kinerja yang serupa dengan kelas penyimpanan S3 lainnya, tetapi dengan latensi permintaan baca dan tulis satu digit milidetik yang konsisten—hingga 10x lebih cepat daripada Standar S3. S3 Express One Zone dirancang dari bawah ke atas untuk mendukung throughput burst hingga level agregat yang sangat tinggi. Kelas penyimpanan S3 Express One Zone menggunakan arsitektur yang dibuat khusus untuk mengoptimalkan kinerja dan memberikan latensi permintaan rendah secara konsisten dengan menyimpan data pada perangkat keras berkinerja tinggi. Protokol objek untuk S3 Express One Zone telah ditingkatkan untuk melancarkan autentikasi dan overhead metadata.

Untuk lebih mengurangi latensi dan mendukung hingga 2 juta pembacaan dan hingga 200.000 penulisan per detik, S3 Express One Zone menyimpan data dalam bucket direktori Amazon S3. Secara default, setiap bucket direktori mendukung hingga 200.000 pembacaan dan hingga 100.000 penulisan per detik. Jika beban kerja Anda membutuhkan lebih tinggi dari batas TPS default, Anda dapat meminta peningkatan melalui Support AWS .

Kombinasi perangkat keras dan perangkat lunak berkinerja tinggi yang dibuat khusus yang memberikan kecepatan akses data milidetik satu digit dan bucket direktori yang diskalakan untuk

sejumlah besar transaksi per detik menjadikan S3 Express One Zone sebagai kelas penyimpanan Amazon S3 terbaik untuk operasi intensif permintaan atau aplikasi yang kritis kinerja.

Topik berikut menjelaskan panduan praktik terbaik dan pola desain untuk mengoptimalkan kinerja aplikasi yang menggunakan kelas penyimpanan S3 Express One Zone.

Topik

- Praktik terbaik untuk mengoptimalkan kinerja S3 Express One Zone

Praktik terbaik untuk mengoptimalkan kinerja S3 Express One Zone

Saat membangun aplikasi yang mengunggah dan mengambil objek dari Amazon S3 Express One Zone, ikuti panduan praktik terbaik kami untuk mengoptimalkan kinerja. Untuk menggunakan kelas penyimpanan S3 Express One Zone, Anda harus membuat bucket direktori S3. Kelas penyimpanan S3 Express One Zone tidak didukung untuk digunakan dengan bucket tujuan umum S3.

Untuk panduan kinerja semua kelas penyimpanan Amazon S3 lainnya dan bucket tujuan umum S3, lihat Pola desain praktik terbaik: mengoptimalkan performa Amazon S3.

Untuk kinerja dan skalabilitas yang optimal dengan kelas penyimpanan S3 Express One Zone dan bucket direktori dalam beban kerja skala tinggi, penting untuk memahami cara kerja bucket direktori secara berbeda dari bucket tujuan umum. Kemudian, kami menyediakan praktik terbaik untuk menyelaraskan aplikasi Anda dengan cara kerja bucket direktori.

Cara kerja bucket direktori

Kelas penyimpanan Amazon S3 Express One Zone dapat mendukung beban kerja hingga 2.000.000 GET dan hingga 200.000 transaksi PUT per detik (TPS) per bucket direktori. Dengan S3 Express One Zone, data disimpan dalam bucket direktori S3 di Availability Zones. Objek dalam bucket direktori dapat diakses dalam namespace hierarkis, mirip dengan sistem file dan berbeda dengan bucket tujuan umum S3 yang memiliki namespace datar. Tidak seperti bucket tujuan umum, bucket direktori mengatur kunci secara hierarkis ke dalam direktori alih-alih prefiks. Prefiks adalah string karakter di bagian awal nama kunci objek. Anda dapat menggunakan awalan untuk mengatur data Anda dan mengelola arsitektur penyimpanan objek datar di bucket tujuan umum. Untuk informasi selengkapnya, lihat Organisasi objek menggunakan prefiks.

Dalam bucket direktori, objek diatur dalam namespace hierarkis menggunakan garis miring maju (/) sebagai satu-satunya pembatas yang didukung. Saat Anda mengunggah objek dengan kunci seperti`dir1/dir2/file1.txt`, direktori `dir1/` dan secara otomatis `dir2/` dibuat dan

dikelola oleh Amazon S3. Direktori dibuat selama `PutObject` atau `CreateMultiPartUpload` operasi dan secara otomatis dihapus ketika mereka menjadi kosong setelah `DeleteObject` atau `AbortMultiPartUpload` operasi. Tidak ada batas atas jumlah objek dan subdirektori dalam direktori.

Direktori yang dibuat saat objek diunggah ke bucket direktori dapat diskalakan secara instan untuk mengurangi kemungkinan kesalahan HTTP. `503 (Slow Down)` Penskalaan otomatis ini memungkinkan aplikasi Anda untuk memparalelkan permintaan baca dan tulis di dalam dan di seluruh direktori sesuai kebutuhan. Untuk S3 Express One Zone, direktori individual dirancang untuk mendukung tingkat permintaan maksimum bucket direktori. Tidak perlu mengacak awalan kunci untuk mencapai kinerja optimal karena sistem secara otomatis mendistribusikan objek untuk distribusi beban yang merata, tetapi sebagai hasilnya, kunci tidak disimpan secara leksikografis dalam ember direktori. Ini berbeda dengan bucket tujuan umum S3 di mana kunci yang lebih dekat secara leksikografis lebih mungkin ditempatkan bersama di server yang sama.

Untuk informasi selengkapnya tentang contoh operasi bucket direktori dan interaksi direktori, lihatOperasi bucket direktori dan contoh interaksi direktori.

Praktik terbaik

Ikuti praktik terbaik untuk mengoptimalkan kinerja bucket direktori Anda dan membantu beban kerja Anda menskalakan dari waktu ke waktu.

Gunakan direktori yang berisi banyak entri (objek atau subdirektori)

Bucket direktori memberikan kinerja tinggi secara default untuk semua beban kerja. Untuk optimasi kinerja yang lebih besar dengan operasi tertentu, mengkonsolidasikan lebih banyak entri (yang merupakan objek atau subdirektori) ke dalam direktori akan menghasilkan latensi yang lebih rendah dan tingkat permintaan yang lebih tinggi:

- Operasi API yang bermutasi, sepertiPutObject,, `CreateMultiPartUpload` dan `DeleteObjectAbortMultiPartUpload`, mencapai kinerja optimal saat diimplementasikan dengan direktori yang lebih sedikit dan lebih padat yang berisi ribuan entri, bukan dengan sejumlah besar direktori yang lebih kecil.
- `ListObjectsV2`operasi bekerja lebih baik ketika lebih sedikit direktori perlu dilalui untuk mengisi halaman hasil.

Jangan gunakan entropi dalam awalan

Dalam operasi Amazon S3, entropi mengacu pada keacakan dalam penamaan awalan yang membantu mendistribusikan beban kerja secara merata di seluruh partisi penyimpanan. Namun, karena bucket direktori mengelola distribusi beban secara internal, tidak disarankan untuk menggunakan entropi dalam awalan untuk kinerja terbaik. Ini karena untuk bucket direktori, entropi dapat menyebabkan permintaan menjadi lebih lambat dengan tidak menggunakan kembali direktori yang telah dibuat.

Pola kunci seperti `$HASH/directory/object` bisa berakhir dengan membuat banyak direktori perantara. Dalam contoh berikut, semua `job-1` s adalah direktori yang berbeda karena orang tua mereka berbeda. Direktori akan jarang dan mutasi dan permintaan daftar akan lebih lambat. Dalam contoh ini ada 12 Direktori perantara yang semuanya memiliki satu entri.

```
s3://my-bucket/0cc175b9c0f1b6a831c399e269772661/job-1/file1

s3://my-bucket/92eb5ffee6ae2fec3ad71c777531578f/job-1/file2

s3://my-bucket/4a8a08f09d37b73795649038408b5f33/job-1/file3

s3://my-bucket/8277e0910d750195b448797616e091ad/job-1/file4

s3://my-bucket/e1671797c52e15f763380b45e841ec32/job-1/file5

s3://my-bucket/8fa14cdd754f91cc6554c9e71929cce7/job-1/file6
```

Sebagai gantinya, untuk kinerja yang lebih baik, kita dapat menghapus `$HASH` komponen dan memungkinkan `job-1` untuk menjadi satu direktori, meningkatkan kepadatan direktori. Dalam contoh berikut, direktori perantara tunggal yang memiliki 6 entri dapat menghasilkan kinerja yang lebih baik, dibandingkan dengan contoh sebelumnya.

```
s3://my-bucket/job-1/file1

s3://my-bucket/job-1/file2

s3://my-bucket/job-1/file3

s3://my-bucket/job-1/file4

s3://my-bucket/job-1/file5
```

```
s3://my-bucket/job-1/file6
```

Keuntungan kinerja ini terjadi karena ketika kunci objek awalnya dibuat dan nama kuncinya termasuk direktori, direktori secara otomatis dibuat untuk objek. Unggahan objek berikutnya ke direktori yang sama tidak memerlukan direktori yang akan dibuat, yang mengurangi latensi pada unggahan objek ke direktori yang ada.

Gunakan pemisah selain pembatas/untuk memisahkan bagian kunci Anda jika Anda tidak memerlukan kemampuan untuk mengelompokkan objek secara logis selama panggilan **`ListObjectsV2`**

Karena `/` pembatas diperlakukan khusus untuk ember direktori, itu harus digunakan dengan niat. Sementara bucket direktori tidak secara leksikografis mengurutkan objek, objek dalam direktori masih dikelompokkan bersama dalam output. `ListObjectsV2` Jika Anda tidak memerlukan fungsi ini, Anda dapat mengganti `/` dengan karakter lain sebagai pemisah untuk tidak menyebabkan pembuatan direktori perantara.

Misalnya, asumsikan kunci berikut berada dalam `YYYY/MM/DD/HH/` pola awalan

```
s3://my-bucket/2024/04/00/01/file1

s3://my-bucket/2024/04/00/02/file2

s3://my-bucket/2024/04/00/03/file3

s3://my-bucket/2024/04/01/01/file4

s3://my-bucket/2024/04/01/02/file5

s3://my-bucket/2024/04/01/03/file6
```

Jika Anda tidak perlu mengelompokkan objek berdasarkan jam atau hari dalam `ListObjectsV2` hasil, tetapi Anda perlu mengelompokkan objek berdasarkan bulan, pola kunci berikut `YYYY/MM/DD-HH-` akan menghasilkan direktori yang jauh lebih sedikit dan kinerja yang lebih baik untuk `ListObjectsV2` operasi.

```
s3://my-bucket/2024/04/00-01-file1

s3://my-bucket/2024/04/00-01-file2
```

```
s3://my-bucket/2024/04/00-01-file3

s3://my-bucket/2024/04/01-02-file4

s3://my-bucket/2024/04/01-02-file5

s3://my-bucket/2024/04/01-02-file6
```

Gunakan operasi daftar yang dibatasi jika memungkinkan

`ListObjectsV2`Permintaan tanpa `delimiter` melakukan traversal rekursif kedalaman pertama dari semua direktori. `ListObjectsV2`Permintaan dengan hanya `delimiter` mengambil entri dalam direktori yang ditentukan oleh `prefix` parameter, mengurangi latensi permintaan dan meningkatkan kunci agregat per detik. Untuk bucket direktori, gunakan operasi daftar terbatas jika memungkinkan. Daftar yang dibatasi menghasilkan direktori yang dikunjungi lebih sedikit kali, yang mengarah ke lebih banyak kunci per detik dan latensi permintaan yang lebih rendah.

Misalnya, untuk direktori dan objek berikut di bucket direktori Anda:

```
s3://my-bucket/2024/04/12-01-file1

s3://my-bucket/2024/04/12-01-file2

...

s3://my-bucket/2024/05/12-01-file1

s3://my-bucket/2024/05/12-01-file2

...

s3://my-bucket/2024/06/12-01-file1

s3://my-bucket/2024/06/12-01-file2

...

s3://my-bucket/2024/07/12-01-file1

s3://my-bucket/2024/07/12-01-file2

...
```

Untuk `ListObjectsV2` kinerja yang lebih baik, gunakan daftar terbatas untuk daftar subdirektori dan objek Anda, jika logika aplikasi Anda memungkinkan untuk itu. Misalnya, Anda dapat menjalankan perintah berikut untuk operasi daftar yang dibatasi,

```
aws s3api list-objects-v2 --bucket my-bucket --prefix '2024/' --delimiter '/'
```

Outputnya adalah daftar subdirektori.

```
{
    "CommonPrefixes": [
        {
            "Prefix": "2024/04/"
        },
        {
            "Prefix": "2024/05/"
        },
        {
            "Prefix": "2024/06/"
        },
        {
            "Prefix": "2024/07/"
        }
    ]
}
```

Untuk membuat daftar setiap subdirektori dengan kinerja yang lebih baik, Anda dapat menjalankan perintah seperti contoh berikut:

Perintah:

```
aws s3api list-objects-v2 --bucket my-bucket --prefix '2024/04' --delimiter '/'
```

Output:

```
{
    "Contents": [
        {
            "Key": "2024/04/12-01-file1"
        },
        {
            "Key": "2024/04/12-01-file2"
        }
```

```
    ]
}
```

Lokasi bersama penyimpanan S3 Express One Zone dengan sumber daya komputasi Anda

Dengan S3 Express One Zone, setiap bucket direktori terletak di satu Availability Zone yang Anda pilih saat membuat bucket. Anda dapat memulai dengan membuat bucket direktori baru di Zona Ketersediaan lokal untuk beban kerja atau sumber daya komputasi Anda. Anda kemudian dapat segera memulai membaca dan menulis latensi sangat rendah. Bucket direktori adalah jenis bucket S3 tempat Anda dapat memilih Availability Zone Wilayah AWS untuk mengurangi latensi antara komputasi dan penyimpanan.

Jika Anda mengakses bucket direktori di Availability Zones, Anda mungkin mengalami sedikit peningkatan latensi. Untuk mengoptimalkan performa, kami menyarankan agar Anda sebisa mungkin mengakses bucket direktori dari instans Amazon Elastic Container Service, Amazon Elastic Kubernetes Service, dan Amazon Elastic Compute Cloud yang berada di Zona Ketersediaan yang sama.

Gunakan koneksi bersamaan untuk mencapai throughput tinggi dengan objek di atas 1MB

Anda dapat mencapai kinerja terbaik dengan mengeluarkan beberapa permintaan sekaligus ke bucket direktori untuk menyebarkan permintaan Anda melalui koneksi terpisah untuk memaksimalkan bandwidth yang dapat diakses. Seperti bucket tujuan umum, S3 Express One Zone tidak memiliki batasan jumlah koneksi yang dibuat ke bucket direktori Anda. Direktori individu dapat menskalakan kinerja secara horizontal dan otomatis ketika terjadi sejumlah besar penulisan bersamaan ke direktori yang sama.

Koneksi TCP individual ke bucket direktori memiliki batas atas tetap pada jumlah byte yang dapat diunggah atau diunduh per detik. Ketika objek menjadi lebih besar, waktu permintaan menjadi didominasi oleh streaming byte daripada pemrosesan transaksi. Untuk menggunakan beberapa koneksi untuk memparalelkan unggahan atau pengunduhan objek yang lebih besar, Anda dapat mengurangi end-to-end latensi. Jika menggunakan `Java 2.x` SDK, Anda harus mempertimbangkan untuk menggunakan S3 Transfer Manager yang akan memanfaatkan peningkatan kinerja seperti operasi API unggahan multibagian dan pengambilan rentang byte untuk mengakses data secara paralel.

Gunakan titik akhir VPC Gateway

Titik akhir Gateway menyediakan koneksi langsung dari VPC Anda ke bucket direktori, tanpa memerlukan gateway internet atau perangkat NAT untuk VPC Anda. Untuk mengurangi jumlah waktu

yang dihabiskan paket Anda di jaringan, Anda harus mengonfigurasi VPC Anda dengan titik akhir VPC gateway untuk bucket direktori. Untuk informasi selengkapnya, lihat Jaringan untuk ember direktori.

Gunakan otentikasi sesi dan gunakan kembali token sesi saat valid

Bucket direktori menyediakan mekanisme otentikasi token sesi untuk mengurangi latensi pada operasi API yang sensitif terhadap kinerja. Anda dapat melakukan satu panggilan `CreateSession` untuk mendapatkan token sesi yang kemudian berlaku untuk semua permintaan dalam 5 menit berikutnya. Untuk mendapatkan latensi terendah dalam panggilan API Anda, pastikan untuk memperoleh token sesi dan menggunakannya kembali selama masa pakai token tersebut sebelum menyegarkannya.

Jika Anda menggunakan AWS SDKs, SDKs tangani penyegaran token sesi secara otomatis untuk menghindari gangguan layanan saat sesi berakhir. Kami menyarankan Anda menggunakan AWS SDKs untuk memulai dan mengelola permintaan ke operasi `CreateSession` API.

Untuk informasi selengkapnya tentang `CreateSession`, lihat Mengotorisasi operasi API titik akhir Zonal dengan `CreateSession`.

Gunakan klien berbasis CRT

AWS Common Runtime (CRT) adalah seperangkat pustaka modular, berkinerja, dan efisien yang ditulis dalam C dan dimaksudkan untuk bertindak sebagai dasar dari. AWS SDKs CRT menyediakan throughput yang ditingkatkan, manajemen koneksi yang ditingkatkan, dan waktu startup yang lebih cepat. CRT tersedia melalui semua AWS SDKs kecuali Go.

Untuk informasi selengkapnya tentang cara mengonfigurasi CRT untuk SDK yang Anda gunakan, lihat pustaka AWS Common Runtime (CRT), Mempercepatthroughput Amazon S3 dengan AWS Common Runtime, Memperkenalkan klien S3 berbasis CRT, dan Manajer Transfer S3 di SDK for Java 2.x, Menggunakan operasi S3 untuk AmazonS3, dan Mengkonfigurasi AWS klien HTTP berbasis CRT. CrtClient AWS

Gunakan versi terbaru dari AWS SDKs

Ini AWS SDKs menyediakan dukungan bawaan untuk banyak pedoman yang direkomendasikan untuk mengoptimalkan kinerja Amazon S3. SDKs Menawarkan API yang lebih sederhana untuk memanfaatkan Amazon S3 dari dalam aplikasi dan diperbarui secara berkala untuk mengikuti praktik terbaik terbaru. Misalnya, permintaan coba ulang SDKs secara otomatis setelah `503` kesalahan HTTP dan menangani respons koneksi lambat.

Jika menggunakan `Java 2.x` SDK, Anda harus mempertimbangkan untuk menggunakan S3 Transfer Manager, yang secara otomatis menskalakan koneksi secara horizontal untuk mencapai ribuan permintaan per detik menggunakan permintaan rentang byte jika diperlukan. Permintaan rentang byte dapat meningkatkan kinerja karena Anda dapat menggunakan koneksi bersamaan ke S3 untuk mengambil rentang byte yang berbeda dari dalam objek yang sama. Ini membantu Anda mencapai throughput agregat yang lebih tinggi dibandingkan permintaan objek utuh tunggal. Jadi penting untuk menggunakan versi terbaru dari AWS SDKs untuk mendapatkan fitur optimasi kinerja terbaru.

Penyelesaian masalah kinerja

Apakah Anda menyetel permintaan coba lagi untuk aplikasi yang sensitif terhadap latensi?

S3 Express One Zone dibuat khusus untuk menghadirkan tingkat kinerja tinggi yang konsisten tanpa penyetelan tambahan. Namun, menetapkan nilai batas waktu yang agresif dan percobaan ulang dapat membantu mendorong latensi dan kinerja yang konsisten. AWS SDKs Memiliki batas waktu yang dapat dikonfigurasi dan nilai coba lagi yang dapat Anda sesuaikan dengan toleransi aplikasi spesifik Anda.

Apakah Anda menggunakan pustaka AWS Common Runtime (CRT) dan tipe instans Amazon EC2 yang optimal?

Aplikasi yang melakukan sejumlah besar operasi baca dan tulis cenderung membutuhkan lebih banyak memori atau kapasitas komputasi daripada aplikasi yang tidak melakukannya. Saat meluncurkan instans Amazon Elastic Compute Cloud (Amazon EC2) untuk beban kerja yang menuntut kinerja, pilih jenis instans yang memiliki jumlah sumber daya ini yang dibutuhkan aplikasi Anda. Penyimpanan berkinerja tinggi S3 Express One Zone idealnya dipasangkan dengan tipe instans yang lebih besar dan lebih baru dengan jumlah memori sistem yang lebih besar dan lebih kuat CPUs dan GPUs dapat memanfaatkan penyimpanan berkinerja lebih tinggi. Kami juga merekomendasikan penggunaan versi terbaru dari CRT-enabled AWS SDKs, yang dapat mempercepat permintaan baca dan tulis secara paralel dengan lebih baik.

Apakah Anda menggunakan AWS SDKs otentikasi berbasis sesi?

Dengan Amazon S3, Anda juga dapat mengoptimalkan performa saat menggunakan permintaan HTTP REST API dengan mengikuti praktik terbaik yang sama yang merupakan bagian dari permintaan. AWS SDKs Namun, dengan mekanisme otorisasi dan otentikasi berbasis sesi yang digunakan oleh S3 Express One Zone, kami sangat menyarankan Anda menggunakan token AWS SDKs to manage `CreateSession` dan managed session token. AWS SDKs Secara otomatis membuat dan menyegarkan token atas nama Anda dengan menggunakan operasi `CreateSession`

API. Menggunakan `CreateSession` save on per-request round-trip latency to AWS Identity and Access Management (IAM) untuk mengotorisasi setiap permintaan.

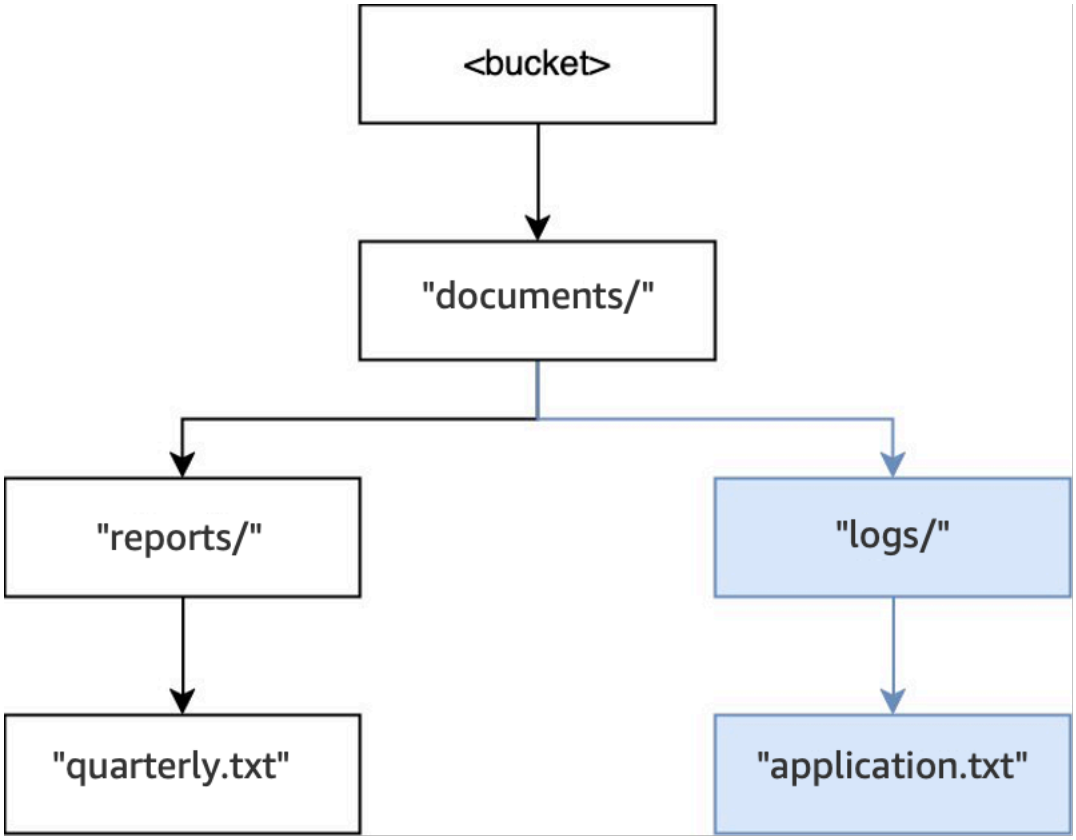
Operasi bucket direktori dan contoh interaksi direktori

Berikut ini menunjukkan tiga contoh tentang cara kerja bucket direktori.

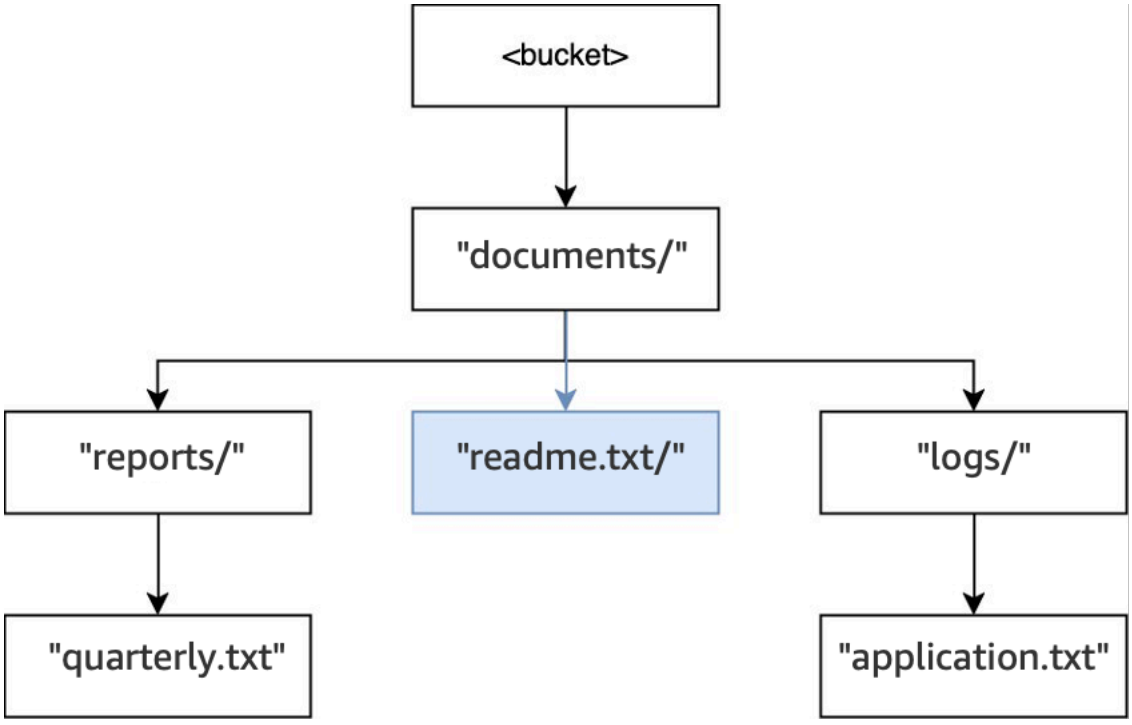
Contoh 1: Bagaimana **`PutObject`** permintaan S3 ke bucket direktori berinteraksi dengan direktori

1. Ketika operasi `PUT(<bucket>, "documents/reports/quarterly.txt")` dijalankan dalam bucket kosong, direktori di `documents/` dalam root bucket dibuat, direktori di `reports/` dalamnya `documents/` dibuat, dan objek di `quarterly.txt` dalamnya `reports/` dibuat. Untuk operasi ini, dua direktori dibuat selain objek.

2. Kemudian, ketika operasi lain `PUT(<bucket>, "documents/logs/application.txt")` dijalankan, direktori `documents/` sudah ada, direktori di `logs/` dalamnya `documents/` tidak ada dan dibuat, dan objek di `application.txt` dalamnya `logs/` dibuat. Untuk operasi ini, hanya satu direktori yang dibuat selain objek.



3. Terakhir, ketika `PUT(<bucket>, "documents/readme.txt")` operasi dijalankan, direktori di `documents/` dalam root sudah ada dan objek `readme.txt` dibuat. Untuk operasi ini, tidak ada direktori yang dibuat.



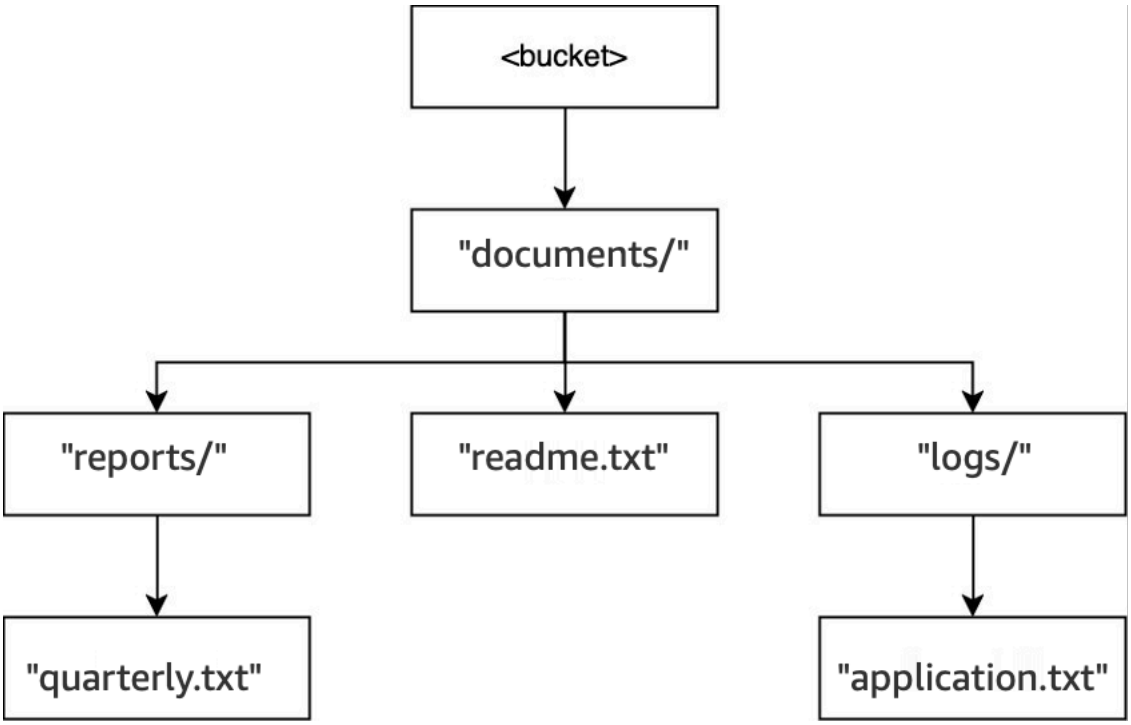
Contoh 2: Bagaimana **ListObjectsV2** permintaan S3 ke bucket direktori berinteraksi dengan direktori

Untuk `ListObjectsV2` permintaan S3 tanpa menentukan pembatas, bucket dilintasi secara mendalam terlebih dahulu. Output dikembalikan dalam urutan yang konsisten. Namun, meskipun urutan ini tetap sama di antara permintaan, urutannya tidak leksikografis. Untuk bucket dan direktori yang dibuat pada contoh sebelumnya:

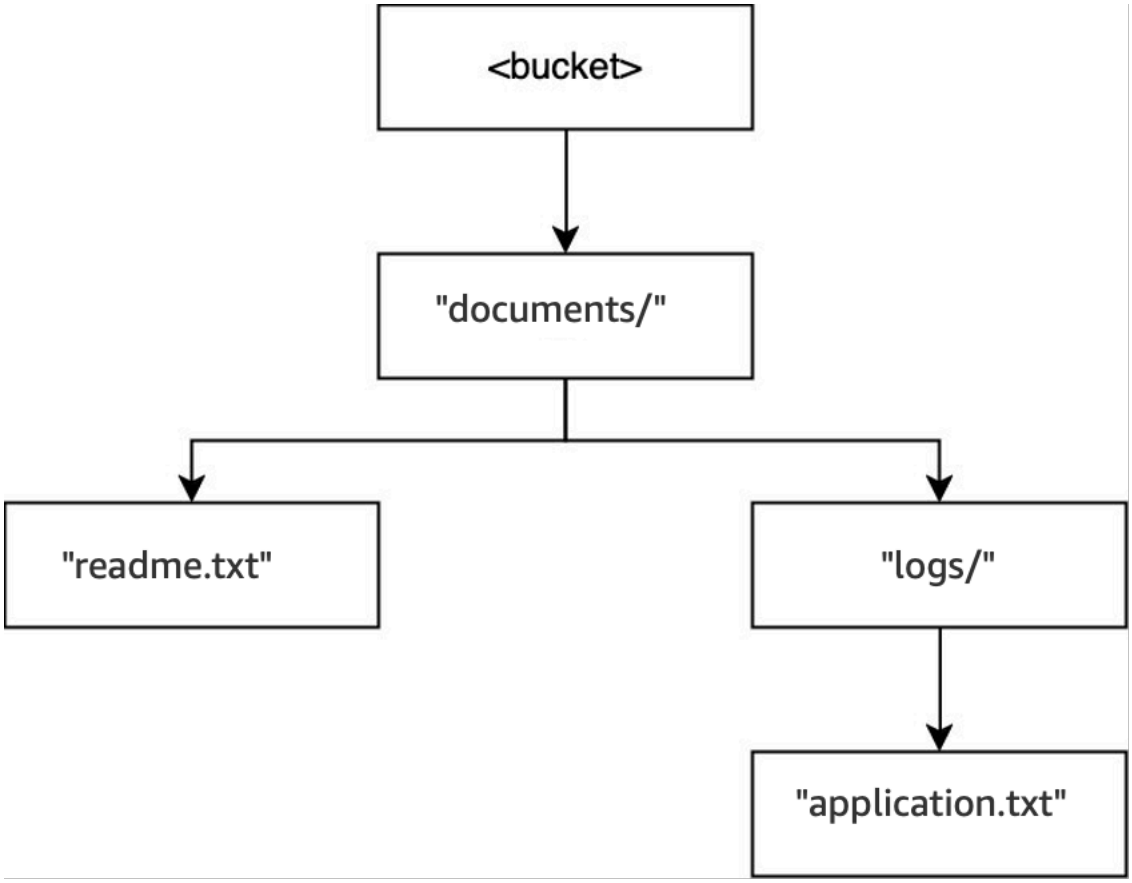
1. Ketika `LIST(<bucket>)` dieksekusi, direktori `documents/` dimasukkan dan melintasi dimulai.
2. Subdirektori `logs/` dimasukkan dan melintasi dimulai.
3. Objek `application.txt` ditemukan di dalam`logs/`.
4. Tidak ada lagi entri di dalamnya`logs/`. Operasi Daftar keluar dari `logs/` dan masuk `documents/` lagi.
5. `documents/`Direktori terus dilalui dan objek ditemukan`readme.txt`.
6. `documents/`Direktori terus dilalui dan subdirektori `reports/` dimasukkan dan melintasi dimulai.
7. Objek `quarterly.txt` ditemukan di dalam`reports/`.
8. Tidak ada lagi entri di dalamnya`reports/`. Daftar keluar dari `reports/` dan masuk `documents/` lagi.
9. Tidak ada lagi entri di dalam `documents/` dan Daftar kembali.

Dalam contoh ini, `logs/` dipesan sebelumnya `readme.txt` dan `readme.txt` dipesan sebelumnya`reports/`.

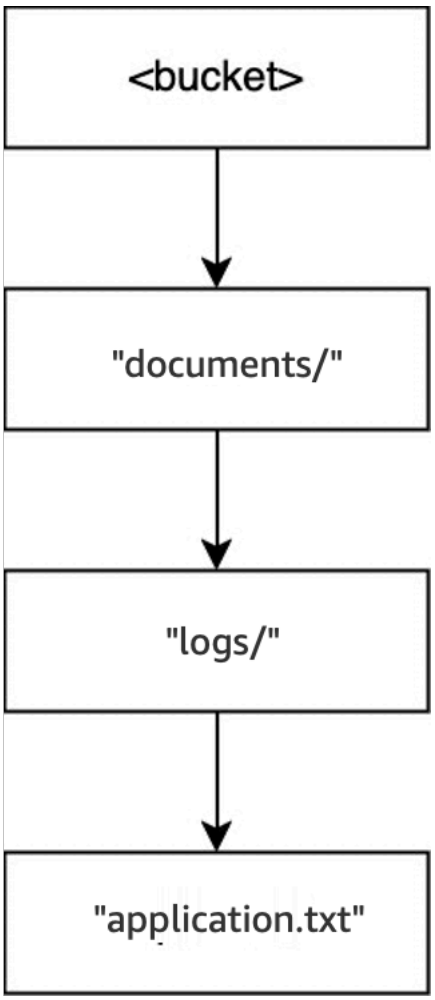
Contoh 3: Bagaimana **`DeleteObject`** permintaan S3 ke bucket direktori berinteraksi dengan direktori



1. Dalam ember yang sama, ketika operasi `DELETE(<bucket>, "documents/reports/quarterly.txt")` dijalankan, objek `quarterly.txt` dihapus, membiarkan direktori `reports/` kosong dan menyebabkannya segera dihapus. `documents/`Direktori tidak kosong karena memiliki direktori `logs/` dan objek di `readme.txt` dalamnya, jadi tidak dihapus. Untuk operasi ini, hanya satu objek dan satu direktori yang dihapus.



2. Ketika operasi `DELETE(<bucket>, "documents/readme.txt")` dijalankan, objek `readme.txt` dihapus. `documents/`masih belum kosong karena berisi direktori`logs/`, jadi tidak dihapus. Untuk operasi ini, tidak ada direktori yang dihapus dan hanya objek yang dihapus.



3. Terakhir, ketika operasi `DELETE(<bucket>, "documents/logs/application.txt")` dijalankan, dihapus, `application.txt` dibiarkan `logs/` kosong dan menyebabkannya segera dihapus. Ini kemudian dibiarkan `documents/` kosong dan menyebabkannya juga segera dihapus. Untuk operasi ini, dua direktori dan satu objek dihapus. Ember itu sekarang kosong.

Beban kerja residensi data

AWS Dedicated Local Zones (Dedicated Local Zones) adalah jenis AWS Infrastruktur yang dikelola sepenuhnya oleh AWS, dibangun untuk penggunaan eksklusif oleh Anda atau komunitas Anda, dan ditempatkan di lokasi atau pusat data yang ditentukan oleh Anda untuk membantu mematuhi

persyaratan peraturan. Dedicated Local Zones adalah jenis penawaran AWS Local Zones (Local Zones). Untuk informasi selengkapnya, lihat Zona Lokal AWS Khusus.

Di Zona Lokal Khusus, Anda dapat membuat bucket direktori S3 untuk menyimpan data dalam perimeter data tertentu, yang membantu mendukung kasus penggunaan residensi dan isolasi data. Bucket direktori di Dedicated Local Zones dapat mendukung kelas penyimpanan S3 Express One Zone dan S3 One Zone-Infrequent Access (S3 One Zone-IA; Z-IA). Bucket direktori saat ini tidak tersedia di lokasi AWS Local Zones lainnya.

Anda dapat menggunakan AWS Management Console, REST API, AWS Command Line Interface (AWS CLI), dan AWS SDKs di Dedicated Local Zones.

Untuk informasi selengkapnya tentang bekerja dengan bucket direktori di Local Zones, lihat topik berikut:

Topik

- Konsep untuk bucket direktori di Local Zones
- Aktifkan akun untuk Local Zones
- Konektivitas pribadi dari VPC Anda
- Membuat bucket direktori di Zona Lokal
- Mengautentikasi dan mengotorisasi bucket direktori di Local Zones

Konsep untuk bucket direktori di Local Zones

Sebelum membuat bucket direktori di Local Zone, Anda harus memiliki Local Zone ID tempat Anda ingin membuat bucket. Anda dapat menemukan semua informasi Zona Lokal dengan menggunakan operasi DescribeAvailabilityZonesAPI. Operasi API ini mencantumkan informasi tentang Local Zones, termasuk Zona Lokal IDs, nama Wilayah induk, grup perbatasan jaringan, dan status keikutsertaan. Setelah Anda memiliki ID Zona Lokal dan Anda ikut serta, Anda dapat membuat keranjang direktori di Zona Lokal. Nama bucket direktori terdiri dari nama dasar yang Anda berikan dan akhiran yang berisi ID Zona lokasi bucket Anda, diikuti oleh`--x-s3`.

Zona Lokal terhubung ke Wilayah induk menggunakan jaringan pribadi Amazon redundan dan bandwidth sangat tinggi. Ini memberi aplikasi yang berjalan di Zona Lokal akses cepat, aman, dan mulus ke bagian lain Layanan AWS di Wilayah induk. ID Zona Induk adalah ID zona yang menangani operasi bidang kontrol Zona Lokal. Network Border Group adalah grup unik yang AWS mengiklankan

alamat IP publik. Untuk informasi selengkapnya tentang Local Zones, Wilayah induk, dan ID Zona induk, lihat konsep AWS Local Zones di Panduan Pengguna AWS Local Zones.

Semua bucket direktori menggunakan `s3express` namespace, yang terpisah dari `s3` namespace untuk bucket tujuan umum. Untuk bucket direktori, permintaan dirutekan ke titik akhir Regional atau titik akhir Zonal. Routing ditangani secara otomatis untuk Anda jika Anda menggunakan AWS Management Console, AWS CLI, atau. AWS SDKs

Sebagian besar operasi API tingkat ember (seperti `CreateBucket` dan`DeleteBucket`) dirutekan ke titik akhir Regional, dan disebut sebagai operasi API titik akhir Regional. Titik akhir regional dalam format. `s3express-control.ParentRegionCode.amazonaws.com` Semua operasi API tingkat objek (sepertiPut`Object`) dan dua operasi API tingkat ember (dan`HeadBucket`) dirutekan ke titik akhir Zonal, `CreateSession` dan disebut sebagai operasi API titik akhir Zonal. Titik akhir zona dalam format. `s3express-LocalZoneID.ParentRegionCode.amazonaws.com` Untuk daftar lengkap operasi API menurut jenis titik akhir, lihat Operasi API bucket direktori.

S3 tersedia di Zona Lokal Beijing:

ID Zona Lokal: `cnn1-pkx1-az1`

- Nama Wilayah Induk: `China (Beijing)`
- Kode Wilayah Induk: `cn-north-1`
- Zona Induk IDs: `cnn1-az1`
- Nama Zona Lokal: `China (Beijing)`
- Kode Zona Lokal: `cnn1-pkx1-az1`
- Grup Perbatasan Jaringan: `cn-north-1-pkx-1`
- Contoh ember ARN: `arn:aws-cn:s3express:cn-north-1:`*`123456789012`*`:bucket/`*`amzn-s3-demo-bucket`*`--cnn1-pkx1-az1--x-s3`
- Titik akhir regional: `s3express-control.cn-north-1.amazonaws.com.cn`
- Titik akhir zona: `s3express-cnn1-pkx1-az1.cn-north-1.amazonaws.com.cn`
- Kelas penyimpanan: `S3 One Zone-Infrequent Access (S3 One Zone-IA; Z-IA)`

Untuk mengakses bucket direktori di Local Zones dari virtual private cloud (VPC), Anda dapat menggunakan titik akhir VPC gateway. Tidak dikenakan biaya tambahan untuk menggunakan titik akhir gateway. Untuk mengonfigurasi titik akhir VPC gateway untuk mengakses bucket direktori dan objek di Local Zones, lihat. Konektivitas pribadi dari VPC Anda

Aktifkan akun untuk Local Zones

Topik berikut menjelaskan bagaimana akun diaktifkan untuk Dedicated Local Zones.

Untuk semua layanan di Zona Lokal AWS Khusus (Zona Lokal Khusus), termasuk Amazon S3, administrator Anda harus mengaktifkan Akun AWS sebelum Anda dapat membuat atau mengakses sumber daya apa pun di Zona Lokal Khusus. Anda dapat menggunakan operasi DescribeAvailabilityZonesAPI untuk mengonfirmasi akses ID akun Anda ke Zona Lokal.

Untuk lebih melindungi data Anda di Amazon S3, secara default, Anda hanya memiliki akses ke sumber daya S3 yang Anda buat. Bucket di Local Zones memiliki semua pengaturan Akses Publik Blok S3 yang diaktifkan secara default dan Kepemilikan Objek S3 diatur ke pemilik bucket yang diberlakukan. Pengaturan ini tidak dapat dimodifikasi. Secara opsional, untuk membatasi akses hanya dalam grup perbatasan jaringan Zona Lokal, Anda dapat menggunakan kunci kondisi `s3express:AllAccessRestrictedToLocalZoneGroup` dalam kebijakan IAM Anda. Untuk informasi selengkapnya, lihat Mengautentikasi dan mengotorisasi bucket direktori di Local Zones.

Konektivitas pribadi dari VPC Anda

Anda dapat menggunakan titik akhir gateway untuk mengakses bucket direktori di AWS Local Zones (Local Zones) dari virtual private cloud (VPC) Anda, tanpa memerlukan gateway internet atau perangkat NAT untuk VPC Anda, dan tanpa biaya tambahan. Topik berikut menjelaskan konfigurasi titik akhir VPC gateway antara VPC dan bucket direktori di Local Zones.

Untuk mengonfigurasi titik akhir VPC gateway

1. Buka Konsol VPC Amazon.
2. Di panel navigasi, pilih Titik Akhir.
3. Pilih Buat Titik Akhir.
4. Buat nama untuk titik akhir Anda.
5. Untuk Kategori layanan, pilih Layanan AWS.
6. Untuk Layanan, tambahkan filter Type=Gateway dan kemudian pilih tombol opsi di sebelah com.amazonaws. ***region***.s3express.

7. Untuk VPC, pilih VPC yang akan digunakan untuk menciptakan titik akhir.
8. Untuk tabel Route, pilih tabel rute di Zona Lokal Anda yang akan digunakan oleh titik akhir. Setelah titik akhir dibuat, catatan rute akan ditambahkan ke tabel rute yang Anda pilih pada langkah ini.
9. Untuk Kebijakan, pilih Akses penuh untuk mengizinkan semua operasi oleh semua pengguna utama pada semua sumber daya melalui titik akhir VPC. Jika tidak, pilih Kustom untuk melampirkan kebijakan titik akhir VPC yang mengontrol izin prinsipal untuk melakukan tindakan pada sumber daya melalui titik akhir VPC.
10. (Opsional) Untuk menambahkan tanda, pilih Tambahkan tanda baru dan masukkan kunci dan nilai tanda.
11. Pilih Buat Titik Akhir.

Untuk mempelajari lebih lanjut tentang titik akhir VPC gateway, lihat Titik akhir Gateway di Panduan.AWS PrivateLink Untuk kasus penggunaan residensi data, sebaiknya aktifkan akses ke bucket Anda hanya dari VPC menggunakan titik akhir VPC gateway. Ketika akses dibatasi ke VPC atau titik akhir VPC, Anda dapat mengakses objek melalui, REST API AWS Management Console, dan. AWS CLI AWS SDKs

> **Note**
>
> Untuk membatasi akses ke VPC atau titik akhir VPC menggunakan, Anda harus menggunakan AWS Management Console Akses Pribadi. AWS Management Console Untuk informasi selengkapnya, lihat Akses AWS Management Console Pribadi di AWS Management Console panduan.

Membuat bucket direktori di Zona Lokal

Di Zona Lokal Khusus, Anda dapat membuat bucket direktori untuk menyimpan dan mengambil objek dalam perimeter data tertentu untuk membantu memenuhi kasus penggunaan residensi data dan isolasi data Anda. Bucket direktori S3 adalah satu-satunya jenis bucket yang didukung di Local Zones, dan berisi tipe lokasi bucket yang disebut. `LocalZone` Nama bucket direktori terdiri dari nama dasar yang Anda berikan dan akhiran yang berisi ID Zona lokasi bucket Anda dan`--x-s3`. Anda dapat memperoleh daftar Local Zone IDs dengan menggunakan operasi DescribeAvailabilityZonesAPI. Untuk informasi selengkapnya, lihat Aturan penamaan bucket direktori.

> **Note**
>
> - Untuk semua layanan di AWS Dedicated Local Zones (Dedicated Local Zones), termasuk S3, administrator Anda harus mengaktifkan Akun AWS sebelum Anda dapat membuat atau mengakses sumber daya apa pun di Zona Lokal Khusus. Untuk informasi selengkapnya, lihat Aktifkan akun untuk Local Zones.
> - Untuk persyaratan residensi data, sebaiknya aktifkan akses ke bucket Anda hanya dari titik akhir VPC gateway. Untuk informasi selengkapnya, lihat Konektivitas pribadi dari VPC Anda.
> - Untuk membatasi akses hanya dalam grup perbatasan jaringan Zona Lokal, Anda dapat menggunakan kunci kondisi `s3express:AllAccessRestrictedToLocalZoneGroup` dalam kebijakan IAM Anda. Untuk informasi selengkapnya, lihat Mengautentikasi dan mengotorisasi bucket direktori di Local Zones.

Berikut ini menjelaskan cara membuat bucket direktori dalam satu Zona Lokal dengan AWS Management Console, AWS CLI, dan AWS SDKs.

Menggunakan konsol S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi di bagian atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah induk dari Zona Lokal tempat Anda ingin membuat bucket direktori.

   > **Note**
   >
   > Untuk informasi selengkapnya tentang Wilayah induk, lihatKonsep untuk bucket direktori di Local Zones.

3. Di panel navigasi kiri, pilih Bucket.
4. Pilih Buat bucket.

   Halaman Buat bucket terbuka.
5. Di bawah Konfigurasi umum, lihat Wilayah AWS tempat bucket Anda akan dibuat.
6. Di bawah Jenis Bucket, pilih Direktori.

> **Note**
>
> - Jika Anda telah memilih Region yang tidak mendukung bucket direktori, jenis bucket default ke bucket tujuan umum. Untuk membuat bucket direktori, Anda harus memilih Region yang didukung. Untuk daftar Wilayah yang mendukung bucket direktori, lihatthe section called “Titik akhir regional dan Zonal untuk bucket direktori”.
> - Setelah membuat bucket, Anda tidak dapat mengubah tipe bucket.

7. Di bawah Lokasi Bucket, pilih Zona Lokal yang ingin Anda gunakan.

> **Note**
>
> Zona Lokal tidak dapat diubah setelah bucket dibuat.

8. Di bawah Lokasi Bucket, pilih kotak centang untuk mengetahui bahwa jika terjadi pemadaman Zona Lokal, data Anda mungkin tidak tersedia atau hilang.

> **Important**
>
> Meskipun bucket direktori disimpan di beberapa perangkat dalam satu Zona Lokal, bucket direktori tidak menyimpan data secara berlebihan di seluruh Local Zones.

9. Untuk Nama bucket, masukkan nama untuk bucket direktori Anda.

   Untuk informasi selengkapnya tentang aturan penamaan untuk bucket direktori, lihatAturan penamaan bucket tujuan umum. Sufiks secara otomatis ditambahkan ke nama dasar yang Anda berikan saat membuat bucket direktori menggunakan konsol. Akhiran ini mencakup ID Zona dari Zona Lokal yang Anda pilih.

   Setelah membuat bucket, Anda tidak dapat mengubah namanya.

> **Important**
>
> Jangan sertakan informasi sensitif, seperti nomor akun, dalam nama bucket. Nama ember terlihat di titik URLs itu ke objek di ember.

10. Di bawah Kepemilikan Objek, setelan yang diberlakukan pemilik Bucket diaktifkan secara otomatis, dan semua daftar kontrol akses (ACLs) dinonaktifkan. Untuk bucket direktori, ACLs dinonaktifkan dan tidak dapat diaktifkan.

    Dengan mengaktifkan pengaturan yang diberlakukan oleh pemilik Bucket, pemilik bucket secara otomatis memiliki dan memiliki kontrol penuh atas setiap objek di bucket. ACLs tidak lagi memengaruhi izin akses ke data di bucket S3. Bucket menggunakan kebijakan secara eksklusif untuk menentukan kontrol akses. Mayoritas kasus penggunaan modern di Amazon S3 tidak lagi memerlukan penggunaan. ACLs Untuk informasi selengkapnya, lihat Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.
11. Di bawah pengaturan Blokir Akses Publik untuk bucket ini, semua setelan Blokir Akses Publik untuk bucket direktori Anda diaktifkan secara otomatis. Pengaturan ini tidak dapat dimodifikasi untuk bucket direktori. Untuk informasi lebih lanjut tentang pemblokiran akses publik, lihat Melakukan blok akses publik ke penyimpanan Amazon S3 Anda.
12. Di bawah enkripsi Default, bucket direktori menggunakan enkripsi sisi Server dengan kunci terkelola Amazon S3 (SSE-S3) untuk mengenkripsi data secara default. Anda juga memiliki opsi untuk mengenkripsi data dalam bucket direktori dengan enkripsi sisi Server dengan kunci (SSE-KMS). AWS Key Management Service
13. Pilih Buat bucket.

    Setelah membuat bucket, Anda dapat menambahkan file dan folder ke bucket. Untuk informasi selengkapnya, lihat the section called “Bekerja dengan objek di dalam bucket direktori”.

Menggunakan AWS CLI

Contoh ini menunjukkan cara membuat bucket direktori di Local Zone dengan menggunakan AWS CLI. Untuk menggunakan perintah, ganti *user input placeholders* dengan informasi Anda sendiri.

Saat membuat bucket direktori, Anda harus memberikan detail konfigurasi dan menggunakan konvensi penamaan berikut:*bucket-base-name*--*zone-id*--x-s3.

```
aws s3api create-bucket
--bucket bucket-base-name--zone-id--x-s3
--create-bucket-configuration 'Location={Type=LocalZone,Name=local-zone-
id},Bucket={DataRedundancy=SingleLocalZone,Type=Directory}'
--region parent-region-code
```

Untuk informasi selengkapnya tentang ID Zona Lokal dan Kode Wilayah Induk, lihatKonsep untuk bucket direktori di Local Zones. Untuk informasi selengkapnya tentang AWS CLI perintah, lihat create-bucket di Command Reference.AWS CLI

Menggunakan AWS SDKs

SDK for Go

Contoh ini menunjukkan cara membuat bucket direktori di Local Zone dengan menggunakan AWS SDK untuk Go.

Example

```
var bucket = "bucket-base-name--zone-id--x-s3" // The full directory bucket name

func runCreateBucket(c *s3.Client) {
    resp, err := c.CreateBucket(context.Background(), &s3.CreateBucketInput{
        Bucket: &bucket,
        CreateBucketConfiguration: &types.CreateBucketConfiguration{
            Location: &types.LocationInfo{
                Name: aws.String("local-zone-id"),
                Type: types.LocationTypeLocalZone,
            },
            Bucket: &types.BucketInfo{
                DataRedundancy: types.DataRedundancySingleLocalZone,
                Type:           types.BucketTypeDirectory,
            },
        },
    })
    var terr *types.BucketAlreadyOwnedByYou
    if errors.As(err, &terr) {
        fmt.Printf("BucketAlreadyOwnedByYou: %s\n", aws.ToString(terr.Message))
        fmt.Printf("noop...\n") // No operation performed, just printing a message
        return
    }
    if err != nil {
        log.Fatal(err)
    }

    fmt.Printf("bucket created at %s\n", aws.ToString(resp.Location))
}
```

SDK for Java 2.x

Contoh ini menunjukkan cara membuat bucket direktori di Local Zone dengan menggunakan AWS SDK for Java 2.x.

Example

```
public static void createBucket(S3Client s3Client, String bucketName) {

   //Bucket name format is {base-bucket-name}--{local-zone-id}--x-s3
   //example: doc-example-bucket--local-zone-id--x-s3 is a valid name for a
 directory bucket created in a Local Zone.

   CreateBucketConfiguration bucketConfiguration =
 CreateBucketConfiguration.builder()
            .location(LocationInfo.builder()
                    .type(LocationType.LOCAL_ZONE)
                    .name("local-zone-id").build()) //this must match the Local
 Zone ID in your bucket name
            .bucket(BucketInfo.builder()
                   .type(BucketType.DIRECTORY)
                   .dataRedundancy(DataRedundancy.SINGLE_LOCAL_ZONE)
                   .build()).build();
   try {

            CreateBucketRequest bucketRequest =
 CreateBucketRequest.builder().bucket(bucketName).createBucketConfiguration(bucketConfigurat
            CreateBucketResponse response = s3Client.createBucket(bucketRequest);
            System.out.println(response);
   }

   catch (S3Exception e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
        }
   }
```

AWS SDK untuk JavaScript

Contoh ini menunjukkan cara membuat bucket direktori di Local Zone dengan menggunakan AWS SDK untuk JavaScript.

Example

```
// file.mjs, run with Node.js v16 or higher
// To use with the preview build, place this in a folder
// inside the preview build directory, such as /aws-sdk-js-v3/workspace/

import { S3 } from "@aws-sdk/client-s3";

const region = "parent-region-code";
const zone = "local-zone-id";
const suffix = `${zone}--x-s3`;

const s3 = new S3({ region });

const bucketName = `bucket-base-name--${suffix}`; // Full directory bucket name

const createResponse = await s3.createBucket(
    { Bucket: bucketName,
      CreateBucketConfiguration: {Location: {Type: "LocalZone", Name: "local-zone-
id"},
      Bucket: { Type: "Directory", DataRedundancy: "SingleLocalZone" }}
    }
   );
```

SDK untuk .NET

Contoh ini menunjukkan cara membuat bucket direktori di Local Zone dengan menggunakan SDK untuk .NET.

Example

```
using (var amazonS3Client = new AmazonS3Client())
{
    var putBucketResponse = await amazonS3Client.PutBucketAsync(new PutBucketRequest
    {

       BucketName = "bucket-base-name--local-zone-id--x-s3",
       PutBucketConfiguration = new PutBucketConfiguration
       {
         BucketInfo = new BucketInfo { DataRedundancy =
 DataRedundancy.SingleLocalZone, Type = BucketType.Directory },
         Location = new LocationInfo { Name = "local-zone-id", Type =
 LocationType.LocalZone }
```

```
        }
     }).ConfigureAwait(false);
}
```

SDK for PHP

Contoh ini menunjukkan cara membuat bucket direktori di Local Zone dengan menggunakan AWS SDK for PHP.

Example

```
require 'vendor/autoload.php';

$s3Client = new S3Client([

    'region'      => 'parent-region-code',
]);


$result = $s3Client->createBucket([
    'Bucket' => 'bucket-base-name--local-zone-id--x-s3',
    'CreateBucketConfiguration' => [
        'Location' => ['Name'=> 'local-zone-id', 'Type'=> 'LocalZone'],
        'Bucket' => ["DataRedundancy" => "SingleLocalZone" ,"Type" => "Directory"]
  ],
]);
```

SDK for Python

Contoh ini menunjukkan cara membuat bucket direktori di Local Zone dengan menggunakan AWS SDK untuk Python (Boto3).

Example

```
import logging
import boto3
from botocore.exceptions import ClientError

def create_bucket(s3_client, bucket_name, local_zone):
    '''
    Create a directory bucket in a specified Local Zone
```

```
    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket to create; for example, 'bucket-base-name--local-
zone-id--x-s3'
    :param local_zone: String; Local Zone ID to create the bucket in
    :return: True if bucket is created, else False
    '''

    try:
        bucket_config = {
                'Location': {
                    'Type': 'LocalZone',
                    'Name': local_zone
                },
                'Bucket': {
                    'Type': 'Directory',
                    'DataRedundancy': 'SingleLocalZone'
                }
            }
        s3_client.create_bucket(
            Bucket = bucket_name,
            CreateBucketConfiguration = bucket_config
        )
    except ClientError as e:
        logging.error(e)
        return False
    return True


if __name__ == '__main__':
    bucket_name = 'BUCKET_NAME'
    region = 'parent-region-code'
    local_zone = 'local-zone-id'
    s3_client = boto3.client('s3', region_name = region)
    create_bucket(s3_client, bucket_name, local_zone)
```

SDK for Ruby

Contoh ini menunjukkan cara membuat bucket direktori di Local Zone dengan menggunakan AWS SDK untuk Ruby.

Example

```
s3 = Aws::S3::Client.new(region:'parent-region-code')
```

```
s3.create_bucket(
  bucket: "bucket-base-name--local-zone-id--x-s3",
  create_bucket_configuration: {
    location: { name: 'local-zone-id', type: 'LocalZone' },
    bucket: { data_redundancy: 'SingleLocalZone', type: 'Directory' }
  }
)
```

Mengautentikasi dan mengotorisasi bucket direktori di Local Zones

Bucket direktori di Local Zones mendukung otorisasi AWS Identity and Access Management (IAM) dan otorisasi berbasis sesi. Untuk informasi selengkapnya tentang otentikasi dan otorisasi untuk bucket direktori, lihat. Mengautentikasi dan mengotorisasi permintaan

Sumber daya

Amazon Resource Names (ARNs) untuk bucket direktori berisi `s3express` namespace, Region AWS induk, Akun AWS ID, dan nama bucket direktori yang menyertakan ID Zona. Untuk mengakses dan melakukan tindakan pada bucket direktori Anda, Anda harus menggunakan format ARN berikut ini:

```
arn:aws:s3express:region-code:account-id:bucket/bucket-base-name--ZoneID--x-s3
```

Untuk bucket direktori di Zona Lokal, ID Zona adalah ID dari Zona Lokal. Untuk informasi selengkapnya tentang bucket direktori di Local Zones, lihatKonsep untuk bucket direktori di Local Zones. Untuk informasi selengkapnya ARNs, lihat Amazon Resource Names (ARNs) di Panduan Pengguna IAM. Untuk informasi selengkapnya tentang sumber daya, lihat IAM JSON Policy Elements: Resource di Panduan Pengguna IAM.

Kunci kondisi untuk bucket direktori di Local Zones

Di Local Zones, Anda dapat menggunakan semua kunci kondisi ini dalam kebijakan IAM Anda. Selain itu, untuk membuat perimeter data di sekitar grup perbatasan jaringan Zona Lokal, Anda dapat menggunakan tombol kondisi `s3express:AllAccessRestrictedToLocalZoneGroup` untuk menolak semua permintaan dari luar grup.

Kunci kondisi berikut dapat digunakan untuk lebih menyempurnakan kondisi di mana pernyataan kebijakan IAM berlaku. Untuk mengetahui daftar lengkap operasi API, tindakan kebijakan, dan kunci kondisi yang didukung oleh bucket direktori, lihat Tindakan kebijakan untuk bucket direktori.

> **Note**
>
> Kunci kondisi berikut hanya berlaku untuk Local Zones dan tidak didukung di Availability Zones dan Wilayah AWS.

Operasi API	Tindakan kebijakan	Deskripsi	Kunci syarat	Deskripsi	Tipe
Operasi API titik akhir zona	`s3expres: :CreateSe ssion`	Memberikan izin untuk membuat token sesi, yang digunakan untuk memberikan akses ke semua operasi API titik akhir Zonal, seperti,,, `CreateSes sion HeadBucke t` , `CopyObject` dan. `PutObject` `GetObject`	`s3expres: :AllAcce: sRestric edToLoca ZoneGrou`	Memfilter semua akses ke bucket kecuali permintaa n berasal dari grup perbatasan jaringan Zona AWS Lokal yang disediakan dalam kunci kondisi ini. Nilai: Nilai grup perbatasan jaringan Zona Lokal	`String`

Contoh kebijakan

Untuk membatasi akses objek ke permintaan dari dalam batas residensi data yang Anda tentukan (khususnya, Grup Zona Lokal yang merupakan kumpulan Local Zones yang diasuh ke yang sama Wilayah AWS), Anda dapat menetapkan salah satu kebijakan berikut:

- Kebijakan kontrol layanan (SCP). Untuk selengkapnya SCPs, lihat Kebijakan kontrol layanan (SCPs) di Panduan AWS Organizations Pengguna.
- Kebijakan berbasis identitas IAM untuk peran IAM.
- Kebijakan titik akhir VPC. Untuk informasi selengkapnya tentang kebijakan titik akhir VPC, lihat Mengontrol akses ke titik akhir VPC menggunakan kebijakan titik akhir di Panduan.AWS PrivateLink
- Kebijakan bucket S3.

> **Note**
>
> Kunci kondisi `s3express:AllAccessRestrictedToLocalZoneGroup` tidak mendukung akses dari lingkungan lokal. Untuk mendukung akses dari lingkungan lokal, Anda harus menambahkan IP sumber ke kebijakan. Untuk informasi selengkapnya, lihat aws: SourceIp di Panduan Pengguna IAM.

Example — Kebijakan SCP

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Access-to-specific-LocalZones-only",
            "Effect": "Deny",
            "Action": [
                "s3express:*",
            ],
            "Resource": "*",
            "Condition": {
                "StringNotEqualsIfExists": {
                    "s3express:AllAccessRestrictedToLocalZoneGroup": [
                        "local-zone-network-border-group-value"
                    ]
                }
            }
        }
    ]
}
```

Example — Kebijakan berbasis identitas IAM (melekat pada peran IAM)

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Deny",
        "Action": "s3express:CreateSession",
```

```
        "Resource": "*",
        "Condition": {
            "StringNotEqualsIfExists": {
                "s3express:AllAccessRestrictedToLocalZoneGroup": [
                    "local-zone-network-border-group-value"
                ]
            }
        }
    }
}
```

Example — Kebijakan titik akhir VPC

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Access-to-specific-LocalZones-only",
            "Principal": "*",
            "Action": "s3express:CreateSession",
            "Effect": "Deny",
            "Resource": "*",
            "Condition": {
                 "StringNotEqualsIfExists": {
                     "s3express:AllAccessRestrictedToLocalZoneGroup": [
                         "local-zone-network-border-group-value"
                     ]
                 }
            }
        }
    ]
}
```

Example — kebijakan ember

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Sid": "Access-to-specific-LocalZones-only",
            "Principal": "*",
            "Action": "s3express:CreateSession",
            "Effect": "Deny",
            "Resource": "*",
            "Condition": {
                 "StringNotEqualsIfExists": {
                     "s3express:AllAccessRestrictedToLocalZoneGroup": [
                         "local-zone-network-border-group-value"
                     ]
                 }
            }
        }
    ]
}
```

Perbedaan untuk ember direktori

Saat menggunakan Amazon S3, Anda dapat memilih jenis bucket yang paling sesuai dengan persyaratan aplikasi dan kinerja Anda. Ember direktori adalah jenis bucket yang paling baik digunakan untuk kasus penggunaan latensi rendah atau residensi data. Untuk mempelajari selengkapnya tentang bucket direktori, lihatBekerja dengan bucket direktori.

Untuk informasi selengkapnya tentang perbedaan bucket direktori, lihat topik berikut.

Topik

- Perbedaan untuk ember direktori
- Operasi API didukung untuk bucket direktori
- Fitur Amazon S3 tidak didukung oleh bucket direktori

Perbedaan untuk ember direktori

- Nama ember direktori
  - Nama bucket direktori terdiri dari nama dasar yang Anda berikan dan akhiran yang berisi ID AWS Zona (Availability Zone atau Local Zone) tempat bucket Anda berada, diikuti oleh`--x-s`. Untuk daftar aturan dan contoh nama bucket direktori, lihat Aturan penamaan bucket direktori.
- **`ListObjectsV2`**perilaku

  - Untuk bucket direktori, `ListObjectsV2` tidak mengembalikan objek dalam urutan leksikografis (abjad). Selain itu, prefiks harus diakhiri dengan pembatas dan hanya “/” yang dapat ditentukan sebagai pembatas.
  - Untuk bucket direktori, `ListObjectsV2` respons menyertakan awalan yang hanya terkait dengan unggahan multibagian yang sedang berlangsung.
- Perilaku penghapusan—Saat Anda menghapus objek di bucket direktori, Amazon S3 secara rekursif menghapus direktori kosong apa pun di jalur objek. Misalnya, jika Anda menghapus kunci objek`dir1/dir2/file1.txt`, Amazon S3 akan menghapus. `file1.txt` Jika direktori `dir1/` dan `dir2/` kosong dan tidak berisi objek lain, Amazon S3 juga menghapus direktori tersebut.
- ETags dan checksum — Entity tags (ETags) untuk bucket direktori adalah string alfanumerik acak yang unik untuk objek dan bukan checksum. MD5 Untuk informasi selengkapnya tentang penggunaan checksum tambahan dengan bucket direktori, lihat. Praktik terbaik checksum tambahan S3
- Kunci objek dalam permintaan **`DeleteObjects`**
  - Kunci objek dalam permintaan `DeleteObjects` harus berisi setidaknya satu karakter spasi non-putih. String dari semua karakter spasi putih tidak didukung dalam permintaan `DeleteObjects`.
  - Kunci objek dalam permintaan `DeleteObjects` tidak dapat berisi karakter kontrol Unicode, kecuali karakter baris baru (`\n`), tab (`\t`), dan kembali ke awal (`\r`).
- Titik akhir Regional dan Zonal — Operasi API manajemen ember untuk bucket direktori tersedia melalui titik akhir Regional dan disebut sebagai operasi API endpoint Regional. Contoh operasi API endpoint Regional adalah CreateBucket dan DeleteBucket. Setelah membuat bucket direktori, Anda dapat menggunakan operasi API titik akhir Zonal untuk mengunggah dan mengelola objek di bucket direktori. Operasi API titik akhir Zona tersedia melalui titik akhir Zona. Contoh operasi API titik akhir Zonal adalah `PutObject` dan `CopyObject`. Saat menggunakan bucket direktori, Anda harus menentukan Wilayah di semua permintaan. Untuk titik akhir Regional, Anda perlumenentukan Wilayah, misalnya `s3express-control.us-west-2.amazonaws.com`. Untuk titik akhir Zonal, Anda menentukan Wilayah dan Zona Ketersediaan, misalnya `s3express-usw2-az1.us-west-2.amazonaws.com`. Untuk informasi selengkapnya, lihat Titik akhir regional dan Zonal untuk bucket direktori.
- Unggahan multipart — Anda dapat mengunggah dan menyalin objek besar yang disimpan dalam bucket direktori dengan menggunakan proses upload multipart. Namun, berikut ini adalah beberapa perbedaan saat menggunakan proses upload multipart dengan objek yang disimpan dalam bucket direktori. Untuk informasi selengkapnya, lihat the section called “Menggunakan unggahan multibagian dengan bucket direktori”.

  - Tanggal pembuatan objek adalah tanggal penyelesaian unggahan multi-bagian.
  - Nomor bagian multi-bagian harus menggunakan nomor bagian berurutan. Jika Anda mencoba menyelesaikan permintaan unggahan multi-bagian dengan nomor bagian yang tidak berurutan, Amazon S3 akan menampilkan kesalahan HTTP `400 (Bad Request)`.
  - Pemrakarsa unggahan multi-bagian dapat membatalkan permintaan unggahan multi-bagian hanya jika mereka telah diberi izin eksplisit `AbortMultipartUpload` untuk mengakses melalui izin `s3express:CreateSession`. Untuk informasi selengkapnya, lihat Mengotorisasi operasi API endpoint Regional dengan IAM.
- Mengosongkan bucket direktori — `s3 rm` Perintah melalui (AWS Command Line Interface CLI), `delete` operasi melalui Mountpoint, dan tombol opsi Empty bucket melalui tombol tidak dapat menghapus unggahan multipart AWS Management Console yang sedang berlangsung dalam bucket direktori. Untuk menghapus unggahan multibagian yang sedang berlangsung ini, gunakan `ListMultipartUploads` operasi untuk mencantumkan unggahan multibagian yang sedang berlangsung di bucket dan gunakan `AbortMultipartUpload` operasi untuk membatalkan semua unggahan multibagian yang sedang berlangsung.
- AWS Local Zones — Local Zones hanya didukung untuk bucket direktori bukan bucket tujuan umum.
  - Menambahkan data ke objek yang ada tidak didukung untuk bucket direktori yang berada di Local Zones. Anda hanya dapat menambahkan data ke objek yang ada di bucket direktori yang berada di Availability Zones.
  - Siklus Hidup S3 tidak didukung untuk bucket direktori di Local Zones.

Operasi API didukung untuk bucket direktori

Bucket direktori mendukung operasi API titik akhir Regional (tingkat bucket, atau bidang kontrol) dan Zonal (level objek, atau bidang data). Untuk informasi selengkapnya, lihat Jaringan untuk ember direktori dan Titik akhir dan titik akhir VPC gateway. Untuk daftar operasi API yang didukung, lihatOperasi API bucket direktori.

Fitur Amazon S3 tidak didukung oleh bucket direktori

Fitur Amazon S3 berikut tidak didukung oleh bucket direktori:

- AWS kebijakan terkelola
- AWS PrivateLink untuk S3

- MD5 checksum
- Hapus autentikasi multi-faktor (MFA)
- Kunci Objek S3
- Pembayaran oleh Pemohon
- Pemberian Akses S3
- Metrik CloudWatch permintaan Amazon
- Notifikasi Peristiwa S3
- Tindakan transisi Siklus Hidup S3
- Titik Akses Multi-Wilayah S3
- Titik Akses Lambda Objek S3
- Penentuan Versi S3
- Inventaris S3
- Replikasi S3
- Tanda objek
- Pilih S3
- Log akses server
- Hosting situs web statis
- Lensa Penyimpanan S3
- Grup Lensa Penyimpanan S3
- Akselerasi Transfer S3
- Enkripsi sisi server dua lapis dengan kunci () (DSSE-KMS AWS Key Management Service)AWS KMS
- Enkripsi di sisi server dengan kunci yang disediakan pelanggan (SSE-C)
- Opsi untuk menyalin pengaturan bucket yang ada saat membuat bucket baru di konsol Amazon S3
- Pesan kesalahan akses yang disempurnakan ditolak (HTTP`403 Forbidden`)

Jaringan untuk ember direktori

Untuk mengakses bucket direktori dan objek di dalamnya, Anda menggunakan titik akhir API Regional dan Zonal yang berbeda dari titik akhir Amazon S3 standar. Bergantung pada operasi API

S3 yang Anda gunakan, titik akhir Zonal atau Regional diperlukan. Untuk daftar lengkap operasi API menurut jenis titik akhir, lihat Perbedaan untuk ember direktori.

Anda dapat mengakses operasi API Zonal dan Regional melalui titik akhir cloud privat virtual (VPC) gateway.

Topik berikut menjelaskan persyaratan jaringan untuk mengakses S3 Express One Zone dengan menggunakan titik akhir VPC gateway.

Topik

- Titik akhir
- Mengkonfigurasi titik akhir gateway VPC

Titik akhir

Anda dapat mengakses bucket direktori dan objek di dalam dari VPC Anda dengan menggunakan titik akhir VPC gateway. Bucket direktori menggunakan titik akhir API Regional dan Zonal. Bergantung pada operasi Amazon S3 API yang Anda gunakan, titik akhir Regional atau Zonal diperlukan. Tidak dikenakan biaya tambahan untuk menggunakan titik akhir gateway.

Operasi API tingkat bucket (atau bidang kontrol) tersedia melalui titik akhir Regional dan disebut sebagai operasi API titik akhir Regional. Contoh operasi API titik akhir Regional adalah `CreateBucket` dan `DeleteBucket`. Saat membuat bucket direktori, Anda memilih satu Zona (Availability Zone atau Local Zone) tempat bucket direktori Anda akan dibuat. Setelah membuat bucket direktori, Anda dapat menggunakan operasi API titik akhir Zonal untuk mengunggah dan mengelola objek di bucket direktori.

Operasi API tingkat objek (atau bidang data) tersedia melalui titik akhir Zonal dan disebut sebagai operasi API titik akhir Zonal. Contoh operasi API titik akhir Zonal adalah `CreateSession` dan `PutObject`.

Untuk informasi selengkapnya tentang titik akhir dan lokasi yang mendukung bucket direktori di Availability Zones, lihat. Titik akhir untuk bucket direktori di Availability Zones

Untuk informasi selengkapnya tentang titik akhir dan lokasi yang mendukung bucket direktori di Local Zones, lihat. Aktifkan akun untuk Local Zones

Mengkonfigurasi titik akhir gateway VPC

Untuk mengonfigurasi titik akhir VPC gateway untuk bucket direktori akses di Availability Zones, lihat. the section called “Mengkonfigurasi titik akhir gateway VPC”

Untuk mengonfigurasi titik akhir VPC gateway untuk bucket direktori akses di Local Zones, lihat. Konektivitas pribadi dari VPC Anda

Aturan penamaan bucket direktori

Saat membuat bucket direktori di Amazon S3, aturan penamaan bucket berikut berlaku. Untuk aturan penamaan bucket bertujuan umum, lihat Aturan penamaan bucket tujuan umum.

Nama bucket direktori terdiri dari nama dasar yang Anda berikan, dan akhiran yang berisi ID AWS Zona (Availability Zone atau Local Zone) tempat bucket Anda berada dan`--x-s3`. *`zone-id`*Bisa berupa ID Availability Zone atau Local Zone.

```
base-name--zoneid--x-s3
```

Misalnya, nama bucket direktori berikut berisi ID Zona Ketersediaan `usw2-az1`:

```
bucket-base-name--usw2-az1--x-s3
```

> **Note**
>
> Saat Anda membuat bucket direktori menggunakan konsol, akhiran secara otomatis ditambahkan ke nama dasar yang Anda berikan. Akhiran ini mencakup ID Zona Zona (Availability Zone atau Local Zone) yang Anda pilih.
> Saat membuat bucket direktori menggunakan API, Anda harus memberikan akhiran lengkap, termasuk ID Zone, dalam permintaan Anda. Untuk daftar Zone IDs, lihatTitik akhir.

Aturan penamaan berikut berlaku untuk bucket direktori.

- Jadilah unik dalam Zona yang dipilih (AWS Availability Zone atau AWS Local Zone).
- Panjang nama harus antara 3 (menit) dan 63 (maks) karakter, termasuk akhiran.
- Hanya terdiri dari huruf kecil, angka, tanda hubung (-).
- Dimulai dan diakhiri dengan huruf atau angka.

- Harus menyertakan akhiran berikut:`--`*`zone-id`*`--x-s3`.
- Nama bucket tidak boleh dimulai dengan prefiks `xn--`.
- Nama bucket tidak boleh dimulai dengan prefiks `sthree-`.
- Nama bucket tidak boleh dimulai dengan prefiks `sthree-configurator`.
- Nama bucket tidak boleh dimulai dengan prefiks `amzn-s3-demo-`.
- Nama bucket tidak boleh diakhiri dengan sufiks `-s3alias`. Sufiks ini dicadangkan untuk nama alias titik akses. Untuk informasi selengkapnya, lihat Alias titik akses.
- Nama bucket tidak boleh diakhiri dengan sufiks `--ol-s3`. Sufiks ini dicadangkan untuk nama alias Titik Akses Lambda Objek. Untuk informasi selengkapnya, lihat Cara menggunakan alias gaya bucket untuk bucket S3 Anda Titik Akses Lambda Objek.
- Nama bucket tidak boleh diakhiri dengan sufiks `.mrap`. Akhiran ini dicadangkan untuk nama Titik Akses Multi-Wilayah. Untuk informasi selengkapnya, lihat Aturan penamaan Titik Akses Multi-Wilayah Amazon S3.

Melihat properti bucket direktori

Anda dapat melihat dan mengonfigurasi properti untuk bucket direktori Amazon S3 dengan menggunakan konsol Amazon S3. Untuk informasi selengkapnya, lihat Bekerja dengan bucket direktori.

Menggunakan konsol S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket direktori.
3. Di daftar Bucket direktori, pilih nama bucket yang ingin Anda lihat propertinya.
4. Pilih tab Properti.
5. Pada tab Properties, Anda dapat melihat properti berikut untuk bucket:

    - Ikhtisar bucket direktori — Anda dapat melihat Wilayah AWS, Zona (Availability Zone atau Local Zone), Amazon Resource Name (ARN), dan tanggal pembuatan bucket.
    - Pengaturan enkripsi sisi server — Amazon S3 menerapkan enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3) sebagai tingkat dasar enkripsi untuk semua bucket S3. Amazon S3 mengenkripsi sebuah objek sebelum menyimpannya ke disk, dan mendekripsi

objek tersebut saat Anda mengunduhnya. Untuk informasi selengkapnya, lihat Pengaturan dan pemantauan enkripsi default untuk bucket direktori.

Untuk informasi selengkapnya tentang fitur untuk bucket direktori, lihat Membuat dan menggunakan bucket direktori.

Mengelola kebijakan bucket direktori

Anda dapat menambahkan, menghapus, memperbarui, dan melihat kebijakan bucket untuk bucket direktori Amazon S3 dengan menggunakan konsol Amazon S3, dan CLI. AWS SDKs AWS Untuk informasi selengkapnya, lihat topik berikut. Untuk informasi selengkapnya tentang tindakan yang didukung AWS Identity and Access Management (IAM), lihatMengotorisasi operasi API endpoint Regional dengan IAM. Untuk contoh kebijakan bucket untuk bucket direktori, lihat Contoh kebijakan bucket untuk bucket direktori.

Topik

- Menambahkan kebijakan bucket
- Melihat kebijakan bucket
- Menghapus kebijakan bucket

Menambahkan kebijakan bucket

Untuk menambahkan kebijakan bucket ke bucket direktori, Anda dapat menggunakan konsol Amazon S3, the AWS SDKs, atau. AWS CLI

Menggunakan konsol S3

Untuk membuat atau mengedit kebijakan bucket

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket direktori.
3. Di daftar bucket Direktori, pilih nama bucket yang ingin Anda tambahkan kebijakan.
4. Pilih tab Izin.
5. Di Bawah Kebijakan bucket, pilih Edit. Halaman Edit kebijakan bucket akan muncul.
6. Untuk membuat kebijakan secara otomatis, pilih Generator kebijakan.

Jika Anda memilih Policy Generator, AWS Policy Generator akan terbuka di jendela baru.

Jika Anda tidak ingin menggunakan Generator AWS Kebijakan, Anda dapat menambahkan atau mengedit pernyataan JSON di bagian Kebijakan.

a. Pada halaman AWS Pembuat Kebijakan, untuk Pilih Jenis Kebijakan, pilih Kebijakan Bucket S3.

b. Tambahkan pernyataan dengan memasukkan informasi di bidang yang disediakan, lalu pilih Tambah Pernyataan. Ulangi langkah ini untuk pernyataan sebanyak yang ingin Anda tambahkan. Untuk informasi selengkapnya tentang persyaratan kebijakan, lihat Referensi kebijakan IAM JSON di Panduan Pengguna IAM.

> **Note**
>
> Demi kenyamanan Anda, halaman kebijakan Edit bucket menampilkan Bucket ARN (Nama Sumber Daya Amazon) dari bucket saat ini di atas bidang teks Kebijakan. Anda dapat menyalin ARN ini untuk digunakan dalam pernyataan di halaman Pembuat Kebijakan AWS .

c. Setelah Anda selesai menambahkan pernyataan, pilih Buat Kebijakan.

d. Salin teks kebijakan yang dihasilkan, pilih Tutup, dan kembali ke halaman Edit kebijakan bucket di konsol Amazon S3.

7. Di kotak Kebijakan, edit kebijakan yang ada atau tempel kebijakan bucket dari AWS Policy Generator. Pastikan peringatan keamanan, kesalahan, peringatan umum, dan saran telah ditangani sebelum menyimpan kebijakan.

> **Note**
>
> Kebijakan bucket dibatasi hingga ukuran 20 KB.

8. Pilih Simpan perubahan, yang mengembalikan Anda ke tab Izin.

Menggunakan AWS SDKs

SDK for Java 2.x

Example

PutBucketPolicy AWS SDK for Java 2.x

```
public static void setBucketPolicy(S3Client s3Client, String bucketName, String
 policyText) {

      //sample policy text
      /**
       * policy_statement = {
        *         'Version': '2012-10-17',
        *         'Statement': [
        *             {
        *                 'Sid': 'AdminPolicy',
        *                 'Effect': 'Allow',
        *                 'Principal': {
        *                     "AWS": "111122223333"
        *                 },
        *                 'Action': 's3express:*',
        *                 'Resource':
 'arn:aws:s3express:region:111122223333:bucket/bucket-base-name--zone-id--x-s3'
        *             }
        *         ]
        *     }
        */
        System.out.println("Setting policy:");
        System.out.println("----");
        System.out.println(policyText);
        System.out.println("----");
        System.out.format("On Amazon S3 bucket: \"%s\"\n", bucketName);

        try {
            PutBucketPolicyRequest policyReq = PutBucketPolicyRequest.builder()
                    .bucket(bucketName)
                    .policy(policyText)
                    .build();
            s3Client.putBucketPolicy(policyReq);
            System.out.println("Done!");
        }
```

```
            catch (S3Exception e) {
                System.err.println(e.awsErrorDetails().errorMessage());
                System.exit(1);
            }
        }
```

Menggunakan AWS CLI

Contoh ini menunjukkan cara menambahkan kebijakan bucket ke bucket direktori dengan menggunakan AWS CLI. Untuk menggunakan perintah ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3api put-bucket-policy --bucket bucket-base-name--zone-id--x-s3 --policy file://
bucket_policy.json
```

bucket_policy.json:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AdminPolicy",
            "Effect": "Allow",
            "Principal": {
                "AWS": "111122223333"
            },
            "Action": "s3express*",
            "Resource": "arn:aws:s3express:us-west-2:111122223333:bucket/amzn-s3-demo-
bucket--usw2-az1--x-s3"
        }
    ]
}
```

Untuk informasi selengkapnya, lihat put-bucket-policy di AWS Command Line Interface.

Melihat kebijakan bucket

Untuk melihat kebijakan bucket untuk bucket direktori, gunakan contoh berikut.

Menggunakan AWS CLI

Contoh ini menunjukkan cara melihat kebijakan bucket yang dilampirkan ke bucket direktori dengan menggunakan AWS CLI. Untuk menggunakan perintah ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3api get-bucket-policy --bucket bucket-base-name--zone-id--x-s3
```

Untuk informasi selengkapnya, lihat get-bucket-policy di AWS Command Line Interface.

Menghapus kebijakan bucket

Untuk menghapus kebijakan bucket untuk bucket direktori, gunakan contoh berikut.

Menggunakan AWS SDKs

SDK for Java 2.x

Example

`DeleteBucketPolicy` AWS SDK for Java 2.x

```
public static void deleteBucketPolicy(S3Client s3Client, String bucketName) {
      try {
          DeleteBucketPolicyRequest deleteBucketPolicyRequest =
 DeleteBucketPolicyRequest
                  .builder()
                  .bucket(bucketName)
                  .build()
          s3Client.deleteBucketPolicy(deleteBucketPolicyRequest);
          System.out.println("Successfully deleted bucket policy");
      }

      catch (S3Exception e) {
          System.err.println(e.awsErrorDetails().errorMessage());
          System.exit(1);
      }
```

Menggunakan AWS CLI

Contoh ini menunjukkan cara menghapus kebijakan bucket untuk bucket direktori dengan menggunakan AWS CLI. Untuk menggunakan perintah ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3api delete-bucket-policy --bucket bucket-base-name--zone-id--x-s3
```

Untuk informasi selengkapnya, lihat delete-bucket-policy di AWS Command Line Interface.

Mengosongkan bucket direktori

Anda dapat mengosongkan bucket direktori Amazon S3 dengan menggunakan konsol Amazon S3. Untuk informasi selengkapnya tentang bucket direktori, lihat Bekerja dengan bucket direktori.

Sebelum Anda mengosongkan bucket direktori, perhatikan hal berikut ini:

- Saat mengosongkan bucket direktori, Anda menghapus semua objek tetapi menyimpan bucket direktori.
- Setelah Anda mengosongkan bucket direktori, tindakan kosong tidak dapat dibatalkan.
- Objek yang ditambahkan ke bucket direktori saat aksi bucket kosong sedang berlangsung mungkin akan dihapus.

Jika Anda juga ingin menghapus ember, perhatikan hal berikut:

- Semua objek di bucket direktori harus dihapus sebelum bucket itu sendiri dapat dihapus.
- Unggahan multibagian yang sedang berlangsung di bucket direktori harus dibatalkan sebelum bucket itu sendiri dapat dihapus.

> **Note**
>
> `s3 rm`Perintah melalui AWS Command Line Interface (CLI), `delete` operasi melalui Mountpoint, dan tombol opsi Empty bucket melalui tombol tidak dapat menghapus unggahan multibagian AWS Management Console yang sedang berlangsung dalam ember direktori. Untuk menghapus unggahan multibagian yang sedang berlangsung ini, gunakan `ListMultipartUploads` operasi untuk mencantumkan unggahan multibagian yang

sedang berlangsung di bucket dan gunakan `AbortMultipartUpload` operasi untuk membatalkan semua unggahan multibagian yang sedang berlangsung.

Untuk mengosongkan bucket direktori, lihat Menghapus bucket direktori. Untuk membatalkan unggahan multipart yang sedang berlangsung, lihat. the section called “Membatalkan unggahan multibagian”

Untuk menghapus bucket bertujuan umum, lihat Mengosongkan ember tujuan umum.

Menggunakan konsol S3

Untuk mengosongkan ember direktori

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket direktori.
3. Pilih tombol opsi di sebelah nama ember yang ingin Anda kosongkan, lalu pilih Kosong.
4. Di halaman Bucket kosong, konfirmasikan bahwa Anda ingin mengosongkan bucket dengan mengetik **`permanently delete`** ke dalam bidang teks, lalu pilih Kosongkan.
5. Pantau perkembangan proses pengosongan bucket di halaman Empty bucket: status.

Menghapus bucket direktori

Anda hanya dapat menghapus ember direktori Amazon S3 kosong. Sebelum menghapus bucket direktori, Anda harus menghapus semua objek di bucket dan membatalkan semua unggahan multipart yang sedang berlangsung.

Jika bucket direktori dilampirkan ke titik akses, Anda harus menghapus titik akses terlebih dahulu. Untuk informasi selengkapnya, lihat Hapus titik akses Anda untuk bucket direktori.

Untuk mengosongkan bucket direktori, lihat Mengosongkan bucket direktori. Untuk membatalkan unggahan multipart yang sedang berlangsung, lihat. the section called “Membatalkan unggahan multibagian”

Untuk menghapus bucket bertujuan umum, lihat Menghapus bucket tujuan umum.

Menggunakan konsol S3

Setelah mengosongkan bucket direktori dan membatalkan semua unggahan multipart yang sedang berlangsung, Anda dapat menghapus bucket.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket direktori.
3. Dalam daftar bucket Direktori, pilih tombol opsi di sebelah bucket yang ingin Anda hapus.
4. Pilih Hapus.
5. Pada halaman Delete bucket, masukkan nama bucket di kolom teks untuk mengonfirmasi penghapusan bucket Anda.

   > **Important**
   >
   > Menghapus bucket direktori tidak dapat dibatalkan.

6. Untuk menghapus bucket direktori, pilih Hapus bucket.

Menggunakan AWS SDKs

Contoh berikut menghapus bucket direktori dengan menggunakan AWS SDK for Java 2.x dan AWS SDK untuk Python (Boto3).

SDK for Java 2.x

Example

```
public static void deleteBucket(S3Client s3Client, String bucketName) {

    try {
        DeleteBucketRequest del = DeleteBucketRequest.builder()
                .bucket(bucketName)
                .build();
        s3Client.deleteBucket(del);
        System.out.println("Bucket " + bucketName + " has been deleted");
    }
    catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
```

```
        System.exit(1);
    }
}
```

SDK for Python

Example

```
import logging
import boto3
from botocore.exceptions import ClientError

def delete_bucket(s3_client, bucket_name):
    '''
    Delete a directory bucket in a specified Region

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket to delete; for example, 'doc-example-bucket--usw2-
az1--x-s3'
    :return: True if bucket is deleted, else False
    '''

    try:
        s3_client.delete_bucket(Bucket = bucket_name)
    except ClientError as e:
        logging.error(e)
        return False
    return True

if __name__ == '__main__':
    bucket_name = 'BUCKET_NAME'
    region = 'us-west-2'
    s3_client = boto3.client('s3', region_name = region)
```

Menggunakan AWS CLI

Contoh ini menunjukkan cara menghapus bucket direktori dengan menggunakan file AWS CLI. Untuk menggunakan perintah ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3api delete-bucket --bucket bucket-base-name--zone-id--x-s3 --region us-west-2
```

Untuk informasi selengkapnya, lihat delete-bucket di. AWS Command Line Interface

Mendaftar bucket direktori

Contoh berikut menunjukkan cara membuat daftar bucket direktori dengan menggunakan AWS Management Console, AWS SDKs, dan AWS CLI.

Menggunakan konsol S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi di bagian atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah di mana Anda ingin melihat daftar bucket direktori Anda.
3. Di panel navigasi kiri, pilih Bucket direktori. Daftar ember direktori muncul. Untuk melihat objek di bucket, properti bucket, izin bucket, metrik, titik akses yang terkait dengan bucket, atau untuk mengelola bucket, pilih nama bucket.

Menggunakan AWS SDKs

SDK for Java 2.x

Example

Contoh berikut mencantumkan bucket direktori dengan menggunakan file. AWS SDK for Java 2.x

```
public static void listBuckets(S3Client s3Client) {
       try {
           ListDirectoryBucketsRequest listDirectoryBucketsRequest =
ListDirectoryBucketsRequest.builder().build();
           ListDirectoryBucketsResponse response =
s3Client.listDirectoryBuckets(listDirectoryBucketsRequest);
           if (response.hasBuckets()) {
               for (Bucket bucket: response.buckets()) {
                   System.out.println(bucket.name());
                   System.out.println(bucket.creationDate());
               }
           }
       }

       catch (S3Exception e) {
           System.err.println(e.awsErrorDetails().errorMessage());
```

```
            System.exit(1);
        }
    }
```

SDK for Python

Example

Contoh berikut mencantumkan bucket direktori dengan menggunakan file. AWS SDK untuk Python (Boto3)

```
import logging
import boto3
from botocore.exceptions import ClientError

def list_directory_buckets(s3_client):
'''
Prints a list of all directory buckets in a Region

:param s3_client: boto3 S3 client
:return: True if there are buckets in the Region, else False
'''
try:
    response = s3_client.list_directory_buckets()
    for bucket in response['Buckets']:
        print (bucket['Name'])
except ClientError as e:
    logging.error(e)
    return False
return True


if __name__ == '__main__':
    region = 'us-east-1'
    s3_client = boto3.client('s3', region_name = region)
    list_directory_buckets(s3_client)
```

SDK untuk .NET

Example

Contoh berikut mencantumkan bucket direktori dengan menggunakan file. AWS SDK untuk .NET

```
var listDirectoryBuckets = await amazonS3Client.ListDirectoryBucketsAsync(new
 ListDirectoryBucketsRequest
{
  MaxDirectoryBuckets = 10
  }).ConfigureAwait(false);
```

SDK for PHP

Example

Contoh berikut mencantumkan bucket direktori dengan menggunakan file. AWS SDK untuk PHP

```
require 'vendor/autoload.php';

$s3Client = new S3Client([
    'region'      => 'us-east-1',
]);
$result = $s3Client->listDirectoryBuckets();
```

SDK for Ruby

Example

Contoh berikut mencantumkan bucket direktori dengan menggunakan file. AWS SDK untuk Ruby

```
s3 = Aws::S3::Client.new(region:'us-west-1')
s3.list_directory_buckets
```

Menggunakan AWS CLI

`list-directory-buckets`Contoh perintah berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk daftar bucket direktori Anda di *`us-east-1`* wilayah tersebut. Untuk menjalankan perintah ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3api list-directory-buckets --region us-east-1
```

Untuk informasi selengkapnya, lihat list-directory-buckets dalam AWS CLI Referensi Perintah.

Menentukan apakah Anda dapat mengakses bucket direktori

Contoh AWS SDK berikut menunjukkan cara menggunakan operasi `HeadBucket` API untuk menentukan apakah bucket direktori Amazon S3 ada dan apakah Anda memiliki izin untuk mengaksesnya.

Menggunakan AWS SDKs

AWS SDK for Java 2.x Contoh berikut menunjukkan cara menentukan apakah ada bucket dan apakah Anda memiliki izin untuk mengaksesnya.

SDK for Java 2.x

Example

AWS SDK for Java 2.x

```
public static void headBucket(S3Client s3Client, String bucketName) {
   try {
        HeadBucketRequest headBucketRequest = HeadBucketRequest
                .builder()
                .bucket(bucketName)
                .build();
        s3Client.headBucket(headBucketRequest);
        System.out.format("Amazon S3 bucket: \"%s\" found.", bucketName);
   }

   catch (S3Exception e) {
       System.err.println(e.awsErrorDetails().errorMessage());
       System.exit(1);
   }
}
```

Menggunakan AWS CLI

Perintah `head-bucket` contoh berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk menentukan apakah ada bucket direktori dan apakah Anda memiliki izin untuk mengaksesnya. Untuk menjalankan perintah ini, ganti placeholder input pengguna dengan informasi Anda sendiri.

```
aws s3api head-bucket --bucket bucket-base-name--zone-id--x-s3
```

Untuk informasi selengkapnya, lihat head-bucket dalam AWS CLI Referensi Perintah.

Bekerja dengan objek di dalam bucket direktori

Setelah membuat bucket direktori Amazon S3, Anda dapat bekerja dengan objek menggunakan konsol Amazon S3 AWS Command Line Interface ,AWS CLI(), dan file. AWS SDKs

Untuk informasi selengkapnya tentang melakukan operasi massal, mengimpor, mengunggah, menyalin, menghapus, dan mengunduh objek dalam bucket direktori, lihat topik berikut.

Topik

- Mengimpor objek ke dalam bucket direktori
- Bekerja dengan Siklus Hidup S3 untuk bucket direktori
- Menggunakan Operasi Batch dengan bucket direktori
- Menambahkan data ke objek dalam ember direktori
- Mengganti nama objek dalam ember direktori
- Mengunggah objek ke ember direktori
- Menyalin objek dari atau ke ember direktori
- Menghapus objek dari ember direktori
- Mengunduh objek dari ember direktori
- Menghasilkan keranjang direktori objek URLs yang telah ditentukan sebelumnya untuk berbagi objek
- Mengambil metadata objek dari bucket direktori
- Daftar objek dari ember direktori

Mengimpor objek ke dalam bucket direktori

Setelah membuat bucket direktori di Amazon S3, Anda dapat mengisi bucket baru dengan data menggunakan tindakan impor. Impor adalah metode yang disederhanakan untuk membuat tugas Operasi Batch S3 guna menyalin objek dari bucket bertujuan umum ke bucket direktori.

> **Note**
>
> Batas berikut berlaku untuk tugas impor:
>
> - Bucket sumber dan bucket tujuan harus sama Wilayah AWS dan akun.
> - Bucket sumber tidak dapat berupa bucket direktori.
> - Objek yang lebih besar dari 5GB tidak didukung dan akan dihilangkan dari operasi salin.
> - Objek di kelas penyimpanan Glacier Flexible Retrieval, Glacier Deep Archive, Intelligent-Tiering Archive Access, dan Intelligent-Tiering Deep Archive harus dipulihkan sebelum dapat diimpor.
> - Objek yang diimpor dengan algoritma MD5 checksum dikonversi untuk menggunakan CRC32 checksum.
> - Objek yang diimpor menggunakan kelas penyimpanan Express One Zone, yang memiliki struktur harga berbeda dari kelas penyimpanan yang digunakan oleh bucket tujuan umum. Pertimbangkan perbedaan biaya ini saat mengimpor banyak objek.

Saat mengonfigurasi tugas impor, Anda menentukan bucket sumber atau awalan tempat objek yang ada akan disalin. Anda juga menyediakan peran AWS Identity and Access Management (IAM) yang memiliki izin untuk mengakses objek sumber. Amazon S3 kemudian memulai tugas Operasi Batch yang menyalin objek dan secara otomatis menerapkan kelas penyimpanan dan pengaturan checksum yang sesuai.

Untuk mengonfigurasi tugas impor, Anda menggunakan konsol Amazon S3.

Menggunakan konsol Amazon S3

Mengimpor objek ke dalam bucket direktori

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket, lalu pilih tab bucket Direktori. Pilih tombol opsi di samping bucket direktori yang ingin Anda tempati untuk mengimpor objek.
3. Pilih Impor.
4. Untuk Sumber, masukkan bucket bertujuan umum (atau jalur bucket termasuk awalan) yang berisi objek yang ingin Anda impor. Untuk memilih bucket bertujuan umum yang ada dari daftar, pilih Jelajahi S3.

5. Untuk Izin mengakses dan menyalin objek sumber, lakukan salah satu hal berikut untuk menentukan peran IAM dengan izin yang diperlukan untuk mengimpor objek sumber Anda:

   - Untuk mengizinkan Amazon S3 membuat peran IAM baru atas nama Anda, pilih Buat peran IAM baru.
   - Untuk memilih peran IAM yang ada dari daftar, pilih Pilih dari peran IAM yang ada.
   - Untuk menentukan peran IAM yang ada dengan memasukkan Amazon Resource Name (ARN), pilih Masukkan ARN peran IAM, lalu masukkan ARN di bidang yang sesuai.

6. Tinjau informasi yang ditampilkan di bagian Tujuan dan Pengaturan objek yang disalin. Jika informasi di bagian Tujuan sudah benar, pilih Impor untuk memulai tugas penyalinan.

   Konsol Amazon S3 menampilkan status tugas baru Anda di halaman Operasi Batch. Untuk informasi selengkapnya tentang tugas, pilih tombol opsi di sebelah nama tugas, lalu pada menu Tindakan, pilih Lihat detail. Untuk membuka bucket direktori tempat objek akan diimpor, pilih Lihat tujuan impor.

Bekerja dengan Siklus Hidup S3 untuk bucket direktori

Siklus Hidup S3 membantu Anda menyimpan objek di S3 Express One Zone dalam bucket direktori dengan biaya efektif dengan menghapus objek kedaluwarsa atas nama Anda. Untuk mengelola siklus hidup objek Anda, buat konfigurasi Siklus Hidup S3 untuk bucket direktori Anda. Konfigurasi Siklus Hidup S3 adalah serangkaian aturan yang menetapkan tindakan yang diterapkan Amazon S3 pada sekelompok objek. Anda dapat mengatur konfigurasi Siklus Hidup Amazon S3 pada bucket direktori dengan menggunakan AWS Command Line Interface (AWS CLI), API REST AWS SDKs Amazon S3, dan. AWS CloudFormation

Dalam konfigurasi siklus hidup, Anda menggunakan aturan untuk menentukan tindakan yang ingin dilakukan Amazon S3 pada objek Anda. Untuk objek yang disimpan dalam bucket direktori, Anda dapat membuat aturan siklus hidup untuk objek kedaluwarsa seiring bertambahnya usia. Anda juga dapat membuat aturan siklus hidup untuk menghapus unggahan multibagian yang tidak lengkap dalam bucket direktori pada frekuensi harian.

Ketika Anda menambahkan konfigurasi Siklus Hidup ke dalam bucket, aturan konfigurasi berlaku untuk objek yang ada dan objek yang Anda tambahkan kemudian. Misalnya, jika Anda menambahkan aturan konfigurasi Siklus Hidup hari ini dengan tindakan kedaluwarsa yang menyebabkan objek dengan awalan tertentu kedaluwarsa 30 hari setelah pembuatan, S3 akan

mengantri untuk menghapus objek yang ada yang berusia lebih dari 30 hari dan yang memiliki awalan yang ditentukan.

Bagaimana Siklus Hidup S3 untuk bucket direktori berbeda

Untuk objek dalam bucket direktori, Anda dapat membuat aturan siklus hidup untuk menghapus objek kedaluwarsa dan menghapus unggahan multibagian yang tidak lengkap. Namun, Siklus Hidup S3 untuk bucket direktori tidak mendukung tindakan transisi antar kelas penyimpanan.

CreateSession

Lifecycle menggunakan operasi publik `DeleteObject` dan `DeleteObjects` API untuk menghapus objek dalam bucket direktori. Untuk menggunakan operasi API ini, S3 Lifecycle akan menggunakan `CreateSession` API untuk membuat kredensyal keamanan sementara untuk mengakses objek dalam bucket direktori. Untuk informasi selengkapnya, lihat `CreateSession`di Referensi API Amazon S3.

Jika Anda memiliki kebijakan aktif yang menolak izin hapus ke prinsipal siklus hidup, ini akan mencegah Anda mengizinkan Siklus Hidup S3 untuk menghapus objek atas nama Anda.

Menggunakan kebijakan bucket untuk memberikan izin kepada prinsipal layanan Siklus Hidup S3

Kebijakan bucket berikut memberikan izin untuk mengizinkan CreateSession panggilan dengan `ReadWrite` sesi default dan mengizinkan prinsipal layanan siklus hidup.

Example – Kebijakan bucket untuk mengizinkan panggilan **`CreateSession`** dengan sesi **`ReadWrite`** default

```
{
  "Version":"2008-10-17",
  "Statement":[
     {
        "Effect":"Allow",
        "Principal": {
           "Service":"lifecycle.s3.amazonaws.com"
         },
         "Action":"s3express:CreateSession",
         "Condition": {
            "StringEquals": {
               "s3express:SessionMode": "ReadWrite"
             }
```

```
            },
            "Resource":"arn:aws:s3express:us-east-2:412345678921:bucket/amzn-s3-demo-
bucket--use2-az2--x-s3"
        }
    ]
}
```

Memantau aturan siklus hidup

Untuk objek yang disimpan dalam bucket direktori, S3 Lifecycle menghasilkan log peristiwa AWS CloudTrail manajemen dan data. Untuk informasi selengkapnya, lihat contoh file CloudTrail log untuk S3 Express One Zone.

Untuk informasi selengkapnya tentang membuat konfigurasi siklus hidup dan memecahkan masalah terkait Siklus Hidup S3, lihat topik berikut:

Topik

- Membuat dan mengelola konfigurasi Siklus Hidup untuk bucket direktori
- Memecahkan masalah Siklus Hidup S3 untuk bucket direktori

Membuat dan mengelola konfigurasi Siklus Hidup untuk bucket direktori

Anda dapat membuat konfigurasi siklus hidup untuk bucket direktori dengan menggunakan AWS Command Line Interface (AWS CLI), AWS SDKs dan REST. APIs

Menggunakan AWS CLI

Anda dapat menggunakan AWS CLI perintah berikut untuk mengelola konfigurasi Siklus Hidup S3:

- `put-bucket-lifecycle-configuration`
- `get-bucket-lifecycle-configuration`
- `delete-bucket-lifecycle`

Untuk petunjuk cara menyiapkan AWS CLI, lihat Mengembangkan dengan Amazon S3 menggunakan AWS CLI di Referensi API Amazon S3.

Konfigurasi Amazon Siklus Hidup S3 adalah berkas XML. Tetapi ketika Anda menggunakan AWS CLI, Anda tidak dapat menentukan format XMLnya. Anda harus menentukan format JSON sebagai

gantinya. Berikut ini adalah contoh konfigurasi siklus hidup XMLdan konfigurasi JSON setara yang dapat Anda tentukan dalam sebuah perintah. AWS CLI

Contoh AWS CLI berikut menempatkan kebijakan konfigurasi siklus hidup pada bucket direktori. Kebijakan ini menetapkan bahwa semua objek yang memiliki awalan bertanda (myprefix) dan ukuran objek yang ditentukan akan kedaluwarsa setelah 7 hari. Untuk menggunakan contoh ini, ganti masing-masing *user input placeholder* dengan informasi Anda sendiri.

Simpan kebijakan konfigurasi siklus ke file JSON. Dalam contoh ini, file tersebut bernama lifecycle1.json.

Example

JSON

```
{
    "Rules": [
        {
        "Expiration": {
            "Days": 7
        },
        "ID": "Lifecycle expiration rule",
        "Filter": {
            "And": {
                "Prefix": "myprefix/",
                "ObjectSizeGreaterThan": 500,
                "ObjectSizeLessThan": 64000
            }
        },
        "Status": "Enabled"
    }
    ]
}
```

Kirim file JSON sebagai bagian dari perintah CLI `put-bucket-lifecycle-configuration`. Untuk menggunakan perintah ini, ganti masing-masing *user input placeholder* dengan informasi Anda sendiri.

```
aws s3api put-bucket-lifecycle-configuration --region us-west-2 --profile default
 --bucket amzn-s3-demo-bucket--usw2-az1--x-s3 --lifecycle-configuration file://lc-
policy.json --checksum-algorithm crc32c
```

Example

XML

```
<LifecycleConfiguration>
    <Rule>
        <ID>Lifecycle expiration rule</ID>
        <Filter>
            <And>
                <Prefix>myprefix/</Prefix>
                <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
                <ObjectSizeLessThan>64000</ObjectSizeLessThan>
            </And>
        </Filter>
        <Status>Enabled</Status>
        <Expiration>
             <Days>7</Days>
        </Expiration>
    </Rule>
</LifecycleConfiguration>
```

Menggunakan AWS SDKs

SDK for Java

Example

```
import
 software.amazon.awssdk.services.s3.model.PutBucketLifecycleConfigurationRequest;
import
 software.amazon.awssdk.services.s3.model.PutBucketLifecycleConfigurationResponse;
import software.amazon.awssdk.services.s3.model.ChecksumAlgorithm;
import software.amazon.awssdk.services.s3.model.BucketLifecycleConfiguration;
import software.amazon.awssdk.services.s3.model.LifecycleRule;
import software.amazon.awssdk.services.s3.model.LifecycleRuleFilter;
import software.amazon.awssdk.services.s3.model.LifecycleExpiration;
import software.amazon.awssdk.services.s3.model.LifecycleRuleAndOperator;
import
 software.amazon.awssdk.services.s3.model.GetBucketLifecycleConfigurationResponse;
import
 software.amazon.awssdk.services.s3.model.GetBucketLifecycleConfigurationRequest;
import software.amazon.awssdk.services.s3.model.DeleteBucketLifecycleRequest;
```

```
import software.amazon.awssdk.services.s3.model.DeleteBucketLifecycleResponse;
import software.amazon.awssdk.services.s3.model.AbortIncompleteMultipartUpload;

// PUT a Lifecycle policy
LifecycleRuleFilter objectExpirationFilter =
 LifecycleRuleFilter.builder().and(LifecycleRuleAndOperator.builder().prefix("dir1/").object
LifecycleRuleFilter mpuExpirationFilter =
 LifecycleRuleFilter.builder().prefix("dir2/").build();

LifecycleRule objectExpirationRule =
 LifecycleRule.builder().id("lc").filter(objectExpirationFilter).status("Enabled").expiratio
                    .days(10)
                    .build())
                .build();
LifecycleRule mpuExpirationRule = LifecycleRule.builder().id("lc-
mpu").filter(mpuExpirationFilter).status("Enabled").abortIncompleteMultipartUpload(AbortInco
                        .daysAfterInitiation(10)
                        .build())
                .build();

PutBucketLifecycleConfigurationRequest putLifecycleRequest =
 PutBucketLifecycleConfigurationRequest.builder()
                .bucket("amzn-s3-demo-bucket--usw2-az1--x-s3")
                .checksumAlgorithm(ChecksumAlgorithm.CRC32)
                .lifecycleConfiguration(
                        BucketLifecycleConfiguration.builder()
                                .rules(objectExpirationRule, mpuExpirationRule)
                                .build()
                ).build();

PutBucketLifecycleConfigurationResponse resp =
 client.putBucketLifecycleConfiguration(putLifecycleRequest);

// GET the Lifecycle policy
GetBucketLifecycleConfigurationResponse getResp =
 client.getBucketLifecycleConfiguration(GetBucketLifecycleConfigurationRequest.builder().buc
s3-demo-bucket--usw2-az1--x-s3").build());

// DELETE the Lifecycle policy
DeleteBucketLifecycleResponse delResp =
 client.deleteBucketLifecycle(DeleteBucketLifecycleRequest.builder().bucket("amzn-
s3-demo-bucket--usw2-az1--x-s3").build());
```

SDK for Go

Example

```
package main

import (
    "context"
    "log"

    "github.com/aws/aws-sdk-go-v2/aws"
    "github.com/aws/aws-sdk-go-v2/config"
    "github.com/aws/aws-sdk-go-v2/service/s3"
    "github.com/aws/aws-sdk-go-v2/service/s3/types"
)
// PUT a Lifecycle policy
func putBucketLifecycleConfiguration(client *s3.Client, bucketName string ) error {
    lifecycleConfig := &s3.PutBucketLifecycleConfigurationInput{
        Bucket: aws.String(bucketName),
        LifecycleConfiguration: &types.BucketLifecycleConfiguration{
            Rules: []types.LifecycleRule{
                {
                    ID:     aws.String("lc"),
                    Filter: &types.LifecycleRuleFilter{
                        And: &types.LifecycleRuleAndOperator{
                            Prefix: aws.String("foo/"),
                            ObjectSizeGreaterThan: aws.Int64(1000000),
                            ObjectSizeLessThan:    aws.Int64(100000000),
                            },
                        },

                    Status: types.ExpirationStatusEnabled,
                    Expiration: &types.LifecycleExpiration{
                        Days: aws.Int32(int32(1)),
                    },
                },
                {
                    ID:     aws.String("abortmpu"),
                    Filter: &types.LifecycleRuleFilter{
                        Prefix: aws.String("bar/"),
                    },
                    Status: types.ExpirationStatusEnabled,
                    AbortIncompleteMultipartUpload:
 &types.AbortIncompleteMultipartUpload{
```

```
                        DaysAfterInitiation: aws.Int32(int32(5)),
                    },
                },
            },
        },
    }
    _, err := client.PutBucketLifecycleConfiguration(context.Background(),
 lifecycleConfig)
    return err
}
// Get the Lifecycle policy
func getBucketLifecycleConfiguration(client *s3.Client, bucketName string) error {
    getLifecycleConfig := &s3.GetBucketLifecycleConfigurationInput{
        Bucket: aws.String(bucketName),
    }

    resp, err := client.GetBucketLifecycleConfiguration(context.Background(),
 getLifecycleConfig)
    if err != nil {
        return err
    }
    return nil
}
// Delete the Lifecycle policy
func deleteBucketLifecycleConfiguration(client *s3.Client, bucketName string) error
 {
    deleteLifecycleConfig := &s3.DeleteBucketLifecycleInput{
        Bucket: aws.String(bucketName),
    }
    _, err := client.DeleteBucketLifecycle(context.Background(),
 deleteLifecycleConfig)
    return err
}
func main() {
    cfg, err := config.LoadDefaultConfig(context.Background(),
 config.WithRegion("us-west-2")) // Specify your region here
    if err != nil {
        log.Fatalf("unable to load SDK config, %v", err)
    }
    s3Client := s3.NewFromConfig(cfg)
    bucketName := "amzn-s3-demo-bucket--usw2-az1--x-s3"
    putBucketLifecycleConfiguration(s3Client, bucketName)
    getBucketLifecycleConfiguration(s3Client, bucketName)
    deleteBucketLifecycleConfiguration(s3Client, bucketName)
```

```
    getBucketLifecycleConfiguration(s3Client, bucketName)
}
```

SDK for .NET

Example

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class LifecycleTest
    {
        private const string bucketName = "amzn-s3-demo-bucket--usw2-az1--x-s3";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 client;
        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            AddUpdateDeleteLifecycleConfigAsync().Wait();
        }

        private static async Task AddUpdateDeleteLifecycleConfigAsync()
        {
            try
            {
                var lifeCycleConfiguration = new LifecycleConfiguration()
                {
                    Rules = new List <LifecycleRule>
                        {
                            new LifecycleRule
                            {
                                 Id = "delete rule",
                                  Filter = new LifecycleFilter()
                                 {
                                     LifecycleFilterPredicate = new
 LifecyclePrefixPredicate()
```

```
                                    {
                                        Prefix = "projectdocs/"
                                    }
                                },
                                Status = LifecycleRuleStatus.Enabled,
                                Expiration = new LifecycleRuleExpiration()
                                {
                                      Days = 10
                                }
                            }
                        }
                };

                // Add the configuration to the bucket.
                await AddExampleLifecycleConfigAsync(client,
lifeCycleConfiguration);

                // Retrieve an existing configuration.
                lifeCycleConfiguration = await RetrieveLifecycleConfigAsync(client);

                // Add a new rule.
                lifeCycleConfiguration.Rules.Add(new LifecycleRule
                {
                    Id = "mpu abort rule",
                    Filter = new LifecycleFilter()
                    {
                        LifecycleFilterPredicate = new LifecyclePrefixPredicate()
                        {
                            Prefix = "YearlyDocuments/"
                        }
                    },
                    Expiration = new LifecycleRuleExpiration()
                    {
                        Days = 10
                    },
                    AbortIncompleteMultipartUpload = new
LifecycleRuleAbortIncompleteMultipartUpload()
                    {
                        DaysAfterInitiation = 10
                    }
                });

                // Add the configuration to the bucket.
```

```
                await AddExampleLifecycleConfigAsync(client,
lifeCycleConfiguration);

                // Verify that there are now two rules.
                lifeCycleConfiguration = await RetrieveLifecycleConfigAsync(client);
                Console.WriteLine("Expected # of rulest=2; found:{0}",
lifeCycleConfiguration.Rules.Count);

                // Delete the configuration.
                await RemoveLifecycleConfigAsync(client);

                // Retrieve a nonexistent configuration.
                lifeCycleConfiguration = await RetrieveLifecycleConfigAsync(client);

            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered ***. Message:'{0}' when writing
an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
        }

        static async Task AddExampleLifecycleConfigAsync(IAmazonS3 client,
LifecycleConfiguration configuration)
        {

            PutLifecycleConfigurationRequest request = new
PutLifecycleConfigurationRequest
            {
                BucketName = bucketName,
                Configuration = configuration
            };
            var response = await client.PutLifecycleConfigurationAsync(request);
        }

        static async Task <LifecycleConfiguration>
RetrieveLifecycleConfigAsync(IAmazonS3 client)
        {
```

```
            GetLifecycleConfigurationRequest request = new
 GetLifecycleConfigurationRequest
            {
                BucketName = bucketName
            };
            var response = await client.GetLifecycleConfigurationAsync(request);
            var configuration = response.Configuration;
            return configuration;
        }

        static async Task RemoveLifecycleConfigAsync(IAmazonS3 client)
        {
            DeleteLifecycleConfigurationRequest request = new
 DeleteLifecycleConfigurationRequest
            {
                BucketName = bucketName
            };
            await client.DeleteLifecycleConfigurationAsync(request);
        }
    }
}
```

SDK for Python

Example

```
import boto3

client = boto3.client("s3", region_name="us-west-2")
bucket_name = 'amzn-s3-demo-bucket--usw2-az1--x-s3'

client.put_bucket_lifecycle_configuration(
    Bucket=bucket_name,
    ChecksumAlgorithm='CRC32',
    LifecycleConfiguration={
        'Rules': [
            {
                'ID': 'lc',
                'Filter': {
                    'And': {
                        'Prefix': 'foo/',
                        'ObjectSizeGreaterThan': 1000000,
                        'ObjectSizeLessThan': 100000000,
```

```
                    }
                },
                'Status': 'Enabled',
                'Expiration': {
                    'Days': 1
                }
            },
            {
                'ID': 'abortmpu',
                'Filter': {
                    'Prefix': 'bar/'
                },
                'Status': 'Enabled',
                'AbortIncompleteMultipartUpload': {
                    'DaysAfterInitiation': 5
                }
            }
        ]
    }
)

result = client.get_bucket_lifecycle_configuration(
    Bucket=bucket_name
)

client.delete_bucket_lifecycle(
    Bucket=bucket_name
)
```

Memecahkan masalah Siklus Hidup S3 untuk bucket direktori

Topik

- Saya mengatur konfigurasi siklus hidup saya tetapi objek di ember direktori saya tidak kedaluwarsa
- Bagaimana cara memantau tindakan yang diambil oleh aturan siklus hidup saya?

Saya mengatur konfigurasi siklus hidup saya tetapi objek di ember direktori saya tidak kedaluwarsa

Siklus Hidup S3 untuk bucket direktori menggunakan publik APIs untuk menghapus objek di S3 Express One Zone. Untuk menggunakan public level objek APIs, Anda harus memberikan izin `CreateSession` dan mengizinkan izin Siklus Hidup S3 untuk menghapus objek Anda. Jika Anda

memiliki kebijakan aktif yang menolak penghapusan, ini akan mencegah Anda mengizinkan Siklus Hidup S3 menghapus objek atas nama Anda.

Penting untuk mengonfigurasi kebijakan bucket Anda dengan benar untuk memastikan bahwa objek yang ingin Anda hapus memenuhi syarat untuk kedaluwarsa. Anda dapat memeriksa AWS CloudTrail log Anda untuk `AccessDenied` Trails untuk pemanggilan `CreateSession` API CloudTrail untuk memverifikasi apakah akses telah ditolak. Memeriksa CloudTrail log Anda dapat membantu Anda dalam memecahkan masalah akses dan mengidentifikasi akar penyebab kesalahan akses ditolak. Anda kemudian dapat memperbaiki kontrol akses yang salah dengan memperbarui kebijakan yang relevan.

Jika Anda mengonfirmasi bahwa kebijakan bucket disetel dengan benar dan masih mengalami masalah, sebaiknya tinjau aturan siklus hidup untuk memastikan bahwa kebijakan tersebut diterapkan pada subset objek yang tepat.

Bagaimana cara memantau tindakan yang diambil oleh aturan siklus hidup saya?

Anda dapat menggunakan log peristiwa AWS CloudTrail data untuk memantau tindakan yang diambil oleh Siklus Hidup S3 di bucket direktori. Untuk informasi selengkapnya, lihat contoh file CloudTrail log.

Menggunakan Operasi Batch dengan bucket direktori

Anda dapat menggunakan Operasi Batch Amazon S3 untuk melakukan operasi pada objek yang disimpan dalam bucket S3. Untuk mempelajari Operasi Batch S3, lihat Melakukan operasi batch skala besar pada objek Amazon S3.

Topik berikut membahas melakukan operasi batch pada objek yang disimpan di kelas penyimpanan S3 Express One Zone dalam bucket direktori.

Topik

- Menggunakan Operasi Batch dengan bucket direktori
- Perbedaan utama

Menggunakan Operasi Batch dengan bucket direktori

Anda dapat melakukan operasi Salin dan operasi AWS Lambda fungsi Invoke pada objek yang disimpan dalam bucket direktori. Dengan Copy, Anda dapat menyalin objek di antara bucket dengan

tipe yang sama (misalnya, dari bucket direktori ke bucket direktori). Anda juga dapat menyalin antara bucket tujuan umum dan bucket direktori. Dengan AWS Lambda fungsi Invoke, Anda dapat menggunakan fungsi Lambda untuk melakukan tindakan pada objek di bucket direktori Anda dengan kode yang Anda tentukan.

Menyalin objek

Anda dapat menyalin antara jenis bucket yang sama atau antara bucket direktori dan bucket tujuan umum. Saat menyalin ke bucket direktori, Anda harus menggunakan format Amazon Resource Name (ARN) yang benar untuk jenis bucket ini. Format ARN untuk bucket direktori adalah `arn:aws:s3express:region:account-id:bucket/bucket-base-name--x-s3`.

> **Note**
>
> Menyalin objek di berbagai objek Wilayah AWS tidak didukung saat bucket sumber atau tujuan berada di Zona AWS Lokal. Bucket sumber dan tujuan harus memiliki induk Wilayah AWS yang sama. Bucket sumber dan tujuan dapat berupa jenis lokasi bucket yang berbeda (Availability Zone atau Local Zone).

Anda juga dapat mengisi bucket direktori Anda dengan data menggunakan tindakan Impor di konsol S3. Impor adalah metode yang disederhanakan untuk membuat pekerjaan Operasi Batch guna menyalin objek dari bucket tujuan umum ke bucket direktori. Untuk tugas Impor salinan dari bucket tujuan umum ke bucket direktori, S3 secara otomatis menghasilkan manifes. Untuk informasi selengkapnya, lihat Mengimpor objek ke bucket direktori dan Menentukan manifes.

Memanggil fungsi Lambda () **`LambdaInvoke`**

Ada persyaratan khusus untuk menggunakan Operasi Batch untuk menginvokasi fungsi Lambda yang bertindak pada bucket direktori. Misalnya, Anda harus menyusun permintaan Lambda Anda dengan menggunakan skema pemanggilan v2 JSON, dan menentukan InvocationSchemaVersion 2.0 kapan Anda membuat pekerjaan. Untuk informasi selengkapnya, lihat Memanggil AWS Lambda fungsi.

Perbedaan utama

Berikut ini adalah daftar perbedaan utama saat Anda menggunakan Operasi Batch untuk melakukan operasi massal pada objek yang disimpan dalam bucket direktori dengan kelas penyimpanan S3 Express One Zone:

- Untuk bucket direktori, SSE-S3 dan enkripsi sisi server dengan kunci AWS Key Management Service (AWS KMS) (SSE-KMS) didukung. Jika Anda membuat `CopyObject` permintaan yang menetapkan untuk menggunakan enkripsi sisi server dengan kunci yang disediakan pelanggan (SSE-C) pada bucket direktori (sumber atau tujuan), respons akan mengembalikan kesalahan HTTP. `400 (Bad Request)`

  Sebaiknya enkripsi default bucket menggunakan konfigurasi enkripsi yang diinginkan dan Anda tidak mengganti enkripsi default bucket dalam `CreateSession` permintaan atau permintaan PUT objek. Kemudian, objek baru secara otomatis dienkripsi dengan pengaturan enkripsi yang diinginkan. Untuk informasi selengkapnya tentang perilaku pengesampingan enkripsi dalam bucket direktori dan cara mengenkripsi salinan objek baru dalam bucket direktori dengan SSE-KMS, lihat Menentukan enkripsi sisi server dengan unggahan objek baru. AWS KMS

  Kunci Bucket S3 tidak didukung, saat Anda menyalin objek terenkripsi SSE-KMS dari bucket tujuan umum ke bucket direktori, dari bucket direktori ke bucket tujuan umum, atau antar bucket direktori, melalui operasi dalam Operasi Batch. Copy Dalam hal ini, Amazon S3 melakukan panggilan ke AWS KMS setiap kali permintaan salinan dibuat untuk objek yang dienkripsi KMS. Untuk informasi selengkapnya tentang penggunaan SSE-KMS pada bucket direktori, lihat dan. Pengaturan dan pemantauan enkripsi default untuk bucket direktori Menggunakan enkripsi sisi server dengan AWS KMS kunci (SSE-KMS) di bucket direktori
- Objek dalam bucket direktori tidak dapat ditandai. Anda hanya dapat menentukan set tanda kosong. Secara default, Operasi Batch menyalin tanda. Jika Anda menyalin objek yang memiliki tag antara bucket tujuan umum dan bucket direktori, Anda akan menerima respons. `501 (Not Implemented)`
- S3 Express One Zone menawarkan Anda opsi untuk memilih algoritma checksum yang digunakan untuk memvalidasi data Anda selama unggahan atau unduhan. Anda dapat memilih salah satu algoritma pemeriksaan integritas data Secure Hash Algorithms (SHA) atau Cyclic Redundancy Check (CRC) berikut:,, SHA-1, dan SHA-256. CRC32 CRC32 MD5checksum berbasis tidak didukung dengan kelas penyimpanan S3 Express One Zone.
- Secara default, semua bucket Amazon S3 menyetel setelan Kepemilikan Objek S3 ke pemilik bucket yang diberlakukan dan daftar kontrol akses () dinonaktifkan. ACLs Untuk bucket direktori, pengaturan ini tidak dapat dimodifikasi. Anda dapat menyalin objek dari bucket tujuan umum ke bucket direktori. Namun, Anda tidak dapat menimpa ACL default saat menyalin ke atau dari bucket direktori.
- Terlepas dari cara Anda menentukan manifes, daftar itu harus disimpan di dalam sebuah bucket yang bersifat umum. Operasi Batch tidak dapat mengimpor manifes yang ada dari (atau

menyimpan manifes yang dihasilkan ke) bucket direktori. Namun, objek yang dijelaskan dalam manifes dapat disimpan dalam bucket direktori.

- Operasi Batch tidak dapat menentukan bucket direktori sebagai lokasi dalam laporan Inventaris S3. Laporan inventaris tidak mendukung bucket direktori. Anda dapat membuat file manifes untuk objek dalam bucket direktori dengan menggunakan operasi `ListObjectsV2` API untuk mencantumkan objek. Anda kemudian dapat memasukkan daftar dalam file CSV.

Memberi Akses

Untuk melakukan tugas penyalinan, Anda harus memiliki izin berikut:

- Untuk menyalin objek dari satu bucket direktori ke bucket direktori lain, Anda harus memiliki izin `s3express:CreateSession`.
- Untuk menyalin objek dari bucket direktori ke bucket tujuan umum, Anda harus memiliki izin `s3express:CreateSession` dan izin `s3:PutObject`untuk menulis salinan objek ke bucket tujuan.
- Untuk menyalin objek dari bucket tujuan umum ke bucket direktori, Anda harus memiliki `s3express:CreateSession` izin dan `s3:GetObject` izin untuk membaca objek sumber yang sedang disalin.

  Untuk informasi selengkapnya, lihat CopyObject dalam Referensi API Amazon Simple Storage Service.
- Untuk menginvokasi fungsi Lambda, Anda harus memberikan izin ke sumber daya berdasarkan fungsi Lambda Anda. Untuk menentukan izin mana yang diperlukan, periksa operasi API yang sesuai.

Menambahkan data ke objek dalam ember direktori

Anda dapat menambahkan data ke akhir objek yang ada yang disimpan di kelas penyimpanan S3 Express One Zone di bucket direktori. Kami menyarankan Anda menggunakan kemampuan untuk menambahkan data ke objek jika data ditulis terus menerus selama periode waktu tertentu atau jika Anda perlu membaca objek saat Anda menulis ke objek. Menambahkan data ke objek adalah umum untuk kasus penggunaan seperti menambahkan entri log baru ke file log atau menambahkan segmen video baru ke file video saat ditranskode kemudian dialirkan. Dengan menambahkan data ke objek, Anda dapat menyederhanakan aplikasi yang sebelumnya menggabungkan data dalam penyimpanan lokal sebelum menyalin objek akhir ke Amazon S3.

Tidak ada persyaratan ukuran minimum untuk data yang dapat Anda tambahkan ke objek. Namun, ukuran maksimum data yang dapat Anda tambahkan ke objek dalam satu permintaan adalah 5GB. Ini adalah batas yang sama dengan ukuran permintaan terbesar saat mengunggah data menggunakan API Amazon S3 apa pun.

Dengan setiap operasi append yang berhasil, Anda membuat bagian dari objek dan setiap objek dapat memiliki hingga 10.000 bagian. Ini berarti Anda dapat menambahkan data ke objek hingga 10.000 kali. Jika objek dibuat menggunakan unggahan multipart S3, setiap bagian yang diunggah dihitung dengan total maksimum 10.000 bagian. Misalnya, Anda dapat menambahkan hingga 9.000 kali ke objek yang dibuat oleh unggahan multipart yang terdiri dari 1.000 bagian.

> **Note**
>
> Jika Anda mencapai batas bagian, Anda akan menerima TooManyPartskesalahan. Anda dapat menggunakan `CopyObject` API untuk mengatur ulang hitungan.

Jika Anda ingin mengunggah bagian ke objek secara paralel dan Anda tidak perlu membaca bagian saat bagian sedang diunggah, kami sarankan Anda menggunakan unggahan multipart Amazon S3. Untuk informasi selengkapnya, lihat Menggunakan unggahan multibagian.

Menambahkan data ke objek hanya didukung untuk objek dalam bucket direktori yang disimpan di kelas penyimpanan S3 Express One Zone. Untuk informasi selengkapnya tentang S3 Express One Zone Zone, lihat Memulai S3 Express One Zone.

Untuk mulai menambahkan data ke objek di bucket direktori Anda, Anda dapat menggunakan, AWS SDKs AWS CLI, dan API. `PutObject` Saat Anda membuat `PutObject` permintaan, Anda mengatur `x-amz-write-offset-bytes` header ke ukuran objek yang Anda tambahkan. Untuk menggunakan operasi `PutObject` API, Anda harus menggunakan `CreateSession` API untuk membuat kredensyal keamanan sementara untuk mengakses objek di bucket direktori Anda. Untuk informasi selengkapnya, lihat `PutObject`dan `CreateSession`di Referensi API Amazon S3.

Setiap operasi append yang berhasil ditagih sebagai `PutObject` permintaan. Untuk mempelajari lebih lanjut tentang harga, lihat `Amazon S3 pricing`.

> **Note**
>
> Dimulai dengan rilis 1.12, Mountpoint untuk Amazon S3 mendukung penambahan data ke objek yang disimpan di S3 Express One Zone. Untuk memulai, Anda harus ikut serta

dengan menyetel `--incremental-upload` bendera. Untuk informasi lebih lanjut tentang Mountpoint, lihat Bekerja dengan Mountpoint.

Jika Anda menggunakan algoritma CRC (Cyclic Redundancy Check) saat mengunggah data yang ditambahkan, Anda dapat mengambil checksum berbasis CRC objek lengkap menggunakan permintaan atau. `HeadObject` `GetObject` Jika Anda menggunakan algoritma SHA-1 atau SHA-256 saat mengunggah data yang ditambahkan, Anda dapat mengambil checksum dari bagian yang ditambahkan dan memverifikasi integritasnya menggunakan checksum SHA yang dikembalikan pada tanggapan sebelumnya. PutObject Untuk informasi selengkapnya, lihat Perlindungan dan enkripsi data.

Menambahkan data ke objek Anda dengan menggunakan AWS CLI AWS SDKs , dan REST API

Anda dapat menambahkan data ke objek Anda dengan menggunakan AWS Command Line Interface (AWS CLI), AWS SDKs dan REST API.

Menggunakan AWS CLI

`put-object`Contoh perintah berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk menambahkan data ke objek. Untuk menjalankan perintah ini, ganti *`user input placeholders`* dengan informasi Anda sendiri

```
aws s3api put-object --bucket amzn-s3-demo-bucket--azid--x-s3 --key sampleinput/
file001.bin --body bucket-seed/file001.bin --write-offset-bytes size-of-sampleinput/
file001.bin
```

Menggunakan AWS SDKs

SDK for Java

Anda dapat menggunakan AWS SDK untuk Java untuk menambahkan data ke objek Anda.

```
var putObjectRequestBuilder = PutObjectRequest.builder()
                                              .key(key)
                                              .bucket(bucketName)
                                              .writeOffsetBytes(9);
var response = s3Client.putObject(putObjectRequestBuilder.build());
```

SDK for Python

```
s3.put_object(Bucket='amzn-s3-demo-bucket--use2-az2--x-s3', Key='2024-11-05-sdk-
test', Body=b'123456789', WriteOffsetBytes=9)
```

Penggunaan API REST

Anda dapat mengirim permintaan REST untuk menambahkan data ke objek. Untuk informasi selengkapnya, lihat `PutObject`.

Mengganti nama objek dalam ember direktori

Dengan menggunakan `RenameObject` operasi ini, Anda dapat mengganti nama objek yang ada secara atomik di bucket direktori yang menggunakan kelas penyimpanan S3 Express One Zone, tanpa pergerakan data apa pun. Anda dapat mengganti nama objek dengan menentukan nama objek yang ada sebagai sumber dan nama baru objek sebagai tujuan dalam bucket direktori yang sama. Operasi `RenameObject` API tidak akan berhasil pada objek yang diakhiri dengan karakter pembatas garis miring (`/`). Untuk informasi selengkapnya, lihat Menamai objek Amazon S3.

`RenameObject`Operasi biasanya selesai dalam milidetik terlepas dari ukuran objek. Kemampuan ini mempercepat aplikasi seperti manajemen file log, pemrosesan media, dan analisis data. Selain itu, `RenameObject` mempertahankan semua properti metadata objek, termasuk kelas penyimpanan, jenis enkripsi, tanggal pembuatan, tanggal modifikasi terakhir, dan properti checksum.

> **Note**
>
> `RenameObject`hanya didukung untuk objek yang disimpan di kelas penyimpanan S3 Express One Zone.

Untuk memberikan akses ke `RenameObject` operasi, kami sarankan Anda menggunakan `CreateSession` operasi untuk otorisasi berbasis sesi. Secara khusus, Anda memberikan `s3express:CreateSession` izin ke bucket direktori dalam kebijakan bucket atau kebijakan berbasis identitas. Kemudian, Anda melakukan panggilan `CreateSession` API di bucket direktori untuk mendapatkan token sesi. Dengan token sesi di header permintaan Anda, Anda dapat membuat permintaan API untuk operasi ini. Setelah token sesi kedaluwarsa, Anda membuat panggilan `CreateSession` API lain untuk menghasilkan token sesi baru untuk digunakan. AWS CLI dan

AWS SDKs akan membuat dan mengelola sesi Anda termasuk menyegarkan token sesi secara otomatis untuk menghindari gangguan layanan ketika sesi berakhir. Untuk informasi selengkapnya tentang otorisasi, lihat `CreateSession`di Referensi API Amazon S3. Untuk mempelajari lebih lanjut tentang operasi API titik akhir Zonal, lihat Mengotorisasi operasi API titik akhir Zonal dengan. `CreateSession`

Jika Anda tidak ingin menimpa objek yang ada, Anda dapat menambahkan header `If-None-Match` bersyarat dengan nilai `'*'` dalam permintaan. `RenameObject` Amazon S3 mengembalikan `412 Precondition Failed` kesalahan jika nama objek sudah ada. Untuk informasi selengkapnya, lihat `RenameObject` di Referensi API Amazon S3.

`RenameObject`adalah operasi API titik akhir Zonal (operasi tingkat objek atau bidang data) yang dicatat. AWS CloudTrail Anda dapat menggunakan CloudTrail untuk mengumpulkan informasi tentang `RenameObject` operasi yang dilakukan pada objek Anda dalam ember direktori. Untuk informasi selengkapnya, lihat Logging with AWS CloudTrail untuk bucket direktori dan contoh file CloudTrail log untuk bucket direktori.

S3 Express One Zone adalah satu-satunya kelas penyimpanan yang mendukung`RenameObject`, yang harganya sama dengan`PUT`,, `COPY``POST`, dan `LIST` permintaan (per 1.000 permintaan) di S3 Express One Zone. Untuk informasi selengkapnya, lihat Harga Amazon S3.

Mengganti nama objek

Untuk mengganti nama objek di bucket direktori, Anda dapat menggunakan konsol Amazon S3, AWS CLI AWS SDKs, REST API, atau Mountpoint untuk Amazon S3 (versi 1.19.0 atau lebih tinggi).

Menggunakan konsol S3

Untuk mengganti nama objek dalam ember direktori

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi, pilih Bucket, lalu pilih tab Ember direktori. Arahkan ke bucket direktori Amazon S3 yang berisi objek yang ingin Anda ganti namanya.
3. Pilih kotak centang untuk objek yang ingin Anda ganti namanya.
4. Pada menu Tindakan, pilih Ganti nama objek.
5. Di kotak Nama objek baru, masukkan nama baru untuk objek tersebut.

> **Note**
>
> Jika Anda menentukan nama objek yang sama dengan objek yang ada, operasi akan gagal dan Amazon S3 mengembalikan kesalahan. `412 Precondition Failed` Panjang nama kunci objek tidak dapat melebihi 1.024 byte. Awalan yang termasuk dalam jumlah nama objek terhadap panjang total.

6. Pilih Ganti nama objek. Amazon S3 mengganti nama objek Anda.

Menggunakan AWS CLI

`rename-object`Contoh menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk mengganti nama objek. Untuk menjalankan perintah ini, ganti *`user input placeholders`* dengan informasi Anda sendiri

Contoh berikut menunjukkan cara mengganti nama objek dengan pemeriksaan bersyarat pada objek sumber. ETag

```
aws s3api rename-object \
    --bucket amzn-s3-demo-bucket--usw2-az1--x-s3 \
    --key new-file.txt \
    --rename-source amzn-s3-demo-bucket--usw2-az1--x-s3/original-file.txt \
    --source-if-match "\"a1b7c3d2e5f6\""
```

Perintah ini melakukan hal berikut:

- Mengganti nama objek dari *`original-file.txt`* ke *`new-file.txt`* dalam ember *`amzn-s3-demo-bucket--usw2-az1--x-s3`* direktori.
- Hanya melakukan penggantian nama jika objek sumber ETag cocok dengan "*`a1b7c3d4e5f6`*”.

Jika ETag tidak cocok, operasi akan gagal dengan `412 Precondition Failed` kesalahan.

Contoh berikut menunjukkan cara mengganti nama objek dengan pemeriksaan bersyarat pada nama objek yang ditentukan baru.

```
aws s3api rename-object \
    --bucket amzn-s3-demo-bucket--usw2-az1--x-s3 \
    --key new-file.txt \
```

```
    --rename-source amzn-s3-demo-bucket--usw2-az1--x-s3/original-file.txt \
    --destination-if-none-match "\"e5f3g7h8i9j0\""
```

Perintah ini melakukan hal berikut:

- Mengganti nama objek dari *original-file.txt* ke *new-file.txt* dalam ember *amzn-s3-demo-bucket--usw2-az1--x-s3* direktori.
- Hanya melakukan operasi ganti nama jika objek ada dan objek ETag tidak cocok "*e5f3g7h8i9j0*".

Jika objek sudah ada dengan nama yang ditentukan baru dan pencocokan ETag, operasi akan gagal dengan `412 Precondition Failed` kesalahan.

Menggunakan AWS SDKs

SDK for Java

Anda dapat menggunakan AWS SDK untuk Java untuk mengganti nama objek Anda. Untuk menggunakan contoh-contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri

Contoh berikut menunjukkan cara membuat `RenameObjectRequest` menggunakan AWS SDK for Java

```
String key = "key";
String newKey = "new-key";
String expectedETag = "e5f3g7h8i9j0";
RenameObjectRequest renameRequest = RenameObjectRequest.builder()
    .bucket(amzn-s3-demo-bucket--usw2-az1--x-s3)
    .key(newKey)
    .renameSource(key)
    .destinationIfMatch(e5f3g7h8i9j0)
    .build();
```

Kode ini melakukan hal berikut:

- Buat permintaan untuk mengganti nama objek dari "*key*" menjadi "*new-key*" di ember *amzn-s3-demo-bucket--usw2-az1--x-s3* direktori.
- Termasuk kondisi bahwa penggantian nama hanya akan terjadi jika objek ETag cocok "*e5f3g7h8i9j0*".

- Jika ETag tidak cocok atau objek tidak ada, operasi akan gagal.

Contoh berikut menunjukkan cara membuat `RenameObjectRequest` dengan kondisi none-match menggunakan AWS SDK for Java.

```
String key = "key";
String newKey = "new-key";
String noneMatchETag = "e5f3g7h8i9j0";
RenameObjectRequest renameRequest = RenameObjectRequest.builder()
    .bucket(amzn-s3-demo-bucket--usw2-az1--x-s3)
    .key(newKey)
    .renameSource(key)
    .destinationIfNoneMatch(noneMatchETag)
    .build();
```

Kode ini melakukan hal berikut:

- Membuat permintaan untuk mengganti nama objek dari "*key*" menjadi "*new-key*" di ember *amzn-s3-demo-bucket--usw2-az1--x-s3* direktori.
- Termasuk kondisi yang digunakan `.destinationIfNoneMatch(noneMatchETag)` yang memastikan penggantian nama hanya akan terjadi jika objek tujuan ETag tidak cocok "*e5f3g7h8i9j0*".

Operasi akan gagal dengan `412 Precondition Failed` kesalahan jika ada objek dengan nama yang ditentukan baru dan memiliki yang ditentukan ETag.

SDK for Python

Anda dapat menggunakan SDK untuk Python untuk mengganti nama objek Anda. Untuk menggunakan contoh ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

Contoh berikut menunjukkan cara mengganti nama objek menggunakan AWS SDK for Python (Boto3).

```
def basic_rename(bucket, source_key, destination_key):
    try:
        s3.rename_object(
            Bucket=amzn-s3-demo-bucket--usw2-az1--x-s3,
            Key=destination_key,
            RenameSource=f"{amzn-s3-demo-bucket--usw2-az1--x-s3}/{source_key}"
```

```
        )
        print(f"Successfully renamed {source_key} to {destination_key}")
    except ClientError as e:
        print(f"Error renaming object: {e}")
```

Kode ini melakukan hal berikut:

- Mengganti nama objek dari *source_key* ke *destination_key* dalam ember *amzn-s3-demo-bucket--usw2-az1--x-s3* direktori.
- Mencetak pesan sukses jika penggantian nama objek Anda berhasil atau mencetak pesan kesalahan jika gagal.

Contoh berikut menunjukkan cara mengganti nama objek dengan `SourceIfMatch` dan `DestinationIfNoneMatch` kondisi menggunakan AWS SDK for Python (Boto3).

```
def rename_with_conditions(bucket, source_key, destination_key, source_etag,
 dest_etag):
    try:
        s3.rename_object(
            Bucket=amzn-s3-demo-bucket--usw2-az1--x-s3,
            Key=destination_key,
            RenameSource=f"{amzn-s3-demo-bucket--usw2-az1--x-s3}/{source_key}",
            SourceIfMatch=source_ETag,
            DestinationIfNoneMatch=dest_ETag
        )
        print(f"Successfully renamed {source_key} to {destination_key} with
 conditions")
    except ClientError as e:
        print(f"Error renaming object: {e}")
```

Kode ini melakukan hal berikut:

- Melakukan operasi penggantian nama bersyarat dan menerapkan dua kondisi, `SourceIfMatch` dan. `DestinationIfNoneMatch` Kombinasi kondisi ini memastikan bahwa objek belum dimodifikasi dan bahwa objek belum ada dengan nama yang ditentukan baru.
- Mengganti nama objek dari *source_key* ke *destination_key* dalam ember *amzn-s3-demo-bucket--usw2-az1--x-s3* direktori.
- Mencetak pesan sukses jika penggantian nama objek Anda berhasil, atau mencetak pesan kesalahan jika gagal atau jika kondisi tidak terpenuhi.

SDK for Rust

Anda dapat menggunakan SDK for Rust untuk mengganti nama objek Anda. Untuk menggunakan contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

Contoh berikut menunjukkan cara mengganti nama objek di bucket *amzn-s3-demo-bucket--usw2-az1--x-s3* direktori menggunakan SDK untuk Rust.

```
async fn basic_rename_example(client: &Client) -> Result<(), Box<dyn Error>> {
    let response = client
        .rename_object()
        .bucket(" amzn-s3-demo-bucket--usw2-az1--x-s3")
        .key("new-name.txt")  // New name/path for the object
        .rename_source("old-name.txt")  // Original object name/path
        .send()
        .await?;
    Ok(())
}
```

Kode ini melakukan hal berikut:

- Membuat permintaan untuk mengganti nama objek dari "*old-name.tx*" menjadi "*new-name.txt*" di ember *amzn-s3-demo-bucket--usw2-az1--x-s3* direktori.
- Mengembalikan `Result` jenis untuk menangani potensi kesalahan.

Penggunaan API REST

Anda dapat mengirim permintaan REST untuk mengganti nama objek. Untuk informasi selengkapnya, lihat `RenameObject` di Referensi API Amazon S3.

Menggunakan Mountpoint untuk Amazon S3

Dimulai dengan versi 1.19.0 atau lebih tinggi, Mountpoint untuk Amazon S3 mendukung penggantian nama objek di S3 Express One Zone. Untuk informasi lebih lanjut tentang Mountpoint, lihat Bekerja dengan Mountpoint.

Mengunggah objek ke ember direktori

Setelah membuat bucket direktori Amazon S3, Anda dapat mengunggah objek ke dalamnya. Contoh berikut menunjukkan cara mengunggah objek ke bucket direktori dengan menggunakan konsol S3

dan file. AWS SDKs Untuk informasi tentang operasi upload objek massal dengan S3 Express One Zone, lihatManajemen objek.

Menggunakan konsol S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket direktori.
3. Pilih nama bucket yang ingin Anda unggah folder atau file Anda.
4. Dalam daftar Objek, pilih Unggah.
5. Pada halaman Unggah, lakukan salah satu hal berikut:
   - Seret dan lepas file dan folder ke area unggahan bertitik.
   - Pilih Tambahkan file atau Tambah folder, pilih file atau folder yang akan diunggah, lalu pilih Buka atau Unggah.
6. Di bawah Checksum, pilih fungsi Checksum yang ingin Anda gunakan.

   (Opsional) Jika Anda mengunggah satu objek yang berukuran kurang dari 16 MB, Anda juga dapat menentukan nilai checksum yang telah dihitung sebelumnya. Saat Anda memberikan nilai yang telah dihitung sebelumnya, Amazon S3 membandingkannya dengan nilai yang dihitung dengan menggunakan fungsi checksum yang dipilih. Jika nilai tidak cocok, unggahan tidak akan dimulai.
7. Opsi di bagian Izin dan Properti secara otomatis diatur ke pengaturan default dan tidak dapat dimodifikasi. Blokir Akses Publik diaktifkan secara otomatis, dan Versi S3 dan Kunci Objek S3 tidak dapat diaktifkan untuk bucket direktori.

   (Opsional) Jika Anda ingin menambahkan metadata dalam pasangan nilai kunci ke objek Anda, perluas bagian Properti, lalu di bagian Metadata, pilih Tambahkan metadata.
8. Untuk mengunggah file dan folder yang terdaftar, pilih Unggah.

   Amazon S3 mengunggah objek dan folder Anda. Setelah unggahan selesai, Anda melihat pesan sukses di halaman Unggahan: status.

Menggunakan AWS SDKs

SDK for Java 2.x

Example

```
public static void putObject(S3Client s3Client, String bucketName, String objectKey,
 Path filePath) {
       //Using File Path to avoid loading the whole file into memory
       try {
           PutObjectRequest putObj = PutObjectRequest.builder()
                   .bucket(bucketName)
                   .key(objectKey)
                   //.metadata(metadata)
                   .build();
           s3Client.putObject(putObj, filePath);
           System.out.println("Successfully placed " + objectKey +" into bucket
 "+bucketName);

       }

       catch (S3Exception e) {
           System.err.println(e.getMessage());
           System.exit(1);
       }
}
```

SDK for Python

Example

```
import boto3
import botocore
from botocore.exceptions import ClientError


def put_object(s3_client, bucket_name, key_name, object_bytes):
    """
    Upload data to a directory bucket.
    :param s3_client: The boto3 S3 client
    :param bucket_name: The bucket that will contain the object
    :param key_name: The key of the object to be uploaded
    :param object_bytes: The data to upload
    """
```

```
    try:
        response = s3_client.put_object(Bucket=bucket_name, Key=key_name,
                                Body=object_bytes)
        print(f"Upload object '{key_name}' to bucket '{bucket_name}'.")
        return response
    except ClientError:
        print(f"Couldn't upload object '{key_name}' to bucket '{bucket_name}'.")
        raise

def main():
    # Share the client session with functions and objects to benefit from S3 Express
 One Zone auth key
    s3_client = boto3.client('s3')
    # Directory bucket name must end with --zone-id--x-s3
    resp = put_object(s3_client, 'doc-bucket-example--use1-az5--x-s3', 'sample.txt',
 b'Hello, World!')
    print(resp)

if __name__ == "__main__":
    main()
```

Menggunakan AWS CLI

`put-object`Contoh perintah berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk meng-upload objek dari Amazon S3. Untuk menjalankan perintah ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3api put-object --bucket bucket-base-name--zone-id--x-s3 --key sampleinut/
file001.bin --body bucket-seed/file001.bin
```

Untuk informasi selengkapnya, lihat put-object dalam AWS CLI Referensi Perintah.

Topik

- Menggunakan unggahan multibagian dengan bucket direktori

Menggunakan unggahan multibagian dengan bucket direktori

Anda dapat menggunakan proses upload multipart untuk mengunggah satu objek sebagai satu set bagian. Setiap bagian merupakan bagian data objek yang saling berkaitan. Anda dapat mengunggah bagian-bagian objek tersebut secara independen dan dengan urutan apa pun. Jika ada

transmisi bagian mana pun yang gagal, Anda dapat mentransmisikan ulang bagian tersebut tanpa memengaruhi bagian lainnya. Setelah semua bagian objek Anda diunggah, Amazon S3 merakit bagian-bagian ini dan menciptakan objek. Secara umum, saat ukuran objek Anda mencapai 100 MB, Anda harus mempertimbangkan untuk menggunakan unggahan multibagian daripada mengunggah objek tersebut dalam satu operasi.

Penggunaan unggahan multibagian memberikan keuntungan sebagai berikut:

- Peningkatan throughput–Anda dapat mengunggah bagian-bagian secara paralel untuk meningkatkan throughput.
- Pemulihan cepat dari masalah jaringan apa pun - Ukuran bagian yang lebih kecil meminimalkan dampak memulai ulang unggahan yang gagal karena kesalahan jaringan.
- Jeda dan pelanjutan pengunggahan objek–Anda dapat mengunggah bagian-bagian objek kapan saja. Setelah Anda memulai unggahan multipart, tidak ada tanggal kedaluwarsa. Anda harus secara eksplisit menyelesaikan atau membatalkan unggahan multipart.
- Mulai unggahan sebelum Anda mengetahui ukuran akhir objek–Anda dapat mengunggah sebuah objek selagi Anda membuatnya.

Kami menyarankan Anda menggunakan unggahan multipart dengan cara berikut:

- Jika Anda mengunggah objek besar melalui jaringan bandwidth tinggi yang stabil, gunakan unggahan multipart untuk memaksimalkan penggunaan bandwidth yang tersedia dengan mengunggah bagian objek secara paralel untuk kinerja multi-threaded.
- Jika Anda mengunggah melalui jaringan jerawatan, gunakan unggahan multibagian untuk meningkatkan ketahanan terhadap kesalahan jaringan dengan menghindari pengunggahan dimulai ulang. Saat menggunakan unggahan multibagian, Anda perlu mencoba mengunggah kembali hanya bagian yang terputus selama pengunggahan. Anda tidak perlu mengunggah ulang objek Anda dari awal.

Saat Anda menggunakan unggahan multibagian untuk mengunggah objek ke kelas penyimpanan Amazon S3 Express One Zone dalam bucket direktori, proses pengunggahan multibagian mirip dengan proses menggunakan unggahan multibagian untuk mengunggah objek ke bucket tujuan umum. Namun, ada beberapa perbedaan penting.

Untuk informasi selengkapnya tentang penggunaan unggahan multibagian untuk mengunggah objek ke S3 Express One Zone, lihat topik berikut.

Topik

- Proses pengunggahan multipart
- Checksum dengan operasi unggahan multibagian
- Operasi pengunggahan multibagian serentak
- Unggahan dan harga multipart
- Operasi dan izin API unggahan multibagian
- Contoh

Proses pengunggahan multipart

Unggahan multipart adalah proses tiga langkah:

- Anda memulai unggahan.
- Anda mengunggah bagian objek.
- Setelah Anda mengunggah semua bagian, Anda menyelesaikan unggahan multibagian.

Setelah menerima permintaan upload multipart lengkap, Amazon S3 membuat objek dari bagian yang diunggah, dan Anda kemudian dapat mengakses objek seperti halnya objek lain di bucket Anda.

Menginisiasi unggahan multibagian

Saat Anda mengirim permintaan untuk memulai unggahan multibagian, Amazon S3 mengirimkan respons dengan ID unggahan, yang merupakan pengidentifikasi unik untuk unggahan multibagian Anda. Anda harus menyertakan ID unggahan ini setiap kali Anda mengunggah bagian, mendaftar bagian, menyelesaikan unggahan, atau membatalkan sebuah unggahan.

Pengunggahan bagian-bagian

Saat mengunggah sebuah bagian, selain ID pengunggahan, Anda harus menentukan nomor bagiannya. Saat Anda menggunakan unggahan multibagian dengan S3 Express One Zone, nomor bagian multipart harus berupa nomor bagian yang berurutan. Jika Anda mencoba menyelesaikan permintaan unggahan multibagian dengan nomor bagian yang tidak berurutan, kesalahan `HTTP 400 Bad Request` (Pesanan Bagian Tidak Valid) akan dihasilkan.

Nomor bagian secara unik mengidentifikasi bagian dan posisinya dalam objek yang Anda unggah. Jika Anda mengunggah bagian baru dengan menggunakan nomor bagian yang sama dengan bagian yang diunggah sebelumnya, bagian yang diunggah sebelumnya akan ditimpa.

Setiap kali Anda mengunggah bagian, Amazon S3 mengembalikan header tag entitas (ETag) dalam responsnya. Untuk setiap unggahan bagian, Anda harus mencatat nomor bagian dan ETag nilainya. ETagNilai untuk semua unggahan bagian objek akan tetap sama, tetapi setiap bagian akan diberi nomor bagian yang berbeda. Anda harus memasukkan nilai-nilai ini dalam permintaan selanjutnya untuk menyelesaikan unggahan multibagian.

Amazon S3 secara otomatis mengenkripsi semua objek baru yang diunggah ke bucket S3. Saat melakukan pengunggahan multibagian, jika Anda tidak menentukan informasi enkripsi dalam permintaan, pengaturan enkripsi bagian yang diunggah diatur ke konfigurasi enkripsi default bucket tujuan. Konfigurasi enkripsi default bucket Amazon S3 selalu diaktifkan dan diatur secara minimum ke enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3). Untuk bucket direktori, SSE-S3 dan enkripsi sisi server dengan AWS KMS kunci (SSE-KMS) didukung. Untuk informasi selengkapnya, lihat Perlindungan data dan enkripsi.

Penyelesaian unggahan multibagian

Saat Anda menyelesaikan unggahan multibagian, Amazon S3 membuat objek dengan menggabungkan bagian-bagian dalam urutan menaik berdasarkan nomor bagian. Setelah permintaan selesai sepenuhnya, bagiantersebut tidak akan ada lagi.

Permintaan unggahan multibagian lengkap Anda harus menyertakan ID unggahan dan daftar nomor bagian dan ETag nilai yang sesuai. Respons Amazon S3 mencakup ETag yang secara unik mengidentifikasi data objek gabungan. ETag Ini bukan MD5 hash dari data objek.

Pendaftaran unggahan multibagian

Anda dapat mendaftar bagian-bagian dari unggahan multibagian tertentu atau semua unggahan multibagian yang sedang berlangsung. Operasi daftar bagian menampilkan informasi bagian yang telah Anda unggah untuk unggahan multibagian tertentu. Untuk setiap permintaan daftar bagian, Amazon S3 akan menampilkan informasi bagian untuk unggahan multibagian tertentu, hingga maksimum 1.000 bagian. Jika ada lebih dari 1.000 bagian dalam unggahan multibagian, Anda harus menggunakan penomoran halaman untuk mengambil semua bagian.

Daftar suku cadang yang dikembalikan tidak termasuk bagian yang belum selesai diunggah. Dengan menggunakan operasi daftar unggahan multibagian, Anda dapat memperoleh daftar unggahan multibagian yang sedang berlangsung.

Unggahan multibagian yang sedang berlangsung adalah unggahan yang telah Anda mulai, tetapi belum selesai atau dibatalkan. Setiap permintaan akan ditampilkan sebanyak maksimum 1.000 unggahan multibagian. Jika ada lebih dari 1.000 unggahan multibagian yang sedang berlangsung, Anda harus mengirim permintaan tambahan untuk mengambil unggahan multibagian yang tersisa. Gunakan pendaftaran yang ditampilkan untuk verifikasi. Jangan menggunakan hasil pendaftaran ini saat mengirim permintaan selesaikan unggahan multibagian. Sebagai gantinya, pertahankan daftar nomor bagian Anda sendiri yang Anda tentukan saat mengunggah bagian dan ETag nilai terkait yang dikembalikan Amazon S3.

Untuk informasi selengkapnya tentang daftar unggahan multibagian, lihat ListPartsdi Referensi API Amazon Simple Storage Service.

Checksum dengan operasi unggahan multibagian

Saat Anda mengunggah objek, Anda dapat menentukan algoritma checksum untuk memeriksa integritas objek. MD5 tidak didukung untuk bucket direktori. Anda dapat menentukan salah satu algoritma pemeriksaan integritas data Secure Hash Algorithms (SHA) atau Cyclic Redundancy Check (CRC) berikut:

- CRC32
- CRC32C
- SHA-1
- SHA-256

Anda dapat menggunakan Amazon S3 REST API atau AWS SDKs untuk mengambil nilai checksum untuk masing-masing bagian dengan menggunakan atau. `GetObject` `HeadObject` Jika Anda ingin mengambil nilai checksum untuk masing-masing bagian dari unggahan multibagian yang masih dalam proses, Anda dapat menggunakan `ListParts`.

> **Important**
>
> Saat menggunakan algoritma checksum sebelumnya, nomor bagian multipart harus menggunakan nomor bagian berurutan. Jika Anda mencoba menyelesaikan permintaan unggahan multibagian dengan nomor bagian yang tidak berurutan, Amazon S3 menghasilkan kesalahan `HTTP 400 Bad Request` (Pesanan Bagian Tidak Valid).

Untuk informasi selengkapnya tentang cara kerja checksum dengan objek unggahan multibagian, lihat. Memeriksa integritas objek di Amazon S3

Operasi pengunggahan multibagian serentak

Dalam lingkungan pengembangan terdistribusi, aplikasi Anda dapat memulai beberapa pembaruan pada objek yang sama secara bersamaan. Misalnya, aplikasi Anda mungkin memulai beberapa unggahan multipart dengan menggunakan kunci objek yang sama. Untuk setiap unggahan ini, aplikasi Anda kemudian dapat mengunggah bagian dan mengirim sebuah permintaan menyelesaikan unggahan ke Amazon S3 untuk membuat objek. Untuk S3 Express One Zone, waktu pembuatan objek adalah tanggal penyelesaian unggahan multibagian.

> **Important**
>
> Pembuatan versi tidak didukung untuk objek yang disimpan dalam bucket direktori.

Unggahan dan harga multipart

Setelah Anda memulai unggahan multibagian, Amazon S3 akan menyimpan semua bagian hingga Anda menyelesaikan atau membatalkan unggahan. Sepanjang masa pakainya, Anda akan ditagih untuk semua penyimpanan, bandwidth, dan permintaan untuk unggahan multibagian ini dan bagian terkaitnya. Jika Anda membatalkan unggahan multibagian, Amazon S3 menghapus artefak unggahan dan bagian apa pun yang telah Anda unggah, dan Anda tidak lagi ditagih untuk itu. Tidak ada biaya penghapusan awal untuk menghapus unggahan multibagian yang tidak lengkap, terlepas dari kelas penyimpanan yang ditentukan. Untuk informasi selengkapnya tentang harga, lihat Harga Amazon S3.

> **Important**
>
> Jika permintaan upload multipart lengkap tidak berhasil dikirim, bagian objek tidak dirakit dan objek tidak dibuat. Anda akan ditagih untuk semua penyimpanan yang terkait dengan bagian yang diunggah. Penting bagi Anda untuk menyelesaikan unggahan multibagian agar objek dibuat atau membatalkan unggahan multibagian untuk menghapus bagian yang diunggah. Sebelum dapat menghapus bucket direktori, Anda harus menyelesaikan atau membatalkan semua unggahan multipart yang sedang berlangsung. Bucket direktori tidak mendukung konfigurasi Siklus Hidup S3. Jika perlu, Anda dapat mencantumkan unggahan multibagian aktif, lalu membatalkan unggahan, lalu menghapus bucket.

Operasi dan izin API unggahan multibagian

Untuk mengizinkan akses ke operasi API manajemen objek pada bucket direktori, Anda memberikan `s3express:CreateSession` izin dalam kebijakan bucket atau kebijakan berbasis identitas AWS Identity and Access Management (IAM).

Anda harus memiliki izin yang diperlukan untuk menggunakan operasi pengunggahan multibagian. Anda dapat menggunakan kebijakan bucket atau kebijakan berbasis identitas IAM untuk memberikan izin kepada prinsipal IAM untuk melakukan operasi ini. Tabel berikut mencantumkan izin yang diperlukan untuk berbagai operasi pengunggahan multibagian.

Anda dapat mengidentifikasi inisiator unggahan multipart melalui elemen. `Initiator` Jika inisiator adalah Akun AWS, elemen ini memberikan informasi yang sama dengan `Owner` elemen. Jika inisiator adalah seorang pengguna IAM, elemen ini akan menyediakan ARN pengguna dan nama tampilan.

Tindakan	Izin yang diperlukan
Membuat unggahan multibagian	Untuk membuat unggahan multipart, Anda harus diizinkan untuk melakukan `s3express:CreateSession` tindakan pada bucket direktori.
Memulai upload multipart	Untuk memulai unggahan multipart, Anda harus diizinkan untuk melakukan `s3express:CreateSession` tindakan pada bucket direktori.
Unggah bagian	Untuk mengunggah bagian, Anda harus diizinkan untuk melakukan `s3express:CreateSession` tindakan pada bucket direktori. Agar inisiator dapat mengunggah bagian, pemilik bucket harus mengizinkan inisiator untuk melakukan `s3express:CreateSession` tindakan pada bucket direktori.
Unggah bagian (salin)	Untuk mengunggah bagian, Anda harus diizinkan untuk melakukan `s3express:CreateSession` tindakan pada bucket direktori. Agar inisiator dapat mengunggah bagian untuk sebuah objek, pemilik bucket harus mengizinkan inisiator untuk melakukan tindakan `s3express:CreateSession` pada objek.

Tindakan	Izin yang diperlukan
Selesaikan unggahan multibagian	Untuk menyelesaikan unggahan multibagian, Anda harus diizinkan untuk melakukan `s3express:CreateSession` tindakan pada bucket direktori. Agar inisiator menyelesaikan unggahan multibagian, pemilik bucket harus mengizinkan inisiator untuk melakukan `s3express:CreateSession` tindakan pada objek.
Pembatalan sebuah unggahan multipart	Untuk membatalkan unggahan multibagian, Anda harus diizinkan untuk melakukan tindakan. `s3express:CreateSession` Agar inisiator membatalkan unggahan multibagian, inisiator harus diberikan akses izin eksplisit untuk melakukan tindakan. `s3express:CreateSession`
Daftar bagian	Untuk mencantumkan bagian-bagian dalam unggahan multibagian, Anda harus diizinkan untuk melakukan `s3express:CreateSession` tindakan pada bucket direktori.
Membuat daftar unggahan multibagian yang sedang berlangsung	Untuk membuat daftar unggahan multipart yang sedang berlangsung ke bucket, Anda harus diizinkan melakukan `s3:ListBucketMultipartUploads` tindakan pada bucket tersebut.

Dukungan operasi API untuk unggahan multipart

Bagian berikut dalam Referensi API Amazon Simple Storage Service menjelaskan operasi Amazon S3 REST API untuk unggahan multipart.

- CreateMultipartUpload
- UploadPart
- UploadPartCopy
- CompleteMultipartUpload
- AbortMultipartUpload
- ListParts

- ListMultipartUploads

Contoh

Untuk menggunakan unggahan multibagian untuk mengunggah objek ke S3 Express One Zone dalam bucket direktori, lihat contoh berikut.

Topik

- Membuat unggahan multibagian
- Mengunggah bagian-bagian dari unggahan multipart
- Menyelesaikan unggahan multibagian
- Membatalkan unggahan multibagian
- Membuat operasi salin unggahan multibagian
- Mencantumkan unggahan multipart yang sedang berlangsung
- Cantumkan bagian-bagian dari unggahan multipart

Membuat unggahan multibagian

> **Note**
>
> Untuk bucket direktori, saat Anda melakukan `CreateMultipartUpload` operasi dan `UploadPartCopy` operasi, enkripsi default bucket harus menggunakan konfigurasi enkripsi yang diinginkan, dan header permintaan yang Anda berikan dalam `CreateMultipartUpload` permintaan harus sesuai dengan konfigurasi enkripsi default bucket tujuan.

Contoh berikut menunjukkan cara membuat unggahan multipart.

Menggunakan AWS SDKs

SDK for Java 2.x

Example

```
/**
 * This method creates a multipart upload request that generates a unique upload ID
 that is used to track
```

```
 * all the upload parts
 *
 * @param s3
 * @param bucketName - for example, 'doc-example-bucket--use1-az4--x-s3'
 * @param key
 * @return
 */
 private static String createMultipartUpload(S3Client s3, String bucketName, String
 key) {

     CreateMultipartUploadRequest createMultipartUploadRequest =
 CreateMultipartUploadRequest.builder()
             .bucket(bucketName)
             .key(key)
             .build();

     String uploadId = null;

     try {
         CreateMultipartUploadResponse response =
 s3.createMultipartUpload(createMultipartUploadRequest);
         uploadId = response.uploadId();
     }
     catch (S3Exception e) {
         System.err.println(e.awsErrorDetails().errorMessage());
         System.exit(1);
     }
     return uploadId;
```

SDK for Python

Example

```
def create_multipart_upload(s3_client, bucket_name, key_name):
    '''
   Create a multipart upload to a directory bucket

   :param s3_client: boto3 S3 client
   :param bucket_name: The destination bucket for the multipart upload
   :param key_name: The key name for the object to be uploaded
   :return: The UploadId for the multipart upload if created successfully, else None
   '''

   try:
```

```
        mpu = s3_client.create_multipart_upload(Bucket = bucket_name, Key =
 key_name)
        return mpu['UploadId']
    except ClientError as e:
        logging.error(e)
        return None
```

Menggunakan AWS CLI

Contoh ini menunjukkan cara membuat unggahan multipart ke bucket direktori dengan menggunakan file. AWS CLI Perintah ini memulai unggahan multipart ke bucket direktori *bucket-base-name* -- *zone-id* --x-s3 untuk objek. *KEY_NAME* Untuk menggunakan perintah ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3api create-multipart-upload --bucket bucket-base-name--zone-id--x-s3 --
key KEY_NAME
```

Untuk informasi selengkapnya, lihat create-multipart-upload di AWS Command Line Interface.

Mengunggah bagian-bagian dari unggahan multipart

Contoh berikut menunjukkan cara mengunggah bagian dari unggahan multipart.

Menggunakan AWS SDKs

SDK for Java 2.x

Contoh berikut menunjukkan cara memecah satu objek menjadi beberapa bagian dan kemudian mengunggah bagian-bagian tersebut ke bucket direktori dengan menggunakan SDK for Java 2.x.

Example

```
/**
 * This method creates part requests and uploads individual parts to S3 and then
 returns all the completed parts
 *
 * @param s3
 * @param bucketName
 * @param key
 * @param uploadId
 * @throws IOException
 */
```

```
private static ListCompletedPartmultipartUpload(S3Client s3, String bucketName,
String key, String uploadId, String filePath) throws IOException {

        int partNumber = 1;
        ListCompletedPart completedParts = new ArrayList<>();
        ByteBuffer bb = ByteBuffer.allocate(1024 * 1024 * 5); // 5 MB byte buffer

        // read the local file, breakdown into chunks and process
        try (RandomAccessFile file = new RandomAccessFile(filePath, "r")) {
            long fileSize = file.length();
            int position = 0;
            while (position < fileSize) {
                file.seek(position);
                int read = file.getChannel().read(bb);

                bb.flip(); // Swap position and limit before reading from the
buffer.
                UploadPartRequest uploadPartRequest = UploadPartRequest.builder()
                        .bucket(bucketName)
                        .key(key)
                        .uploadId(uploadId)
                        .partNumber(partNumber)
                        .build();

                UploadPartResponse partResponse = s3.uploadPart(
                        uploadPartRequest,
                        RequestBody.fromByteBuffer(bb));

                CompletedPart part = CompletedPart.builder()
                        .partNumber(partNumber)
                        .eTag(partResponse.eTag())
                        .build();
                completedParts.add(part);

                bb.clear();
                position += read;
                partNumber++;
            }
        }

        catch (IOException e) {
            throw e;
        }
        return completedParts;
```

```
    }
```

SDK for Python

Contoh berikut menunjukkan cara memecah satu objek menjadi beberapa bagian dan kemudian mengunggah bagian-bagian tersebut ke bucket direktori dengan menggunakan SDK untuk Python.

Example

```
def multipart_upload(s3_client, bucket_name, key_name, mpu_id, part_size):
    '''
    Break up a file into multiple parts and upload those parts to a directory bucket

    :param s3_client: boto3 S3 client
    :param bucket_name: Destination bucket for the multipart upload
    :param key_name: Key name for object to be uploaded and for the local file
 that's being uploaded
    :param mpu_id: The UploadId returned from the create_multipart_upload call
    :param part_size: The size parts that the object will be broken into, in bytes.
                      Minimum 5 MiB, Maximum 5 GiB. There is no minimum size for the
 last part of your multipart upload.
    :return: part_list for the multipart upload if all parts are uploaded
 successfully, else None
    '''

    part_list = []
    try:
        with open(key_name, 'rb') as file:
            part_counter = 1
            while True:
                file_part = file.read(part_size)
                if not len(file_part):
                    break
                upload_part = s3_client.upload_part(
                    Bucket = bucket_name,
                    Key = key_name,
                    UploadId = mpu_id,
                    Body = file_part,
                    PartNumber = part_counter
                )
                part_list.append({'PartNumber': part_counter, 'ETag':
 upload_part['ETag']})
```

```
                part_counter += 1
    except ClientError as e:
        logging.error(e)
        return None
    return part_list
```

Menggunakan AWS CLI

Contoh ini menunjukkan cara memecah satu objek menjadi beberapa bagian dan kemudian mengunggah bagian-bagian tersebut ke bucket direktori dengan menggunakan file AWS CLI. Untuk menggunakan perintah ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3api upload-part --bucket bucket-base-name--zone-id--x-s3 --
key KEY_NAME --part-number 1 --body LOCAL_FILE_NAME --upload-id
 "AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAEMAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAAA0AAAAAAAAAAH2AfYAA
```

Untuk informasi selengkapnya, lihat bagian unggahan di. AWS Command Line Interface

Menyelesaikan unggahan multibagian

Contoh berikut menunjukkan cara menyelesaikan unggahan multipart.

Menggunakan AWS SDKs

SDK for Java 2.x

Contoh berikut menunjukkan cara menyelesaikan upload multipart dengan menggunakan SDK for Java 2.x.

Example

```
/**
 * This method completes the multipart upload request by collating all the upload
 parts
 * @param s3
 * @param bucketName - for example, 'doc-example-bucket--usw2-az1--x-s3'
 * @param key
 * @param uploadId
 * @param uploadParts
 */
 private static void completeMultipartUpload(S3Client s3, String bucketName, String
 key, String uploadId, ListCompletedPart uploadParts) {
```

```
        CompletedMultipartUpload completedMultipartUpload =
CompletedMultipartUpload.builder()
                .parts(uploadParts)
                .build();

        CompleteMultipartUploadRequest completeMultipartUploadRequest =
                CompleteMultipartUploadRequest.builder()
                        .bucket(bucketName)
                        .key(key)
                        .uploadId(uploadId)
                        .multipartUpload(completedMultipartUpload)
                        .build();

        s3.completeMultipartUpload(completeMultipartUploadRequest);
    }

    public static void multipartUploadTest(S3Client s3, String bucketName, String
key, String localFilePath)  {
        System.out.println("Starting multipart upload for: " + key);
        try {
            String uploadId = createMultipartUpload(s3, bucketName, key);
            System.out.println(uploadId);
            ListCompletedPart parts = multipartUpload(s3, bucketName, key, uploadId,
localFilePath);
            completeMultipartUpload(s3, bucketName, key, uploadId, parts);
            System.out.println("Multipart upload completed for: " + key);
        }

        catch (Exception e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

SDK for Python

Contoh berikut menunjukkan cara menyelesaikan upload multipart dengan menggunakan SDK untuk Python.

Example

```
def complete_multipart_upload(s3_client, bucket_name, key_name, mpu_id, part_list):
    '''
    Completes a multipart upload to a directory bucket
```

```
    :param s3_client: boto3 S3 client
    :param bucket_name: The destination bucket for the multipart upload
    :param key_name: The key name for the object to be uploaded
    :param mpu_id: The UploadId returned from the create_multipart_upload call
    :param part_list: The list of uploaded part numbers with their associated ETags
    :return: True if the multipart upload was completed successfully, else False
    '''

    try:
        s3_client.complete_multipart_upload(
            Bucket = bucket_name,
            Key = key_name,
            UploadId = mpu_id,
            MultipartUpload = {
                'Parts': part_list
            }
        )
    except ClientError as e:
        logging.error(e)
        return False
    return True

if __name__ == '__main__':
    MB = 1024 ** 2
    region = 'us-west-2'
    bucket_name = 'BUCKET_NAME'
    key_name = 'OBJECT_NAME'
    part_size = 10 * MB
    s3_client = boto3.client('s3', region_name = region)
    mpu_id = create_multipart_upload(s3_client, bucket_name, key_name)
    if mpu_id is not None:
        part_list = multipart_upload(s3_client, bucket_name, key_name, mpu_id,
 part_size)
        if part_list is not None:
            if complete_multipart_upload(s3_client, bucket_name, key_name, mpu_id,
 part_list):
                print (f'{key_name} successfully uploaded through a ultipart upload
 to {bucket_name}')
            else:
                print (f'Could not upload {key_name} hrough a multipart upload to
 {bucket_name}')
```

Menggunakan AWS CLI

Contoh ini menunjukkan cara menyelesaikan unggahan multipart untuk bucket direktori dengan menggunakan file. AWS CLI Untuk menggunakan perintah ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3api complete-multipart-upload --bucket bucket-base-name--zone-id--x-s3 --
key KEY_NAME --upload-id
 "AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAEMAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAAA0AAAAAAAAAAH2AfYAA
 --multipart-upload file://parts.json
```

Contoh ini mengambil struktur JSON yang menjelaskan bagian-bagian dari upload multipart yang harus dipasang kembali ke dalam file lengkap. Dalam contoh ini, `file://` awalan digunakan untuk memuat struktur JSON dari file di folder lokal bernama. `parts`

parts.json:

```
parts.json
{
  "Parts": [
    {
      "ETag": "6b78c4a64dd641a58dac8d9258b88147",
      "PartNumber": 1
    }
  ]
}
```

Untuk informasi selengkapnya, lihat complete-multipart-upload di AWS Command Line Interface.

Membatalkan unggahan multibagian

Contoh berikut menunjukkan cara membatalkan unggahan multipart.

Menggunakan AWS SDKs

SDK for Java 2.x

Contoh berikut menunjukkan cara membatalkan upload multipart dengan menggunakan SDK for Java 2.x.

Example

```
public static void abortMultiPartUploads( S3Client s3, String bucketName ) {
```

```
        try {
            ListMultipartUploadsRequest listMultipartUploadsRequest =
 ListMultipartUploadsRequest.builder()
                    .bucket(bucketName)
                    .build();

            ListMultipartUploadsResponse response =
 s3.listMultipartUploads(listMultipartUploadsRequest);
            ListMultipartUpload uploads = response.uploads();

            AbortMultipartUploadRequest abortMultipartUploadRequest;
            for (MultipartUpload upload: uploads) {
                abortMultipartUploadRequest = AbortMultipartUploadRequest.builder()
                        .bucket(bucketName)
                        .key(upload.key())
                        .uploadId(upload.uploadId())
                        .build();

                s3.abortMultipartUpload(abortMultipartUploadRequest);
            }

        }

        catch (S3Exception e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

SDK for Python

Contoh berikut menunjukkan cara membatalkan upload multipart dengan menggunakan SDK untuk Python.

Example

```
import logging
import boto3
from botocore.exceptions import ClientError


def abort_multipart_upload(s3_client, bucket_name, key_name, upload_id):
```

```
    '''
    Aborts a partial multipart upload in a directory bucket.

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket where the multipart upload was initiated - for
 example, 'doc-example-bucket--usw2-az1--x-s3'
    :param key_name: Name of the object for which the multipart upload needs to be
 aborted
    :param upload_id: Multipart upload ID for the multipart upload to be aborted
    :return: True if the multipart upload was successfully aborted, False if not
    '''
    try:
        s3_client.abort_multipart_upload(
            Bucket = bucket_name,
            Key = key_name,
            UploadId = upload_id
        )
    except ClientError as e:
        logging.error(e)
        return False
    return True


if __name__ == '__main__':
    region = 'us-west-2'
    bucket_name = 'BUCKET_NAME'
    key_name = 'KEY_NAME'
        upload_id = 'UPLOAD_ID'
    s3_client = boto3.client('s3', region_name = region)
    if abort_multipart_upload(s3_client, bucket_name, key_name, upload_id):
        print (f'Multipart upload for object {key_name} in {bucket_name} bucket has
 been aborted')
    else:
        print (f'Unable to abort multipart upload for object {key_name} in
 {bucket_name} bucket')
```

Menggunakan AWS CLI

Contoh berikut menunjukkan cara membatalkan upload multipart dengan menggunakan file. AWS CLI Untuk menggunakan perintah ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3api abort-multipart-upload --bucket bucket-base-name--zone-id--x-s3 --
key KEY_NAME --upload-id
 "AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAEMAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAAA0AAAAAAAAAAH2AfYAA
MAQAAAAB0OxUFeA7LTbWWFS8WYwhrxDxTIDN-pdEEq_agIHqsbg"
```

Untuk informasi selengkapnya, lihat abort-multipart-upload di AWS Command Line Interface.

Membuat operasi salin unggahan multibagian

> **Note**
>
> - Untuk mengenkripsi salinan bagian objek baru dalam bucket direktori dengan SSE-KMS, Anda harus menentukan SSE-KMS sebagai konfigurasi enkripsi default bucket direktori dengan kunci KMS (khususnya, kunci yang dikelola pelanggan). Kunci yang dikelola AWS(`aws/s3`) tidak didukung. Konfigurasi SSE-KMS Anda hanya dapat mendukung 1 kunci terkelola pelanggan per bucket direktori selama masa pakai bucket. Setelah menentukan kunci terkelola pelanggan untuk SSE-KMS, Anda tidak dapat mengganti kunci terkelola pelanggan untuk konfigurasi SSE-KMS bucket. Anda tidak dapat menentukan setelan enkripsi sisi server untuk salinan bagian objek baru dengan SSE-KMS di header permintaan. UploadPartCopy Selain itu, header permintaan yang Anda berikan dalam `CreateMultipartUpload` permintaan harus sesuai dengan konfigurasi enkripsi default bucket tujuan.
> - Kunci Bucket S3 tidak didukung, saat Anda menyalin objek terenkripsi SSE-KMS dari bucket tujuan umum ke bucket direktori, dari bucket direktori ke bucket tujuan umum, atau di antara bucket direktori, hingga. UploadPartCopy Dalam hal ini, Amazon S3 melakukan panggilan ke AWS KMS setiap kali permintaan salinan dibuat untuk objek yang dienkripsi KMS.

Contoh berikut menunjukkan cara menyalin objek dari satu bucket ke bucket lainnya menggunakan unggahan multipart.

Menggunakan AWS SDKs

SDK for Java 2.x

Contoh berikut menunjukkan cara menggunakan unggahan multipart untuk menyalin objek secara terprogram dari satu bucket ke bucket lainnya dengan menggunakan SDK for Java 2.x.

Example

```
/**
 * This method creates a multipart upload request that generates a unique upload ID
 that is used to track
 * all the upload parts.
 *
 * @param s3
 * @param bucketName
 * @param key
 * @return
 */
 private static String createMultipartUpload(S3Client s3, String bucketName, String
 key) {
        CreateMultipartUploadRequest createMultipartUploadRequest =
 CreateMultipartUploadRequest.builder()
                .bucket(bucketName)
                .key(key)
                .build();
        String uploadId = null;
        try {
            CreateMultipartUploadResponse response =
 s3.createMultipartUpload(createMultipartUploadRequest);
            uploadId = response.uploadId();
        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return uploadId;
  }

  /**
   * Creates copy parts based on source object size and copies over individual parts
   *
   * @param s3
   * @param sourceBucket
   * @param sourceKey
   * @param destnBucket
   * @param destnKey
   * @param uploadId
   * @return
   * @throws IOException
   */
```

```
   public static ListCompletedPart multipartUploadCopy(S3Client s3, String
sourceBucket, String sourceKey, String destnBucket, String destnKey, String
uploadId) throws IOException {

       // Get the object size to track the end of the copy operation.
       HeadObjectRequest headObjectRequest = HeadObjectRequest
               .builder()
               .bucket(sourceBucket)
               .key(sourceKey)
               .build();
       HeadObjectResponse response = s3.headObject(headObjectRequest);
       Long objectSize = response.contentLength();

       System.out.println("Source Object size: " + objectSize);

       // Copy the object using 20 MB parts.
       long partSize = 20 * 1024 * 1024;
       long bytePosition = 0;
       int partNum = 1;
       ListCompletedPart completedParts = new ArrayList<>();
       while (bytePosition < objectSize) {
           // The last part might be smaller than partSize, so check to make sure
           // that lastByte isn't beyond the end of the object.
           long lastByte = Math.min(bytePosition + partSize - 1, objectSize - 1);

           System.out.println("part no: " + partNum + ", bytePosition: " +
bytePosition + ", lastByte: " + lastByte);

           // Copy this part.
           UploadPartCopyRequest req = UploadPartCopyRequest.builder()
                   .uploadId(uploadId)
                   .sourceBucket(sourceBucket)
                   .sourceKey(sourceKey)
                   .destinationBucket(destnBucket)
                   .destinationKey(destnKey)
                   .copySourceRange("bytes="+bytePosition+"-"+lastByte)
                   .partNumber(partNum)
                   .build();
           UploadPartCopyResponse res = s3.uploadPartCopy(req);
           CompletedPart part = CompletedPart.builder()
                   .partNumber(partNum)
                   .eTag(res.copyPartResult().eTag())
                   .build();
           completedParts.add(part);
```

```
            partNum++;
            bytePosition += partSize;
        }
        return completedParts;
    }


    public static void multipartCopyUploadTest(S3Client s3, String srcBucket, String
 srcKey, String destnBucket, String destnKey)  {
        System.out.println("Starting multipart copy for: " + srcKey);
        try {
            String uploadId = createMultipartUpload(s3, destnBucket, destnKey);
            System.out.println(uploadId);
            ListCompletedPart parts = multipartUploadCopy(s3, srcBucket,
 srcKey,destnBucket,  destnKey, uploadId);
            completeMultipartUpload(s3, destnBucket, destnKey, uploadId, parts);
            System.out.println("Multipart copy completed for: " + srcKey);
        } catch (Exception e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

SDK for Python

Contoh berikut menunjukkan cara menggunakan unggahan multipart untuk menyalin objek secara terprogram dari satu bucket ke bucket lainnya dengan menggunakan SDK untuk Python.

Example

```
import logging
import boto3
from botocore.exceptions import ClientError

def head_object(s3_client, bucket_name, key_name):
    '''
    Returns metadata for an object in a directory bucket

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket that contains the object to query for metadata
    :param key_name: Key name to query for metadata
    :return: Metadata for the specified object if successful, else None
    '''
```

```
    try:
        response = s3_client.head_object(
            Bucket = bucket_name,
            Key = key_name
        )
        return response
    except ClientError as e:
        logging.error(e)
        return None

def create_multipart_upload(s3_client, bucket_name, key_name):
    '''
    Create a multipart upload to a directory bucket

    :param s3_client: boto3 S3 client
    :param bucket_name: Destination bucket for the multipart upload
    :param key_name: Key name of the object to be uploaded
    :return: UploadId for the multipart upload if created successfully, else None
    '''

    try:
        mpu = s3_client.create_multipart_upload(Bucket = bucket_name, Key =
 key_name)
        return mpu['UploadId']
    except ClientError as e:
        logging.error(e)
        return None

def multipart_copy_upload(s3_client, source_bucket_name, key_name,
 target_bucket_name, mpu_id, part_size):
    '''
    Copy an object in a directory bucket to another bucket in multiple parts of a
 specified size

    :param s3_client: boto3 S3 client
    :param source_bucket_name: Bucket where the source object exists
    :param key_name: Key name of the object to be copied
    :param target_bucket_name: Destination bucket for copied object
    :param mpu_id: The UploadId returned from the create_multipart_upload call
    :param part_size: The size parts that the object will be broken into, in bytes.
                      Minimum 5 MiB, Maximum 5 GiB. There is no minimum size for the
 last part of your multipart upload.
    :return: part_list for the multipart copy if all parts are copied successfully,
 else None
```

```
    '''

    part_list = []
    copy_source = {
        'Bucket': source_bucket_name,
        'Key': key_name
    }
    try:
        part_counter = 1
        object_size = head_object(s3_client, source_bucket_name, key_name)
        if object_size is not None:
            object_size = object_size['ContentLength']
        while (part_counter - 1) * part_size <object_size:
            bytes_start = (part_counter - 1) * part_size
            bytes_end = (part_counter * part_size) - 1
            upload_copy_part = s3_client.upload_part_copy (
                Bucket = target_bucket_name,
                CopySource = copy_source,
                CopySourceRange = f'bytes={bytes_start}-{bytes_end}',
                Key = key_name,
                PartNumber = part_counter,
                UploadId = mpu_id
            )
            part_list.append({'PartNumber': part_counter, 'ETag':
 upload_copy_part['CopyPartResult']['ETag']})
            part_counter += 1
    except ClientError as e:
        logging.error(e)
        return None
    return part_list

def complete_multipart_upload(s3_client, bucket_name, key_name, mpu_id, part_list):
    '''
    Completes a multipart upload to a directory bucket

    :param s3_client: boto3 S3 client
    :param bucket_name: Destination bucket for the multipart upload
    :param key_name: Key name of the object to be uploaded
    :param mpu_id: The UploadId returned from the create_multipart_upload call
    :param part_list: List of uploaded part numbers with associated ETags
    :return: True if the multipart upload was completed successfully, else False
    '''

    try:
```

```
        s3_client.complete_multipart_upload(
            Bucket = bucket_name,
            Key = key_name,
            UploadId = mpu_id,
            MultipartUpload = {
                'Parts': part_list
            }
        )
    except ClientError as e:
        logging.error(e)
        return False
    return True

if __name__ == '__main__':
    MB = 1024 ** 2
    region = 'us-west-2'
    source_bucket_name = 'SOURCE_BUCKET_NAME'
    target_bucket_name = 'TARGET_BUCKET_NAME'
    key_name = 'KEY_NAME'
    part_size = 10 * MB
    s3_client = boto3.client('s3', region_name = region)
    mpu_id = create_multipart_upload(s3_client, target_bucket_name, key_name)
    if mpu_id is not None:
        part_list = multipart_copy_upload(s3_client, source_bucket_name, key_name,
 target_bucket_name, mpu_id, part_size)
        if part_list is not None:
            if complete_multipart_upload(s3_client, target_bucket_name, key_name,
 mpu_id, part_list):
                print (f'{key_name} successfully copied through multipart copy from
 {source_bucket_name} to {target_bucket_name}')
            else:
                print (f'Could not copy {key_name} through multipart copy from
 {source_bucket_name} to {target_bucket_name}')
```

Menggunakan AWS CLI

Contoh berikut menunjukkan cara menggunakan unggahan multibagian untuk menyalin objek secara terprogram dari satu bucket ke bucket direktori menggunakan file. AWS CLI Untuk menggunakan perintah ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3api upload-part-copy --bucket bucket-base-name--zone-id--x-s3 --
key TARGET_KEY_NAME --copy-source SOURCE_BUCKET_NAME/SOURCE_KEY_NAME --part-number 1 --
```

```
upload-id
 "AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAEMAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAAA0AAAAAAAAAAH2AfYAA
```

Untuk informasi selengkapnya, lihat upload-part-copy di AWS Command Line Interface.

Mencantumkan unggahan multipart yang sedang berlangsung

Untuk mencantumkan unggahan multibagian yang sedang berlangsung ke bucket direktori, Anda dapat menggunakan file AWS SDKs, atau file. AWS CLI

Menggunakan AWS SDKs

SDK for Java 2.x

Contoh berikut menunjukkan cara membuat daftar unggahan multipart yang sedang berlangsung (tidak lengkap) dengan menggunakan SDK for Java 2.x.

Example

```
public static void listMultiPartUploads( S3Client s3, String bucketName) {
       try {
           ListMultipartUploadsRequest listMultipartUploadsRequest =
ListMultipartUploadsRequest.builder()
               .bucket(bucketName)
               .build();

           ListMultipartUploadsResponse response =
s3.listMultipartUploads(listMultipartUploadsRequest);
           List MultipartUpload uploads = response.uploads();
           for (MultipartUpload upload: uploads) {
               System.out.println("Upload in progress: Key = \"" + upload.key() +
"\", id = " + upload.uploadId());
           }
     }
     catch (S3Exception e) {
           System.err.println(e.getMessage());
           System.exit(1);
     }
 }
```

SDK for Python

Contoh berikut menunjukkan cara membuat daftar unggahan multipart yang sedang berlangsung (tidak lengkap) dengan menggunakan SDK untuk Python.

Example

```
import logging
import boto3
from botocore.exceptions import ClientError

def list_multipart_uploads(s3_client, bucket_name):
    '''
    List any incomplete multipart uploads in a directory bucket in e specified gion

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket to check for incomplete multipart uploads
    :return: List of incomplete multipart uploads if there are any, None if not
    '''

    try:
        response = s3_client.list_multipart_uploads(Bucket = bucket_name)
        if 'Uploads' in response.keys():
            return response['Uploads']
        else:
            return None
    except ClientError as e:
        logging.error(e)

if __name__ == '__main__':
    bucket_name = 'BUCKET_NAME'
    region = 'us-west-2'
    s3_client = boto3.client('s3', region_name = region)
    multipart_uploads = list_multipart_uploads(s3_client, bucket_name)
    if multipart_uploads is not None:
        print (f'There are {len(multipart_uploads)} ncomplete multipart uploads for
 {bucket_name}')
    else:
        print (f'There are no incomplete multipart uploads for {bucket_name}')
```

Menggunakan AWS CLI

Contoh berikut menunjukkan cara membuat daftar unggahan multipart yang sedang berlangsung (tidak lengkap) dengan menggunakan. AWS CLI Untuk menggunakan perintah ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3api list-multipart-uploads --bucket bucket-base-name--zone-id--x-s3
```

Untuk informasi selengkapnya, lihat list-multipart-uploads di AWS Command Line Interface.

Cantumkan bagian-bagian dari unggahan multipart

Contoh berikut menunjukkan cara membuat daftar bagian dari unggahan multipart ke bucket direktori.

Menggunakan AWS SDKs

SDK for Java 2.x

Contoh berikut menunjukkan cara membuat daftar bagian dari unggahan multipart ke bucket direktori dengan menggunakan SDK for Java 2.x.

```
public static void listMultiPartUploadsParts( S3Client s3, String bucketName, String
 objKey, String uploadID) {

        try {
            ListPartsRequest listPartsRequest = ListPartsRequest.builder()
                .bucket(bucketName)
                .uploadId(uploadID)
                .key(objKey)
                .build();

            ListPartsResponse response = s3.listParts(listPartsRequest);
            ListPart parts = response.parts();
            for (Part part: parts) {
                System.out.println("Upload in progress: Part number = \"" +
 part.partNumber() + "\", etag = " + part.eTag());
            }

        }

        catch (S3Exception e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }


    }
```

SDK for Python

Contoh berikut menunjukkan cara membuat daftar bagian dari unggahan multipart ke bucket direktori dengan menggunakan SDK untuk Python.

```
import logging
import boto3
from botocore.exceptions import ClientError

def list_parts(s3_client, bucket_name, key_name, upload_id):
    '''
    Lists the parts that have been uploaded for a specific multipart upload to a
 directory bucket.

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket that multipart uploads parts have been uploaded to
    :param key_name: Name of the object that has parts uploaded
    :param upload_id: Multipart upload ID that the parts are associated with
    :return: List of parts associated with the specified multipart upload, None if
 there are no parts
    '''
    parts_list = []
    next_part_marker = ''
    continuation_flag = True
    try:
        while continuation_flag:
            if next_part_marker == '':
                response = s3_client.list_parts(
                    Bucket = bucket_name,
                    Key = key_name,
                    UploadId = upload_id
                )
            else:
                response = s3_client.list_parts(
                    Bucket = bucket_name,
                    Key = key_name,
                    UploadId = upload_id,
                    NextPartMarker = next_part_marker
                )
            if 'Parts' in response:
                for part in response['Parts']:
                    parts_list.append(part)
                if response['IsTruncated']:
                    next_part_marker = response['NextPartNumberMarker']
```

```
                else:
                    continuation_flag = False
            else:
                continuation_flag = False
        return parts_list
    except ClientError as e:
        logging.error(e)
        return None

if __name__ == '__main__':
    region = 'us-west-2'
    bucket_name = 'BUCKET_NAME'
    key_name = 'KEY_NAME'
    upload_id = 'UPLOAD_ID'
    s3_client = boto3.client('s3', region_name = region)
    parts_list = list_parts(s3_client, bucket_name, key_name, upload_id)
    if parts_list is not None:
        print (f'{key_name} has {len(parts_list)} parts uploaded to {bucket_name}')
    else:
        print (f'There are no multipart uploads with that upload ID for
 {bucket_name} bucket')
```

Menggunakan AWS CLI

Contoh berikut menunjukkan cara membuat daftar bagian dari unggahan multipart ke bucket direktori dengan menggunakan. AWS CLI Untuk menggunakan perintah ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3api list-parts --bucket bucket-base-name--zone-id--x-s3 --key KEY_NAME --upload-
id
 "AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAEMAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAAA0AAAAAAAAAAH2AfYAA
```

Untuk informasi selengkapnya, lihat bagian daftar di. AWS Command Line Interface

Menyalin objek dari atau ke ember direktori

Operasi penyalinan membuat salinan objek yang sudah disimpan di Amazon S3. Anda dapat menyalin objek antara bucket direktori dan bucket tujuan umum. Anda juga dapat menyalin objek di dalam bucket dan di bucket dengan tipe yang sama, misalnya, dari bucket direktori ke bucket direktori.

> **Note**
>
> Menyalin objek di berbagai objek Wilayah AWS tidak didukung saat bucket sumber atau tujuan berada di Zona AWS Lokal. Bucket sumber dan tujuan harus memiliki induk Wilayah AWS yang sama. Bucket sumber dan tujuan dapat berupa jenis lokasi bucket yang berbeda (Availability Zone atau Local Zone).

Anda dapat membuat salinan objek hingga 5 GB dalam satu operasi atom. Namun, untuk menyalin objek yang lebih besar dari 5 GB, Anda harus menggunakan operasi API unggahan multibagian. Untuk informasi selengkapnya, lihat Menggunakan unggahan multibagian dengan bucket direktori.

Izin

Untuk menyalin objek, Anda harus memiliki izin berikut:

- Untuk menyalin objek dari satu bucket direktori ke bucket direktori lain, Anda harus memiliki izin `s3express:CreateSession`.
- Untuk menyalin objek dari bucket direktori ke bucket tujuan umum, Anda harus memiliki izin `s3express:CreateSession` dan izin `s3:PutObject`untuk menulis salinan objek ke bucket tujuan.
- Untuk menyalin objek dari bucket tujuan umum ke bucket direktori, Anda harus memiliki `s3express:CreateSession` izin dan `s3:GetObject` izin untuk membaca objek sumber yang sedang disalin.

  Untuk informasi selengkapnya, lihat CopyObject dalam Referensi API Amazon Simple Storage Service.

Enkripsi

Amazon S3 secara otomatis mengenkripsi semua objek baru yang diunggah ke bucket S3. Konfigurasi enkripsi default bucket S3 selalu diaktifkan dan diatur secara minimum ke enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3).

Untuk bucket direktori, SSE-S3 dan enkripsi sisi server dengan kunci AWS Key Management Service (AWS KMS) (SSE-KMS) didukung. Jika bucket tujuan adalah bucket direktori, sebaiknya enkripsi default bucket tujuan menggunakan konfigurasi enkripsi yang diinginkan dan Anda tidak mengganti enkripsi default bucket. Kemudian, objek baru secara otomatis dienkripsi dengan pengaturan enkripsi

yang diinginkan. Selain itu, Kunci Bucket S3 tidak didukung, saat Anda menyalin objek terenkripsi SSE-KMS dari bucket tujuan umum ke bucket direktori, dari bucket direktori ke bucket tujuan umum, atau di antara bucket direktori, hingga. CopyObject Dalam hal ini, Amazon S3 melakukan panggilan ke AWS KMS setiap kali permintaan salinan dibuat untuk objek yang dienkripsi KMS. Untuk informasi selengkapnya tentang perilaku pengesampingan enkripsi dalam bucket direktori, lihat Menentukan enkripsi sisi server dengan unggahan objek baru. AWS KMS

Untuk bucket tujuan umum, Anda dapat menggunakan SSE-S3 (default), enkripsi sisi server dengan () kunci (SSE-KMS), enkripsi sisi server dua lapis dengan AWS Key Management Service kunci (DSSE-KMS), atau enkripsi sisi server dengan AWS KMS kunci yang disediakan pelanggan (SSE-C).AWS KMS

Jika Anda membuat permintaan salinan yang menetapkan untuk menggunakan DSSE-KMS atau SSE-C untuk bucket direktori (baik bucket sumber atau tujuan), respons akan menampilkan kesalahan.

Tanda

Bucket direktori tidak mendukung tanda. Jika Anda menyalin objek yang memiliki tag dari bucket tujuan umum ke bucket direktori, Anda akan menerima `501 (Not Implemented)` respons HTTP. Untuk informasi selengkapnya, lihat CopyObject dalam Referensi API Amazon Simple Storage Service.

ETags

Tag entitas (ETags) untuk S3 Express One Zone adalah string alfanumerik acak dan bukan checksum. MD5 Untuk membantu memastikan integritas objek, gunakan checksum tambahan.

Checksum tambahan

S3 Express One Zone menawarkan kepada Anda opsi untuk memilih algoritma checksum yang digunakan untuk memvalidasi data Anda selama mengunggah atau mengunduh. Anda dapat memilih salah satu algoritma pemeriksaan integritas data Secure Hash Algorithms (SHA) atau Cyclic Redundancy Check (CRC) berikut:, C, SHA-1, dan SHA-256. CRC32 CRC32 MD5checksum berbasis tidak didukung dengan kelas penyimpanan S3 Express One Zone.

Untuk informasi selengkapnya, lihat Praktik terbaik checksum tambahan S3.

Fitur yang didukung

Untuk informasi selengkapnya tentang fitur Amazon S3 mana yang didukung untuk S3 Express One Zone, lihat. Perbedaan untuk ember direktori

Menggunakan konsol S3 (salin ke bucket direktori)

> **Note**
>
> Batasan dan batasan saat Anda menyalin objek ke bucket direktori dengan konsol adalah sebagai berikut:
>
> - `Copy`Tindakan ini berlaku untuk semua objek dalam folder yang ditentukan (awalan). Objek yang ditambahkan ke folder ini saat tindakan sedang berlangsung mungkin terpengaruh.
> - Objek yang dienkripsi dengan kunci enkripsi yang disediakan pelanggan (SSE-C) tidak dapat disalin dengan menggunakan konsol S3. Untuk menyalin objek yang dienkripsi dengan SSE-C, gunakan, AWS SDK AWS CLI, atau Amazon S3 REST API.
> - Objek yang disalin tidak akan mempertahankan pengaturan Object Lock dari objek asli.
> - Jika bucket tempat Anda menyalin objek menggunakan pengaturan yang diberlakukan pemilik bucket untuk Kepemilikan Objek S3, objek tidak ACLs akan disalin ke tujuan yang ditentukan.
> - Jika Anda ingin menyalin objek ke bucket yang menggunakan setelan diberlakukan pemilik bucket untuk Kepemilikan Objek S3, pastikan bucket sumber juga menggunakan setelan yang diberlakukan pemilik bucket, atau hapus semua hibah ACL objek ke akun dan grup lain. AWS
> - Objek yang disalin dari bucket tujuan umum ke bucket direktori tidak akan menyimpan tag objek ACLs, atau nilai Etag. Nilai checksum dapat disalin, tetapi tidak setara dengan Etag. Nilai checksum dapat berubah dibandingkan dengan saat ditambahkan.
> - Semua objek yang disalin ke bucket direktori akan dengan pengaturan yang diberlakukan pemilik bucket untuk Kepemilikan Objek S3.

Untuk menyalin objek dari bucket tujuan umum atau bucket direktori ke bucket direktori

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, jenis bucket yang ingin Anda salin objek:
   - Untuk menyalin dari ember tujuan umum, pilih tab Bucket tujuan umum.
   - Untuk menyalin dari bucket direktori, pilih tab Directory buckets.
3. Pilih bucket tujuan umum atau bucket direktori yang berisi objek yang ingin Anda salin.

4. Pilih tab Objek. Pada halaman Objek, pilih kotak centang di sebelah kiri nama objek yang ingin Anda salin.
5. Pada menu Tindakan, pilih Salin.

   Halaman Salin muncul.
6. Di bawah Tujuan, pilih ember Direktori untuk jenis tujuan Anda. Untuk menentukan jalur tujuan, pilih Browse S3, navigasikan ke tujuan, lalu pilih tombol opsi di sebelah kiri tujuan. Pilih Pilih tujuan di sudut kanan bawah.

   Atau, masukkan jalur tujuan.
7. Di bawah Pengaturan salinan tambahan, pilih apakah Anda ingin Menyalin setelan sumber, Jangan tentukan pengaturan, atau Tentukan pengaturan. Pengaturan sumber salin adalah opsi default. Jika Anda hanya ingin menyalin objek tanpa atribut pengaturan sumber, pilih Jangan tentukan pengaturan. Pilih Tentukan pengaturan untuk menentukan pengaturan enkripsi sisi server, checksum, dan metadata.
8. Pilih Salin di sudut kanan bawah. Amazon S3 menyalin objek Anda ke tujuan.

Menggunakan konsol S3 (salin ke bucket tujuan umum)

> **Note**
>
> Batasan dan batasan saat Anda menyalin objek ke bucket tujuan umum dengan konsol adalah sebagai berikut:
>
> - `Copy`Tindakan ini berlaku untuk semua objek dalam folder yang ditentukan (awalan). Objek yang ditambahkan ke folder ini saat tindakan sedang berlangsung mungkin terpengaruh.
> - Objek yang dienkripsi dengan kunci enkripsi yang disediakan pelanggan (SSE-C) tidak dapat disalin dengan menggunakan konsol S3. Untuk menyalin objek yang dienkripsi dengan SSE-C, gunakan, AWS SDK AWS CLI, atau Amazon S3 REST API.
> - Objek yang disalin tidak akan mempertahankan pengaturan Object Lock dari objek asli.
> - Jika bucket tempat Anda menyalin objek menggunakan pengaturan yang diberlakukan pemilik bucket untuk Kepemilikan Objek S3, objek tidak ACLs akan disalin ke tujuan yang ditentukan.
> - Jika Anda ingin menyalin objek ke bucket yang menggunakan setelan diberlakukan pemilik bucket untuk Kepemilikan Objek S3, pastikan bucket sumber juga menggunakan setelan

yang diberlakukan pemilik bucket, atau hapus semua hibah ACL objek ke akun dan grup lain. AWS

Untuk menyalin objek dari bucket direktori ke bucket tujuan umum

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Pilih tab Ember direktori.
4. Pilih bucket direktori yang berisi objek yang ingin Anda salin.
5. Pilih tab Objek. Pada halaman Objek, pilih kotak centang di sebelah kiri nama objek yang ingin Anda salin.
6. Pada menu Tindakan, pilih Salin.
7. Di bawah Tujuan, pilih Bucket tujuan umum untuk jenis tujuan Anda. Untuk menentukan jalur tujuan, pilih Browse S3, navigasikan ke tujuan, dan pilih tombol opsi di sebelah kiri tujuan. Pilih Pilih tujuan di sudut kanan bawah.

   Atau, masukkan jalur tujuan.
8. Di bawah Pengaturan salinan tambahan, pilih apakah Anda ingin Menyalin setelan sumber, Jangan tentukan pengaturan, atau Tentukan pengaturan. Pengaturan sumber salin adalah opsi default. Jika Anda hanya ingin menyalin objek tanpa atribut pengaturan sumber, pilih Jangan tentukan pengaturan. Pilih Tentukan pengaturan untuk menentukan pengaturan untuk kelas penyimpanan, tag objek ACLs, metadata, enkripsi sisi server, dan checksum tambahan.
9. Pilih Salin di sudut kanan bawah. Amazon S3 menyalin objek Anda ke tujuan.

Menggunakan AWS SDKs

SDK for Java 2.x

Example

```
public static void copyBucketObject (S3Client s3, String sourceBucket, String
objectKey, String targetBucket) {
     CopyObjectRequest copyReq = CopyObjectRequest.builder()
         .sourceBucket(sourceBucket)
```

```
            .sourceKey(objectKey)
            .destinationBucket(targetBucket)
            .destinationKey(objectKey)
            .build();
        String temp = "";

        try {
            CopyObjectResponse copyRes = s3.copyObject(copyReq);
            System.out.println("Successfully copied " + objectKey +" from bucket " +
sourceBucket +" into bucket "+targetBucket);
        }

        catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
}
```

Menggunakan AWS CLI

`copy-object`Contoh perintah berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk menyalin objek dari satu bucket ke bucket lain. Anda dapat menyalin objek di antara jenis bucket. Untuk menjalankan perintah ini, ganti placeholder input pengguna dengan informasi Anda sendiri.

```
aws s3api copy-object --copy-source SOURCE_BUCKET/SOURCE_KEY_NAME --key TARGET_KEY_NAME
 --bucket TARGET_BUCKET_NAME
```

Untuk informasi selengkapnya, lihat copy-object dalam AWS CLI Referensi Perintah.

Menghapus objek dari ember direktori

Anda dapat menghapus objek dari bucket direktori Amazon S3 dengan menggunakan konsol Amazon S3 AWS Command Line Interface ,AWS CLI(), atau. AWS SDKs Untuk informasi selengkapnya, lihat Bekerja dengan bucket direktori dan S3 Express One Zone.

> ⚠ Warning
>
> - Menghapus objek tidak dapat dibatalkan.

- Tindakan ini menghapus semua objek yang telah ditentukan. Saat menghapus folder, tunggu hingga tindakan penghapusannya selesai sebelum menambahkan objek baru ke folder tersebut. Jika tidak, objek baru mungkin juga terhapus.

Note

Saat Anda menghapus beberapa objek secara terprogram dari bucket direktori, perhatikan hal berikut:

- Kunci objek dalam permintaan `DeleteObjects` harus berisi setidaknya satu karakter spasi non-putih. String dari semua karakter spasi putih tidak didukung.
- Kunci objek dalam `DeleteObjects` permintaan tidak dapat berisi karakter kontrol Unicode, kecuali untuk baris baru (`\n`), tab (`\t`), dan carriage return (`\r`).

Menggunakan konsol S3

Untuk menghapus objek

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket direktori.
3. Pilih bucket direktori yang berisi objek yang ingin Anda hapus.
4. Pilih tab Objek. Dalam daftar Objek, pilih kotak centang di sebelah kiri objek atau objek yang ingin Anda hapus.
5. Pilih Hapus.
6. Pada halaman Hapus objek, masukkan **`permanently delete`** di kotak teks.
7. Pilih Hapus objek.

Menggunakan AWS SDKs

SDK for Java 2.x

Example

Contoh berikut menghapus objek dalam bucket direktori dengan menggunakan file. AWS SDK for Java 2.x

```
static void deleteObject(S3Client s3Client, String bucketName, String objectKey) {



        try {

            DeleteObjectRequest del = DeleteObjectRequest.builder()
                    .bucket(bucketName)
                    .key(objectKey)
                    .build();

            s3Client.deleteObject(del);

            System.out.println("Object " + objectKey + " has been deleted");


        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }

    }
```

SDK for Python

Example

Contoh berikut menghapus objek dalam bucket direktori dengan menggunakan file. AWS SDK untuk Python (Boto3)

```

import logging
import boto3
from botocore.exceptions import ClientError
```

```
def delete_objects(s3_client, bucket_name, objects):
    '''
    Delete a list of objects in a directory bucket

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket that contains objects to be deleted; for example,
 'doc-example-bucket--usw2-az1--x-s3'
    :param objects: List of dictionaries that specify the key names to delete
    :return: Response output, else False
    '''

    try:
        response = s3_client.delete_objects(
            Bucket = bucket_name,
            Delete = {
                'Objects': objects
            }
        )
        return response
    except ClientError as e:
        logging.error(e)
        return False


if __name__ == '__main__':
    region = 'us-west-2'
    bucket_name = 'BUCKET_NAME'
    objects = [
        {
            'Key': '0.txt'
        },
        {
            'Key': '1.txt'
        },
        {
            'Key': '2.txt'
        },
        {
            'Key': '3.txt'
        },
        {
            'Key': '4.txt'
        }
    ]
```

```
    s3_client = boto3.client('s3', region_name = region)
    results = delete_objects(s3_client, bucket_name, objects)
    if results is not None:
        if 'Deleted' in results:
            print (f'Deleted {len(results["Deleted"])} objects from {bucket_name}')
        if 'Errors' in results:
            print (f'Failed to delete {len(results["Errors"])} objects from
{bucket_name}')
```

Menggunakan AWS CLI

`delete-object`Contoh perintah berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk menghapus objek dari bucket direktori. Untuk menjalankan perintah ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3api delete-object --bucket bucket-base-name--zone-id--x-s3 --key KEY_NAME
```

Untuk informasi selengkapnya, lihat delete-object dalam AWS CLI Referensi Perintah.

`delete-objects`Contoh perintah berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk menghapus objek dari bucket direktori. Untuk menjalankan perintah ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

`delete.json`File tersebut adalah sebagai berikut:

```
{
    "Objects": [
        {
            "Key": "0.txt"
        },
        {
            "Key": "1.txt"
        },
        {
            "Key": "2.txt"
        },
        {
            "Key": "3.txt"
        }
```

```
    ]
}
```

`delete-objects`Contoh perintah adalah sebagai berikut:

```
aws s3api delete-objects --bucket bucket-base-name--zone-id--x-s3 --delete
 file://delete.json
```

Untuk informasi selengkapnya, lihat delete-objects dalam AWS CLI Referensi Perintah.

Mengunduh objek dari ember direktori

Contoh kode berikut menunjukkan cara membaca data dari (mengunduh) objek di bucket direktori Amazon S3 dengan menggunakan operasi `GetObject` API.

Menggunakan AWS SDKs

SDK for Java 2.x

Example

Contoh kode berikut menunjukkan cara membaca data dari objek dalam ember direktori dengan menggunakan AWS SDK for Java 2.x.

```
public static void getObject(S3Client s3Client, String bucketName, String objectKey)
 {
     try {
         GetObjectRequest objectRequest = GetObjectRequest
            .builder()
            .key(objectKey)
            .bucket(bucketName)
            .build();

         ResponseBytes GetObjectResponse objectBytes =
 s3Client.getObjectAsBytes(objectRequest);
         byte[] data = objectBytes.asByteArray();

         //Print object contents to console
         String s = new String(data, StandardCharsets.UTF_8);
         System.out.println(s);
    }
```

```
    catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

SDK for Python

Example

Contoh kode berikut menunjukkan cara membaca data dari objek dalam ember direktori dengan menggunakan AWS SDK untuk Python (Boto3).

```
import boto3
from botocore.exceptions import ClientError
from botocore.response import StreamingBody

def get_object(s3_client: boto3.client, bucket_name: str, key_name: str) ->
 StreamingBody:
    """
    Gets the object.
    :param s3_client:
    :param bucket_name: The bucket that contains the object.
    :param key_name: The key of the object to be downloaded.
    :return: The object data in bytes.
    """
    try:
        response = s3_client.get_object(Bucket=bucket_name, Key=key_name)
        body = response['Body'].read()
        print(f"Got object '{key_name}' from bucket '{bucket_name}'.")
    except ClientError:
        print(f"Couldn't get object '{key_name}' from bucket '{bucket_name}'.")
        raise
    else:
        return body

def main():
    s3_client = boto3.client('s3')
    resp = get_object(s3_client, 'doc-example-bucket--use1-az4--x-s3', 'sample.txt')
    print(resp)

if __name__ == "__main__":
```

```
    main()
```

Menggunakan AWS CLI

Contoh perintah `get-object` berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk mengunduh objek dari Amazon S3. Perintah ini mendapatkan objek *KEY_NAME* dari bucket direktori*bucket-base-name*`--`*zone-id*`--x-s3`. Objek akan diunduh ke file yang bernama *LOCAL_FILE_NAME*. Untuk menjalankan perintah ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3api get-object --bucket bucket-base-name--zone-id--x-s3 --
key KEY_NAME LOCAL_FILE_NAME
```

Untuk informasi selengkapnya, lihat get-object dalam AWS CLI Referensi Perintah.

Menghasilkan keranjang direktori objek URLs yang telah ditentukan sebelumnya untuk berbagi objek

Contoh kode berikut menunjukkan cara membuat presigned URLs untuk berbagi objek dari bucket direktori Amazon S3.

Menggunakan AWS CLI

Contoh perintah berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk menghasilkan URL presigned untuk objek dari Amazon S3. Perintah ini menghasilkan URL presigned untuk objek *KEY_NAME* dari bucket *bucket-base-name*`--`*zone-id*`--x-s3` direktori. Untuk menjalankan perintah ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3 presign s3://bucket-base-name--zone-id--x-s3/KEY_NAME --expires-in 7200
```

Untuk informasi selengkapnya, lihat presign dalam AWS CLI Referensi Perintah.

Mengambil metadata objek dari bucket direktori

Contoh AWS SDK dan AWS CLI berikut menunjukkan cara menggunakan `HeadObject` operasi API `GetObjectAttributes` dan untuk mengambil metadata dari objek di bucket direktori Amazon S3 tanpa mengembalikan objek itu sendiri.

Menggunakan AWS SDKs

SDK for Java 2.x

Example

```
public static void headObject(S3Client s3Client, String bucketName, String
 objectKey) {
    try {
        HeadObjectRequest headObjectRequest = HeadObjectRequest
                .builder()
                .bucket(bucketName)
                .key(objectKey)
                .build();
        HeadObjectResponse response = s3Client.headObject(headObjectRequest);
        System.out.format("Amazon S3 object: \"%s\" found in bucket: \"%s\" with
 ETag: \"%s\"", objectKey, bucketName, response.eTag());
    }
    catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
```

Menggunakan AWS CLI

`head-object`Contoh perintah berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk mengambil metadata dari objek. Untuk menjalankan perintah ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3api head-object --bucket bucket-base-name--zone-id--x-s3 --key KEY_NAME
```

Untuk informasi selengkapnya, lihat head-object dalam AWS CLI Referensi Perintah.

`get-object-attributes`Contoh perintah berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk mengambil metadata dari objek. Untuk menjalankan perintah ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3api get-object-attributes --bucket bucket-base-name--zone-id--x-s3 --key KEY_NAME
 --object-attributes "StorageClass" "ETag" "ObjectSize"
```

Untuk informasi selengkapnya, lihat get-object-attributes dalam AWS CLI Referensi Perintah.

Daftar objek dari ember direktori

Contoh kode berikut menunjukkan cara mencantumkan objek dalam bucket direktori Amazon S3 dengan menggunakan operasi `ListObjectsV2` API.

Menggunakan AWS CLI

`list-objects-v2`Contoh perintah berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk daftar objek dari Amazon S3. Perintah ini mencantumkan objek dari bucket direktori*`bucket-base-name`*`--`*`zone-id`*`--x-s3`. Untuk menjalankan perintah ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3api list-objects-v2 --bucket bucket-base-name--zone-id--x-s3
```

Untuk informasi selengkapnya, lihat list-objects-v2 dalam AWS CLI Referensi Perintah.

Keamanan untuk ember direktori

Keamanan cloud di AWS adalah prioritas tertinggi. 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 AWS dan Anda. 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 melakukan pengujian dan verifikasi secara berkala terhadap efektivitas keamanan kami sebagai bagian dari AWS Compliance Programs.

  Untuk mempelajari tentang program kepatuhan, lihat Layanan AWS in Scope by Compliance Program.
- 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.

Dokumentasi ini akan membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan bucket direktori. Topik berikut menunjukkan cara mengonfigurasi bucket direktori untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga akan belajar cara menggunakan

Layanan AWS yang lain yang dapat membantu Anda memantau dan mengamankan objek Anda di ember direktori.

Perlindungan data dan enkripsi

Untuk informasi selengkapnya tentang cara mengonfigurasi enkripsi untuk bucket direktori, lihat topik berikut.

Topik

- enkripsi di sisi server
- Pengaturan dan pemantauan enkripsi default untuk bucket direktori
- Menggunakan enkripsi sisi server dengan AWS KMS kunci (SSE-KMS) di bucket direktori
- Enkripsi bergerak
- Penghapusan data

enkripsi di sisi server

Semua bucket direktori memiliki enkripsi yang dikonfigurasi secara default, dan semua objek baru yang diunggah ke bucket direktori secara otomatis dienkripsi saat istirahat. Enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3) adalah konfigurasi enkripsi default untuk setiap bucket direktori. Jika Anda ingin menentukan jenis enkripsi yang berbeda, Anda dapat menggunakan enkripsi sisi server dengan kunci AWS Key Management Service (AWS KMS) (SSE-KMS), dengan mengatur konfigurasi enkripsi default bucket. Untuk informasi selengkapnya tentang SSE-KMS di bucket direktori, lihat. Menggunakan enkripsi sisi server dengan AWS KMS kunci (SSE-KMS) di bucket direktori

Sebaiknya enkripsi default bucket menggunakan konfigurasi enkripsi yang diinginkan dan Anda tidak mengganti enkripsi default bucket dalam `CreateSession` permintaan atau permintaan PUT objek. Kemudian, objek baru secara otomatis dienkripsi dengan pengaturan enkripsi yang diinginkan. Untuk informasi selengkapnya tentang perilaku pengesampingan enkripsi dalam bucket direktori, lihat Menentukan enkripsi sisi server dengan unggahan objek baru. AWS KMS

SSE-KMS dengan bucket direktori berbeda dari SSE-KMS dalam bucket tujuan umum dalam aspek-aspek berikut.

- Konfigurasi SSE-KMS Anda hanya dapat mendukung 1 kunci terkelola pelanggan per bucket direktori selama masa pakai bucket. Kunci yang dikelola AWS(`aws/s3`) tidak didukung. Selain itu,

setelah menentukan kunci terkelola pelanggan untuk SSE-KMS, Anda tidak dapat mengganti kunci terkelola pelanggan untuk konfigurasi SSE-KMS bucket.

Anda dapat mengidentifikasi kunci terkelola pelanggan yang Anda tentukan untuk konfigurasi SSE-KMS bucket, dengan cara berikut:

- Anda membuat permintaan operasi `HeadObject` API untuk menemukan nilai `x-amz-server-side-encryption-aws-kms-key-id` dalam respons Anda.

Untuk menggunakan kunci terkelola pelanggan baru untuk data Anda, sebaiknya salin objek yang ada ke bucket direktori baru dengan kunci terkelola pelanggan baru.

- Untuk operasi API titik akhir Zonal (tingkat objek) kecuali CopyObjectdan UploadPartCopy, Anda mengautentikasi dan mengotorisasi permintaan melalui latensi rendah. CreateSession Sebaiknya enkripsi default bucket menggunakan konfigurasi enkripsi yang diinginkan dan Anda tidak mengganti enkripsi default bucket dalam `CreateSession` permintaan atau permintaan PUT objek. Kemudian, objek baru secara otomatis dienkripsi dengan pengaturan enkripsi yang diinginkan. Untuk mengenkripsi objek baru dalam bucket direktori dengan SSE-KMS, Anda harus menentukan SSE-KMS sebagai konfigurasi enkripsi default bucket direktori dengan kunci KMS (khususnya, kunci yang dikelola pelanggan). Kemudian, ketika sesi dibuat untuk operasi API titik akhir Zonal, objek baru secara otomatis dienkripsi dan didekripsi dengan SSE-KMS dan S3 Bucket Keys selama sesi berlangsung. Untuk informasi selengkapnya tentang perilaku pengesampingan enkripsi dalam bucket direktori, lihat Menentukan enkripsi sisi server dengan unggahan objek baru. AWS KMS

  Dalam panggilan API titik akhir Zonal (kecuali CopyObjectdan UploadPartCopy), Anda tidak dapat mengganti nilai setelan enkripsi (`x-amz-server-side-encryption`,, `x-amz-server-side-encryption-aws-kms-key-idx-amz-server-side-encryption-context`, dan`x-amz-server-side-encryption-bucket-key-enabled`) dari permintaan. `CreateSession` Anda tidak perlu secara eksplisit menentukan nilai setelan enkripsi ini dalam panggilan API titik akhir Zonal, dan Amazon S3 akan menggunakan nilai pengaturan enkripsi dari `CreateSession` permintaan untuk melindungi objek baru di bucket direktori.

  > **Note**
  >
  > Saat Anda menggunakan AWS CLI atau AWS SDKs, for`CreateSession`, token sesi akan diperbarui secara otomatis untuk menghindari gangguan layanan saat sesi berakhir. Konfigurasi enkripsi default bucket AWS CLI atau AWS SDKs gunakan untuk `CreateSession` permintaan tersebut. Itu tidak didukung untuk mengganti nilai pengaturan

enkripsi dalam `CreateSession` permintaan. Selain itu, dalam panggilan API titik akhir Zonal (kecuali CopyObjectdan UploadPartCopy), itu tidak didukung untuk mengganti nilai pengaturan enkripsi dari permintaan. `CreateSession`

- Untuk CopyObject, untuk mengenkripsi salinan objek baru dalam bucket direktori dengan SSE-KMS, Anda harus menentukan SSE-KMS sebagai konfigurasi enkripsi default bucket direktori dengan kunci KMS (khususnya, kunci yang dikelola pelanggan). Kemudian, ketika Anda menentukan pengaturan enkripsi sisi server untuk salinan objek baru dengan SSE-KMS, Anda harus memastikan kunci enkripsi adalah kunci terkelola pelanggan yang sama dengan yang Anda tentukan untuk konfigurasi enkripsi default bucket direktori. Untuk UploadPartCopy, untuk mengenkripsi salinan bagian objek baru dalam bucket direktori dengan SSE-KMS, Anda harus menentukan SSE-KMS sebagai konfigurasi enkripsi default bucket direktori dengan kunci KMS (khususnya, kunci yang dikelola pelanggan). Anda tidak dapat menentukan setelan enkripsi sisi server untuk salinan bagian objek baru dengan SSE-KMS di header permintaan. UploadPartCopy Selain itu, pengaturan enkripsi yang Anda berikan dalam CreateMultipartUploadpermintaan harus sesuai dengan konfigurasi enkripsi default bucket tujuan.
- Kunci Bucket S3 selalu diaktifkan untuk `GET` dan `PUT` beroperasi di bucket direktori dan tidak dapat dinonaktifkan. Kunci Bucket S3 tidak didukung, saat Anda menyalin objek terenkripsi SSE-KMS dari bucket tujuan umum ke bucket direktori, dari bucket direktori ke bucket tujuan umum, atau di antara bucket direktori, melalui, operasi dalam Operasi Batch, atau pekerjaan CopyObject. UploadPartCopyCopyimport Dalam hal ini, Amazon S3 melakukan panggilan ke AWS KMS setiap kali permintaan salinan dibuat untuk objek yang dienkripsi KMS.
- Saat Anda menentukan kunci terkelola AWS KMS pelanggan untuk enkripsi di bucket direktori Anda, hanya gunakan ID kunci atau kunci ARN. Format alias kunci kunci KMS tidak didukung.

Bucket direktori tidak mendukung enkripsi sisi server dua lapis dengan kunci AWS Key Management Service (AWS KMS) (DSSE-KMS), atau enkripsi sisi server dengan kunci enkripsi yang disediakan pelanggan (SSE-C).

Pengaturan dan pemantauan enkripsi default untuk bucket direktori

Bucket Amazon S3 mengaktifkan enkripsi bucket secara default, dan objek baru secara otomatis dienkripsi dengan menggunakan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3). Enkripsi ini berlaku untuk semua objek baru di bucket Amazon S3 Anda, dan Anda tidak akan dikenakan biaya.

Jika Anda memerlukan kontrol lebih besar atas kunci enkripsi Anda, seperti mengelola rotasi kunci dan hibah kebijakan akses, Anda dapat memilih untuk menggunakan enkripsi sisi server dengan kunci AWS Key Management Service (AWS KMS) (SSE-KMS).

> **Note**
>
> - Sebaiknya enkripsi default bucket menggunakan konfigurasi enkripsi yang diinginkan dan Anda tidak mengganti enkripsi default bucket dalam `CreateSession` permintaan atau permintaan PUT objek. Kemudian, objek baru secara otomatis dienkripsi dengan pengaturan enkripsi yang diinginkan. Untuk informasi selengkapnya tentang perilaku pengesampingan enkripsi dalam bucket direktori, lihat Menentukan enkripsi sisi server dengan unggahan objek baru. AWS KMS
> - Untuk mengenkripsi objek baru dalam bucket direktori dengan SSE-KMS, Anda harus menentukan SSE-KMS sebagai konfigurasi enkripsi default bucket direktori dengan kunci KMS (khususnya, kunci yang dikelola pelanggan). Kemudian, ketika sesi dibuat untuk operasi API titik akhir Zonal, objek baru secara otomatis dienkripsi dan didekripsi dengan SSE-KMS dan S3 Bucket Keys selama sesi berlangsung.
> - Saat Anda menyetel enkripsi bucket default ke SSE-KMS, Kunci Bucket S3 selalu diaktifkan untuk `GET` dan PUT beroperasi di bucket direktori dan tidak dapat dinonaktifkan. Kunci Bucket S3 tidak didukung, saat Anda menyalin objek terenkripsi SSE-KMS dari bucket tujuan umum ke bucket direktori, dari bucket direktori ke bucket tujuan umum, atau di antara bucket direktori, melalui, operasi dalam Operasi Batch, atau pekerjaan CopyObject. UploadPartCopyCopyimport Dalam hal ini, Amazon S3 melakukan panggilan ke AWS KMS setiap kali permintaan salinan dibuat untuk objek yang dienkripsi KMS. Untuk informasi selengkapnya tentang cara S3 Bucket Keys mengurangi biaya AWS KMS permintaan Anda, lihatMengurangi biaya SSE-KMS dengan Kunci Bucket Amazon S3.
> - Saat Anda menentukan kunci terkelola AWS KMS pelanggan untuk enkripsi di bucket direktori Anda, hanya gunakan ID kunci atau kunci ARN. Format alias kunci kunci KMS tidak didukung.
> - Enkripsi sisi server dua lapis dengan AWS KMS kunci (DSSE-KMS) dan enkripsi sisi server dengan kunci yang disediakan pelanggan (SSE-C) tidak didukung untuk enkripsi default di bucket direktori.

Untuk informasi selengkapnya tentang mengonfigurasi enkripsi default, lihatMengonfigurasi enkripsi default.

Untuk informasi lebih lanjut tentang izin yang diperlukan untuk enkripsi default, lihat PutBucketEncryption dalam Referensi API Amazon Simple Storage Service.

Anda dapat mengonfigurasi enkripsi default Amazon S3 untuk bucket S3 dengan menggunakan konsol Amazon S3, API REST AWS SDKs Amazon S3, dan (). AWS Command Line Interface AWS CLI

Menggunakan konsol S3

Untuk mengonfigurasi enkripsi default di bucket Amazon S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Dari daftar Bucket, pilih nama bucket yang Anda inginkan.
4. Pilih tab Properti.
5. Di bawah pengaturan enkripsi sisi server, bucket direktori menggunakan enkripsi sisi Server dengan kunci terkelola Amazon S3 (SSE-S3).
6. Pilih Simpan perubahan.

Menggunakan AWS CLI

Contoh ini menunjukkan cara mengonfigurasi enkripsi default dengan menggunakan SSE-S3 atau dengan menggunakan SSE-KMS dengan Kunci Bucket S3.

Untuk informasi selengkapnya tentang enkripsi default, lihat Mengatur perilaku enkripsi sisi server default untuk bucket Amazon S3. Untuk informasi selengkapnya tentang menggunakan enkripsi default AWS CLI untuk mengonfigurasi, lihat put-bucket-encryption.

Example – Enkripsi default dengan SSE-S3

Contoh ini mengonfigurasi enkripsi bucket default dengan kunci terkelola Amazon S3. Untuk menggunakan perintah, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3api put-bucket-encryption --bucket bucket-base-name--zone-id--x-s3 --server-side-
encryption-configuration '{
    "Rules": [
        {
            "ApplyServerSideEncryptionByDefault": {
                "SSEAlgorithm": "AES256"
```

```
            }
        }
    ]
}'
```

Example – Enkripsi default dengan SSE-KMS menggunakan Kunci Bucket S3

Contoh ini mengonfigurasi enkripsi bucket default dengan SSE-KMS menggunakan Kunci Bucket S3. Untuk menggunakan perintah, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3api put-bucket-encryption --bucket bucket-base-name--zone-id--x-s3 --server-side-
encryption-configuration '{
    "Rules": [
            {
                "ApplyServerSideEncryptionByDefault": {
                    "SSEAlgorithm": "aws:kms",
                    "KMSMasterKeyID": "KMS-Key-ARN"
                },
                "BucketKeyEnabled": true
            }
        ]
    }'
```

Penggunaan API REST

Gunakan `PutBucketEncryption` operasi REST API untuk mengatur enkripsi default dengan jenis enkripsi sisi server yang akan digunakan — SSE-S3, atau SSE-KMS.

Untuk informasi selengkapnya, lihat PutBucketEncryption dalam Referensi API Amazon Simple Storage Service.

Menggunakan AWS SDKs

Saat menggunakan AWS SDKs, Anda dapat meminta Amazon S3 untuk digunakan AWS KMS keys untuk enkripsi sisi server. Berikut ini AWS SDKs untuk contoh Java dan .NET mengonfigurasi konfigurasi enkripsi default untuk bucket direktori dengan SSE-KMS dan S3 Bucket Key. Untuk informasi tentang lainnya SDKs, lihat Contoh kode dan pustaka di Pusat AWS Pengembang.

> **Important**
>
> Saat Anda menggunakan enkripsi sisi server AWS KMS key untuk Amazon S3, Anda harus memilih kunci KMS enkripsi simetris. Amazon S3 hanya mendukung kunci KMS enkripsi

simetris. Untuk informasi selengkapnya terkait kunci ini, lihat Membuat kunci enkripsi simetris KMS dalam Panduan Pengembang AWS Key Management Service .

Java

Dengan itu AWS SDK for Java 2.x, Anda dapat meminta Amazon S3 untuk menggunakan metode AWS KMS key dengan menggunakan `applyServerSideEncryptionByDefault` metode untuk menentukan konfigurasi enkripsi default bucket direktori Anda untuk enkripsi data dengan SSE-KMS. Anda membuat kunci KMS enkripsi simetris dan menentukannya dalam permintaan.

```
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.PutBucketEncryptionRequest;
import software.amazon.awssdk.services.s3.model.ServerSideEncryption;
import software.amazon.awssdk.services.s3.model.ServerSideEncryptionByDefault;
import software.amazon.awssdk.services.s3.model.ServerSideEncryptionConfiguration;
import software.amazon.awssdk.services.s3.model.ServerSideEncryptionRule;

public class Main {
    public static void main(String[] args) {
        S3Client s3 = S3Client.create();
        String bucketName = "bucket-base-name--zoneid--x-s3";
        String kmsKeyId = "your-kms-customer-managed-key-id";

        // AWS managed KMS keys aren't supported. Only customer-managed keys are
 supported.
        ServerSideEncryptionByDefault serverSideEncryptionByDefault =
 ServerSideEncryptionByDefault.builder()
                .sseAlgorithm(ServerSideEncryption.AWS_KMS)
                .kmsMasterKeyID(kmsKeyId)
                .build();

        // The bucketKeyEnabled field is enforced to be true.
        ServerSideEncryptionRule rule = ServerSideEncryptionRule.builder()
                .bucketKeyEnabled(true)
                .applyServerSideEncryptionByDefault(serverSideEncryptionByDefault)
                .build();

        ServerSideEncryptionConfiguration serverSideEncryptionConfiguration =
 ServerSideEncryptionConfiguration.builder()
                .rules(rule)
```

```
                .build();

        PutBucketEncryptionRequest putRequest = PutBucketEncryptionRequest.builder()
                .bucket(bucketName)

 .serverSideEncryptionConfiguration(serverSideEncryptionConfiguration)
                .build();

        s3.putBucketEncryption(putRequest);

    }
}
```

Untuk informasi selengkapnya tentang membuat kunci terkelola pelanggan, lihat Memprogram AWS KMS API di Panduan AWS Key Management Service Pengembang.

Untuk contoh kode kerja mengunggah sebuah objek, lihat topik berikut ini. Untuk menggunakan contoh ini, Anda harus memperbarui contoh kode dan memberikan informasi enkripsi seperti yang ditunjukkan dalam fragmen kode sebelumnya.

- Untuk mengunggah objek dalam operasi tunggal, lihat Mengunggah objek ke ember direktori.
- Untuk operasi API unggahan multibagian, lihatMenggunakan unggahan multibagian dengan bucket direktori.

.NET

Dengan itu AWS SDK untuk .NET, Anda dapat meminta Amazon S3 untuk menggunakan properti AWS KMS key dengan menggunakan `ServerSideEncryptionByDefault` properti untuk menentukan konfigurasi enkripsi default bucket direktori Anda untuk enkripsi data dengan SSE-KMS. Anda membuat kunci terkelola pelanggan enkripsi simetris dan menentukannya dalam permintaan.

```
    // Set the bucket server side encryption to use AWSKMS with a customer-managed
 key id.
    // bucketName: Name of the directory bucket. "bucket-base-name--zonsid--x-s3"
    // kmsKeyId: The Id of the customer managed KMS Key. "your-kms-customer-managed-
key-id"
    // Returns True if successful.
```

```
   public static async Task<bool> SetBucketServerSideEncryption(string bucketName,
string kmsKeyId)
   {
       var serverSideEncryptionByDefault = new ServerSideEncryptionConfiguration
       {
           ServerSideEncryptionRules = new List<ServerSideEncryptionRule>
           {
               new ServerSideEncryptionRule
               {
                   ServerSideEncryptionByDefault = new
ServerSideEncryptionByDefault
                   {
                       ServerSideEncryptionAlgorithm =
ServerSideEncryptionMethod.AWSKMS,
                       ServerSideEncryptionKeyManagementServiceKeyId = kmsKeyId
                   }
               }
           }
       };
       try
       {
           var encryptionResponse =await _s3Client.PutBucketEncryptionAsync(new
PutBucketEncryptionRequest
           {
               BucketName = bucketName,
               ServerSideEncryptionConfiguration = serverSideEncryptionByDefault,
           });

           return encryptionResponse.HttpStatusCode == HttpStatusCode.OK;
       }
       catch (AmazonS3Exception ex)
       {
           Console.WriteLine(ex.ErrorCode == "AccessDenied"
               ? $"This account does not have permission to set encryption on
{bucketName}, please try again."
               : $"Unable to set bucket encryption for bucket {bucketName},
{ex.Message}");
       }
       return false;
   }
```

Untuk informasi selengkapnya tentang membuat kunci terkelola pelanggan, lihat Memprogram AWS KMS API di Panduan AWS Key Management Service Pengembang.

Untuk contoh kode kerja mengunggah sebuah objek, lihat topik berikut ini. Untuk menggunakan contoh ini, Anda harus memperbarui contoh kode dan memberikan informasi enkripsi seperti yang ditunjukkan dalam fragmen kode sebelumnya.

- Untuk mengunggah objek dalam operasi tunggal, lihat Mengunggah objek ke ember direktori.
- Untuk operasi API unggahan multibagian, lihatMenggunakan unggahan multibagian dengan bucket direktori.

Memantau enkripsi default untuk bucket direktori dengan AWS CloudTrail

Anda dapat melacak permintaan konfigurasi enkripsi default untuk bucket direktori Amazon S3 dengan menggunakan peristiwa. AWS CloudTrail Nama peristiwa API berikut digunakan dalam CloudTrail log:

- `PutBucketEncryption`
- `GetBucketEncryption`
- `DeleteBucketEncryption`

> **Note**
>
> - EventBridge tidak didukung dalam bucket direktori.
> - Enkripsi sisi server dua lapis dengan kunci AWS Key Management Service (AWS KMS) (DSSE-KMS) atau enkripsi sisi server dengan kunci enkripsi yang disediakan pelanggan (SSE-C) tidak didukung dalam bucket direktori.

Untuk informasi selengkapnya tentang memantau enkripsi default dengan AWS CloudTrail, lihatMemantau enkripsi default dengan AWS CloudTrail dan Amazon EventBridge.

Menggunakan enkripsi sisi server dengan AWS KMS kunci (SSE-KMS) di bucket direktori

Kontrol keamanan AWS KMS dapat membantu Anda memenuhi persyaratan kepatuhan terkait enkripsi. Anda dapat memilih untuk mengonfigurasi bucket direktori untuk menggunakan enkripsi sisi

server dengan kunci AWS Key Management Service (AWS KMS) (SSE-KMS) dan menggunakan kunci KMS ini untuk melindungi data Anda di bucket direktori Amazon S3. Untuk informasi lebih lanjut tentang SSE-KMS, lihat Menggunakan enkripsi sisi server dengan AWS KMS kunci (SSE-KMS).

Izin

Untuk mengunggah atau mengunduh objek yang dienkripsi dengan AWS KMS key ke atau dari Amazon S3, Anda `kms:GenerateDataKey` perlu `kms:Decrypt` dan izin pada kunci. Untuk informasi selengkapnya, lihat Mengizinkan pengguna di akun lain untuk menggunakan kunci KMS untuk operasi kriptografi di Panduan Pengembang AWS Key Management Service . Untuk informasi tentang AWS KMS izin yang diperlukan untuk unggahan multibagian, lihat. API dan izin unggahan multibagian

Untuk informasi selengkapnya tentang kunci KMS untuk SSE-KMS, lihat. Menentukan enkripsi di sisi server dengan AWS KMS (SSE-KMS)

Topik

- AWS KMS keys
- Menggunakan SSE-KMS untuk operasi lintas akun
- Kunci Bucket Amazon S3
- Membutuhkan SSE-KMS
- Konteks enkripsi
- Mengirim permintaan untuk objek AWS KMS terenkripsi
- Mengaudit enkripsi SSE-KMS di bucket direktori
- Menentukan enkripsi sisi server dengan AWS KMS (SSE-KMS) untuk unggahan objek baru di bucket direktori

AWS KMS keys

Konfigurasi SSE-KMS Anda hanya dapat mendukung 1 kunci terkelola pelanggan per bucket direktori selama masa pakai bucket. Kunci yang dikelola AWS(`aws/s3`) tidak didukung. Selain itu, setelah menentukan kunci terkelola pelanggan untuk SSE-KMS, Anda tidak dapat mengganti kunci terkelola pelanggan untuk konfigurasi SSE-KMS bucket.

Anda dapat mengidentifikasi kunci terkelola pelanggan yang Anda tentukan untuk konfigurasi SSE-KMS bucket, dengan cara berikut:

- Anda membuat permintaan operasi `HeadObject` API untuk menemukan nilai `x-amz-server-side-encryption-aws-kms-key-id` dalam respons Anda.

Untuk menggunakan kunci terkelola pelanggan baru untuk data Anda, sebaiknya salin objek yang ada ke bucket direktori baru dengan kunci terkelola pelanggan baru.

Saat Anda menentukan kunci terkelola AWS KMS pelanggan untuk enkripsi di bucket direktori Anda, hanya gunakan ID kunci atau kunci ARN. Format alias kunci kunci KMS tidak didukung.

Untuk informasi selengkapnya tentang kunci KMS untuk SSE-KMS, lihat. AWS KMS keys

Menggunakan SSE-KMS untuk operasi lintas akun

Saat menggunakan enkripsi untuk operasi lintas akun di bucket direktori, perhatikan hal berikut:

- Jika Anda ingin memberikan akses lintas akun ke objek S3 Anda, konfigurasikan kebijakan kunci yang dikelola pelanggan untuk mengizinkan akses dari akun lain.
- Untuk menentukan kunci yang dikelola pelanggan, Anda harus menggunakan ARN kunci KMS yang memenuhi syarat sepenuhnya.

Kunci Bucket Amazon S3

Kunci Bucket S3 selalu diaktifkan untuk `GET` dan `PUT` beroperasi di bucket direktori dan tidak dapat dinonaktifkan. Kunci Bucket S3 tidak didukung, saat Anda menyalin objek terenkripsi SSE-KMS dari bucket tujuan umum ke bucket direktori, dari bucket direktori ke bucket tujuan umum, atau di antara bucket direktori, melalui, operasi dalam Operasi Batch, atau pekerjaan CopyObject. UploadPartCopyCopyimport Dalam hal ini, Amazon S3 melakukan panggilan ke AWS KMS setiap kali permintaan salinan dibuat untuk objek yang dienkripsi KMS.

Untuk operasi API titik akhir Zonal (tingkat objek) kecuali CopyObjectdan UploadPartCopy, Anda mengautentikasi dan mengotorisasi permintaan melalui latensi rendah. CreateSession Sebaiknya enkripsi default bucket menggunakan konfigurasi enkripsi yang diinginkan dan Anda tidak mengganti enkripsi default bucket dalam `CreateSession` permintaan atau permintaan `PUT` objek. Kemudian, objek baru secara otomatis dienkripsi dengan pengaturan enkripsi yang diinginkan. Untuk mengenkripsi objek baru dalam bucket direktori dengan SSE-KMS, Anda harus menentukan SSE-KMS sebagai konfigurasi enkripsi default bucket direktori dengan kunci KMS (khususnya, kunci yang dikelola pelanggan). Kemudian, ketika sesi dibuat untuk operasi API titik akhir Zonal, objek baru secara otomatis dienkripsi dan didekripsi dengan SSE-KMS dan S3 Bucket Keys selama sesi

berlangsung. Untuk informasi selengkapnya tentang perilaku pengesampingan enkripsi dalam bucket direktori, lihat Menentukan enkripsi sisi server dengan unggahan objek baru. AWS KMS

Kunci Bucket S3 digunakan untuk jangka waktu terbatas dalam Amazon S3, yang selanjutnya mengurangi kebutuhan Amazon S3 untuk membuat permintaan AWS KMS untuk menyelesaikan operasi enkripsi. Untuk informasi selengkapnya tentang penggunaan S3 Bucket Keys, lihat Kunci Bucket Amazon S3 danMengurangi biaya SSE-KMS dengan Kunci Bucket Amazon S3.

Membutuhkan SSE-KMS

Untuk meminta SSE-KMS dari semua objek dalam bucket direktori tertentu, Anda dapat menggunakan kebijakan bucket. Misalnya, saat Anda menggunakan operasi `CreateSession` API untuk memberikan izin mengunggah objek baru (`PutObject`,, dan`CreateMultipartUpload`)`CopyObject`, kebijakan bucket berikut menolak izin objek upload (`s3express:CreateSession`) kepada semua orang jika `CreateSession` permintaan tidak menyertakan `x-amz-server-side-encryption-aws-kms-key-id` header yang meminta SSE-KMS.

```
{
   "Version":"2012-10-17",
   "Id":"UploadObjectPolicy",
   "Statement":[{
         "Sid":"DenyObjectsThatAreNotSSEKMS",
         "Effect":"Deny",
         "Principal":"*",
         "Action":"s3express:CreateSession",
         "Resource":"arn:aws:s3express:region:account-id:bucket/bucket-base-name--zone-id--x-s3/*",
         "Condition":{
            "Null":{
               "s3express:x-amz-server-side-encryption-aws-kms-key-id":"true"
            }
         }
      }
   ]
}
```

Untuk mengharuskan seseorang AWS KMS key digunakan untuk mengenkripsi objek dalam ember, Anda dapat menggunakan kunci `s3express:x-amz-server-side-encryption-aws-kms-key-id` kondisi. Untuk menentukan kunci KMS, Anda harus menggunakan kunci Amazon Resource

Name (ARN) yang ada dalam `arn:aws:kms:region:acct-id:key/key-id` format. AWS Identity and Access Management tidak memvalidasi jika string untuk `s3express:x-amz-server-side-encryption-aws-kms-key-id` ada. ID AWS KMS kunci yang digunakan Amazon S3 untuk enkripsi objek harus cocok dengan ID AWS KMS kunci dalam kebijakan, jika tidak Amazon S3 menolak permintaan tersebut.

Untuk informasi selengkapnya tentang cara menggunakan SSE-KMS untuk upload objek baru, lihat. Menentukan enkripsi sisi server dengan AWS KMS (SSE-KMS) untuk unggahan objek baru di bucket direktori

Untuk daftar lengkap kunci kondisi khusus untuk bucket direktori, lihatMengotorisasi operasi API endpoint Regional dengan IAM.

Konteks enkripsi

Untuk bucket direktori, konteks enkripsi adalah sekumpulan pasangan kunci-nilai yang berisi informasi kontekstual tentang data. Nilai konteks enkripsi tambahan tidak didukung. Untuk informasi selengkapnya tentang konteks enkripsi, lihatKonteks enkripsi.

Secara default, jika Anda menggunakan SSE-KMS pada bucket direktori, Amazon S3 menggunakan bucket Amazon Resource Name (ARN) sebagai pasangan konteks enkripsi:

```
arn:aws:s3express:region:account-id:bucket/bucket-base-name--zone-id--x-s3
```

Pastikan kebijakan IAM atau kebijakan AWS KMS utama Anda menggunakan ARN bucket Anda sebagai konteks enkripsi.

Anda dapat memberikan pasangan konteks enkripsi eksplisit secara opsional dengan menggunakan `x-amz-server-side-encryption-context` header dalam permintaan API titik akhir Zonal, seperti. CreateSession Nilai header ini adalah string yang dikodekan Base64 dari JSON yang dikodekan UTF-8, yang berisi konteks enkripsi sebagai pasangan nilai kunci. Untuk bucket direktori, konteks enkripsi harus sesuai dengan konteks enkripsi default — bucket Amazon Resource Name (ARN). Juga, karena konteks enkripsi tidak dienkripsi, pastikan itu tidak menyertakan informasi sensitif.

Anda dapat menggunakan konteks enkripsi untuk mengidentifikasi dan mengategorikan operasi kriptografi Anda. Anda juga dapat menggunakan nilai ARN konteks enkripsi default untuk melacak permintaan yang relevan AWS CloudTrail dengan melihat bucket direktori ARN mana yang digunakan dengan kunci enkripsi mana.

Di `requestParameters` bidang file CloudTrail log, jika Anda menggunakan SSE-KMS pada bucket direktori, nilai konteks enkripsi adalah ARN bucket.

```
"encryptionContext": {
    "aws:s3express:arn": "arn:aws:s3:::arn:aws:s3express:region:account-id:bucket/bucket-base-name--zone-id--x-s3"
}
```

Selain itu, untuk enkripsi objek dengan SSE-KMS di bucket direktori, AWS KMS CloudTrail peristiwa Anda mencatat ARN bucket Anda alih-alih ARN objek Anda.

Mengirim permintaan untuk objek AWS KMS terenkripsi

Bucket direktori hanya dapat diakses melalui HTTPS (TLS). Selain itu, bucket direktori menandatangani permintaan dengan menggunakan AWS Signature Version 4 (SigV4). Untuk informasi selengkapnya tentang mengirim permintaan untuk objek AWS KMS terenkripsi, lihat. Mengirim permintaan untuk objek AWS KMS terenkripsi

Jika objek Anda menggunakan SSE-KMS, jangan mengirim header permintaan enkripsi untuk permintaan `GET` dan permintaan `HEAD`. Jika tidak, Anda akan mendapatkan kesalahan HTTP 400 Bad Request.

Mengaudit enkripsi SSE-KMS di bucket direktori

Untuk mengaudit penggunaan AWS KMS kunci Anda untuk data terenkripsi SSE-KMS Anda, Anda dapat menggunakan log. AWS CloudTrail Anda bisa mendapatkan wawasan tentang operasi kriptografi Anda, seperti GenerateDataKeydan Decrypt. CloudTrail mendukung banyak nilai atribut untuk memfilter pencarian Anda, termasuk nama acara, nama pengguna, dan sumber acara.

Topik

- Menentukan enkripsi sisi server dengan AWS KMS (SSE-KMS) untuk unggahan objek baru di bucket direktori

Menentukan enkripsi sisi server dengan AWS KMS (SSE-KMS) untuk unggahan objek baru di bucket direktori

Untuk bucket direktori, untuk mengenkripsi data Anda dengan enkripsi sisi server, Anda dapat menggunakan enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3) (default) atau enkripsi sisi server dengan () kunci (SSE-KMS). AWS Key Management Service AWS KMS Sebaiknya enkripsi default bucket menggunakan konfigurasi enkripsi yang diinginkan dan Anda tidak mengganti

enkripsi default bucket dalam `CreateSession` permintaan atau permintaan PUT objek. Kemudian, objek baru secara otomatis dienkripsi dengan pengaturan enkripsi yang diinginkan. Untuk informasi selengkapnya tentang perilaku pengesampingan enkripsi dalam bucket direktori, lihat Menentukan enkripsi sisi server dengan unggahan objek baru. AWS KMS

Semua bucket Amazon S3 memiliki enkripsi yang dikonfigurasi secara default, dan semua objek baru yang diunggah ke bucket S3 secara otomatis dienkripsi saat sedang tidak aktif. Enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3) adalah konfigurasi enkripsi default untuk setiap bucket di Amazon S3. Jika Anda ingin menentukan jenis enkripsi yang berbeda untuk bucket direktori, Anda dapat menggunakan enkripsi sisi server dengan kunci AWS Key Management Service (AWS KMS) (SSE-KMS). Untuk mengenkripsi objek baru dalam bucket direktori dengan SSE-KMS, Anda harus menentukan SSE-KMS sebagai konfigurasi enkripsi default bucket direktori dengan kunci KMS (khususnya, kunci yang dikelola pelanggan). Kunci yang dikelola AWS(`aws/s3`) tidak didukung. Konfigurasi SSE-KMS Anda hanya dapat mendukung 1 kunci terkelola pelanggan per bucket direktori selama masa pakai bucket. Setelah menentukan kunci terkelola pelanggan untuk SSE-KMS, Anda tidak dapat mengganti kunci terkelola pelanggan untuk konfigurasi SSE-KMS bucket. Kemudian, ketika Anda menentukan pengaturan enkripsi sisi server untuk objek baru dengan SSE-KMS, Anda harus memastikan kunci enkripsi adalah kunci terkelola pelanggan yang sama dengan yang Anda tentukan untuk konfigurasi enkripsi default bucket direktori. Untuk menggunakan kunci terkelola pelanggan baru untuk data Anda, sebaiknya salin objek yang ada ke bucket direktori baru dengan kunci terkelola pelanggan baru.

Anda dapat menerapkan enkripsi saat mengunggah objek baru, atau menyalin objek yang sudah ada. Jika Anda mengubah enkripsi objek, sebuah objek baru akan dibuat untuk menggantikan objek yang lama.

Anda dapat menentukan SSE-KMS dengan menggunakan operasi REST API, AWS SDKs, dan (). AWS Command Line Interface AWS CLI

> **Note**
>
> - Untuk bucket direktori, perilaku utama enkripsi adalah sebagai berikut:
>   - Saat Anda menggunakan CreateSessionREST API untuk mengautentikasi dan mengotorisasi permintaan API titik akhir Zonal kecuali CopyObjectdan UploadPartCopy, Anda dapat mengganti pengaturan enkripsi ke SSE-S3 atau ke SSE-KMS hanya jika Anda menetapkan enkripsi default bucket dengan SSE-KMS sebelumnya.
>   - Bila Anda menggunakan CreateSessiondengan AWS CLI atau AWS SDKs untuk mengautentikasi dan mengotorisasi permintaan API titik akhir Zonal kecuali

CopyObjectdan UploadPartCopy, Anda tidak dapat mengganti pengaturan enkripsi sama sekali.

  - Ketika Anda membuat CopyObjectpermintaan, Anda dapat mengganti pengaturan enkripsi ke SSE-S3 atau ke SSE-KMS hanya jika Anda menetapkan enkripsi default bucket dengan SSE-KMS sebelumnya. Saat Anda membuat UploadPartCopypermintaan, Anda tidak dapat mengganti pengaturan enkripsi.

- Anda dapat menggunakan Multi-wilayah AWS KMS keys di Amazon S3. Namun, Amazon S3 saat ini memperlakukan kunci multi-Wilayah selayaknya kunci satu Wilayah, dan tidak menggunakan fitur multi-Wilayah dari kunci tersebut. Untuk informasi selengkapnya, lihat Menggunakan kunci multi-Wilayah di Panduan Pengembang AWS Key Management Service .
- Jika Anda ingin menggunakan kunci KMS yang dimiliki oleh akun lain, Anda harus memiliki izin untuk menggunakan kunci tersebut. Untuk informasi selengkapnya tentang izin lintas akun untuk kunci KMS, lihat Membuat kunci KMS yang dapat digunakan oleh akun lain di Panduan Pengembang AWS Key Management Service .

Penggunaan API REST

> **Note**
>
> Hanya 1 kunci terkelola pelanggan yang didukung per bucket direktori selama masa pakai bucket. Kunci yang dikelola AWS(`aws/s3`) tidak didukung. Setelah menentukan SSE-KMS sebagai konfigurasi enkripsi default bucket dengan kunci terkelola pelanggan, Anda tidak dapat mengubah kunci terkelola pelanggan untuk konfigurasi SSE-KMS bucket.

Untuk operasi API titik akhir Zonal (tingkat objek) kecuali CopyObjectdan UploadPartCopy, Anda mengautentikasi dan mengotorisasi permintaan melalui latensi rendah. CreateSession Sebaiknya enkripsi default bucket menggunakan konfigurasi enkripsi yang diinginkan dan Anda tidak mengganti enkripsi default bucket dalam `CreateSession` permintaan atau permintaan PUT objek. Kemudian, objek baru secara otomatis dienkripsi dengan pengaturan enkripsi yang diinginkan. Untuk mengenkripsi objek baru dalam bucket direktori dengan SSE-KMS, Anda harus menentukan SSE-KMS sebagai konfigurasi enkripsi default bucket direktori dengan kunci KMS (khususnya, kunci yang dikelola pelanggan). Kemudian, ketika sesi dibuat untuk operasi API titik akhir Zonal, objek baru secara otomatis dienkripsi dan didekripsi dengan SSE-KMS dan S3 Bucket Keys selama sesi

berlangsung. Untuk informasi selengkapnya tentang perilaku pengesampingan enkripsi dalam bucket direktori, lihat Menentukan enkripsi sisi server dengan unggahan objek baru. AWS KMS

Dalam panggilan API titik akhir Zonal (kecuali CopyObjectdan UploadPartCopy) menggunakan REST API, Anda tidak dapat mengganti nilai setelan enkripsi (`x-amz-server-side-encryption`,, `x-amz-server-side-encryption-aws-kms-key-idx-amz-server-side-encryption-context`, dan`x-amz-server-side-encryption-bucket-key-enabled`) dari permintaan. `CreateSession` Anda tidak perlu secara eksplisit menentukan nilai setelan enkripsi ini dalam panggilan API titik akhir Zonal, dan Amazon S3 akan menggunakan nilai pengaturan enkripsi dari `CreateSession` permintaan untuk melindungi objek baru di bucket direktori.

> **Note**
>
> Saat Anda menggunakan AWS CLI atau AWS SDKs, for`CreateSession`, token sesi akan diperbarui secara otomatis untuk menghindari gangguan layanan saat sesi berakhir. Konfigurasi enkripsi default bucket AWS CLI atau AWS SDKs gunakan untuk `CreateSession` permintaan tersebut. Itu tidak didukung untuk mengganti nilai pengaturan enkripsi dalam `CreateSession` permintaan. Selain itu, dalam panggilan API titik akhir Zonal (kecuali CopyObjectdan UploadPartCopy), tidak didukung untuk mengganti nilai pengaturan enkripsi dari permintaan. `CreateSession`

Untuk CopyObject, untuk mengenkripsi salinan objek baru dalam bucket direktori dengan SSE-KMS, Anda harus menentukan SSE-KMS sebagai konfigurasi enkripsi default bucket direktori dengan kunci KMS (khususnya, kunci yang dikelola pelanggan). Kemudian, ketika Anda menentukan pengaturan enkripsi sisi server untuk salinan objek baru dengan SSE-KMS, Anda harus memastikan kunci enkripsi adalah kunci terkelola pelanggan yang sama dengan yang Anda tentukan untuk konfigurasi enkripsi default bucket direktori. Untuk UploadPartCopy, untuk mengenkripsi salinan bagian objek baru dalam bucket direktori dengan SSE-KMS, Anda harus menentukan SSE-KMS sebagai konfigurasi enkripsi default bucket direktori dengan kunci KMS (khususnya, kunci yang dikelola pelanggan). Anda tidak dapat menentukan setelan enkripsi sisi server untuk salinan bagian objek baru dengan SSE-KMS di header permintaan. UploadPartCopy Selain itu, pengaturan enkripsi yang Anda berikan dalam CreateMultipartUploadpermintaan harus sesuai dengan konfigurasi enkripsi default bucket tujuan.

Topik

- Operasi API REST Amazon S3 yang mendukung SSE-KMS

- Konteks enkripsi (x-amz-server-side-encryption-context)
- AWS KMS ID kunci (x-amz-server-side-encryption-aws-kms-key-id)
- Tombol Bucket S3 () x-amz-server-side-encryption-aws-bucket-key-enabled

Operasi API REST Amazon S3 yang mendukung SSE-KMS

Operasi REST API tingkat objek berikut dalam bucket direktori menerima header`x-amz-server-side-encryption`,`x-amz-server-side-encryption-aws-kms-key-id`, dan `x-amz-server-side-encryption-context` permintaan.

- CreateSession— Saat Anda menggunakan operasi API titik akhir Zonal (tingkat objek) (kecuali CopyObject dan UploadPartCopy), Anda dapat menentukan header permintaan ini.
- PutObject–Saat Anda mengunggah data dengan menggunakan operasi API PUT, Anda dapat menentukan header permintaan ini.
- CopyObject–Saat Anda menyalin objek, Anda memiliki objek sumber dan objek target. Ketika Anda melewati header SSE-KMS dengan `CopyObject` operasi, mereka hanya diterapkan ke objek target.
- CreateMultipartUpload— Saat Anda mengunggah objek besar dengan menggunakan operasi API unggahan multibagian, Anda dapat menentukan header ini. Anda menentukan header ini dalam `CreateMultipartUpload` permintaan.

Header respons dari operasi API REST berikut mengembalikan header `x-amz-server-side-encryption` saat objek disimpan menggunakan enkripsi di sisi server.

- CreateSession
- PutObject
- CopyObject
- POST Object
- CreateMultipartUpload
- UploadPart
- UploadPartCopy
- CompleteMultipartUpload
- GetObject
- HeadObject

> **Important**
>
> - Semua GET dan PUT permintaan untuk objek yang dilindungi oleh AWS KMS gagal jika Anda tidak membuat permintaan ini dengan menggunakan Transport Layer Security (TLS), atau Signature Version 4.
> - Jika objek Anda menggunakan SSE-KMS, jangan mengirim header permintaan enkripsi untuk GET permintaan dan HEAD permintaan, atau Anda akan mendapatkan kesalahan HTTP 400. BadRequest

Konteks enkripsi (**`x-amz-server-side-encryption-context`**)

Jika Anda menentukan`x-amz-server-side-encryption:aws:kms`, Amazon S3 API mendukung Anda untuk secara opsional menyediakan konteks enkripsi eksplisit dengan header. `x-amz-server-side-encryption-context` Untuk bucket direktori, konteks enkripsi adalah sekumpulan pasangan kunci-nilai yang berisi informasi kontekstual tentang data. Nilai harus sesuai dengan konteks enkripsi default — bucket Amazon Resource Name (ARN). Nilai konteks enkripsi tambahan tidak didukung.

Untuk informasi tentang konteks enkripsi dalam bucket direktori, lihatKonteks enkripsi. Untuk informasi umum tentang konteks enkripsi, lihat AWS Key Management Service Konsep-Konteks enkripsi dalam Panduan Pengembang AWS Key Management Service .

AWS KMS ID kunci (**`x-amz-server-side-encryption-aws-kms-key-id`**)

Anda dapat menggunakan header `x-amz-server-side-encryption-aws-kms-key-id` untuk menentukan ID CMK, yang digunakan untuk melindungi data.

Konfigurasi SSE-KMS Anda hanya dapat mendukung 1 kunci terkelola pelanggan per bucket direktori selama masa pakai bucket. Kunci yang dikelola AWS(`aws/s3`) tidak didukung. Selain itu, setelah menentukan kunci terkelola pelanggan untuk SSE-KMS, Anda tidak dapat mengganti kunci terkelola pelanggan untuk konfigurasi SSE-KMS bucket.

Anda dapat mengidentifikasi kunci terkelola pelanggan yang Anda tentukan untuk konfigurasi SSE-KMS bucket, dengan cara berikut:

- Anda membuat permintaan operasi `HeadObject` API untuk menemukan nilai `x-amz-server-side-encryption-aws-kms-key-id` dalam respons Anda.

Untuk menggunakan kunci terkelola pelanggan baru untuk data Anda, sebaiknya salin objek yang ada ke bucket direktori baru dengan kunci terkelola pelanggan baru.

Untuk informasi tentang konteks enkripsi dalam bucket direktori, lihatAWS KMS keys.

Tombol Bucket S3 () **`x-amz-server-side-encryption-aws-bucket-key-enabled`**

Kunci Bucket S3 selalu diaktifkan untuk `GET` dan `PUT` beroperasi di bucket direktori dan tidak dapat dinonaktifkan. Kunci Bucket S3 tidak didukung, saat Anda menyalin objek terenkripsi SSE-KMS dari bucket tujuan umum ke bucket direktori, dari bucket direktori ke bucket tujuan umum, atau di antara bucket direktori, melalui, operasi dalam Operasi Batch, atau pekerjaan CopyObject. UploadPartCopyCopyimport Dalam hal ini, Amazon S3 melakukan panggilan ke AWS KMS setiap kali permintaan salinan dibuat untuk objek yang dienkripsi KMS. Untuk informasi tentang Kunci Bucket S3 di bucket direktori, lihat. Konteks enkripsi

Menggunakan AWS CLI

> **Note**
>
> Saat Anda menggunakan AWS CLI, for`CreateSession`, token sesi akan diperbarui secara otomatis untuk menghindari gangguan layanan saat sesi kedaluwarsa. Itu tidak didukung untuk mengganti nilai pengaturan enkripsi untuk `CreateSession` permintaan. Selain itu, dalam panggilan API titik akhir Zonal (kecuali CopyObjectdan UploadPartCopy), tidak didukung untuk mengganti nilai pengaturan enkripsi dari permintaan. `CreateSession` Untuk mengenkripsi objek baru dalam bucket direktori dengan SSE-KMS, Anda harus menentukan SSE-KMS sebagai konfigurasi enkripsi default bucket direktori dengan kunci KMS (khususnya, kunci yang dikelola pelanggan). Kemudian, ketika sesi dibuat untuk operasi API titik akhir Zonal, objek baru secara otomatis dienkripsi dan didekripsi dengan SSE-KMS dan S3 Bucket Keys selama sesi berlangsung.

Untuk menggunakan contoh AWS CLI perintah berikut, ganti *`user input placeholders`* dengan informasi Anda sendiri.

Saat Anda mengunggah objek baru atau menyalin objek yang ada, Anda dapat menentukan penggunaan enkripsi sisi server dengan AWS KMS kunci untuk mengenkripsi data Anda. Untuk melakukan ini, gunakan `put-bucket-encryption` perintah untuk mengatur konfigurasi enkripsi default bucket direktori sebagai SSE-KMS (). `aws:kms` Secara khusus, tambahkan `--server-`

`side-encryption aws:kms` header ke permintaan. Gunakan tombol `--ssekms-key-id` *`example-key-id`* untuk menambahkan AWS KMS kunci terkelola pelanggan yang Anda buat. Jika Anda menentukan`--server-side-encryption aws:kms`, Anda harus memberikan ID AWS KMS kunci kunci yang dikelola pelanggan Anda. Bucket direktori tidak menggunakan kunci AWS terkelola. Untuk contoh perintah, lihat Menggunakan AWS CLI.

Kemudian, saat Anda mengunggah objek baru dengan perintah berikut, Amazon S3 menggunakan pengaturan bucket untuk enkripsi default untuk mengenkripsi objek secara default.

```
aws s3api put-object --bucket bucket-base-name--zone-id--x-s3 --key example-object-key
 --body filepath
```

Anda tidak perlu menambahkan `-\-bucket-key-enabled` secara eksplisit dalam perintah operasi API titik akhir Zonal Anda. Kunci Bucket S3 selalu diaktifkan untuk `GET` dan `PUT` beroperasi di bucket direktori dan tidak dapat dinonaktifkan. Kunci Bucket S3 tidak didukung, saat Anda menyalin objek terenkripsi SSE-KMS dari bucket tujuan umum ke bucket direktori, dari bucket direktori ke bucket tujuan umum, atau di antara bucket direktori, melalui, operasi dalam Operasi Batch, atau pekerjaan CopyObject. UploadPartCopyCopyimport Dalam hal ini, Amazon S3 melakukan panggilan ke AWS KMS setiap kali permintaan salinan dibuat untuk objek yang dienkripsi KMS.

Anda dapat menyalin objek dari bucket sumber (misalnya, bucket tujuan umum) ke bucket baru (misalnya, bucket direktori) dan menggunakan enkripsi SSE-KMS untuk objek tujuan. Untuk melakukan ini, gunakan `put-bucket-encryption` perintah untuk mengatur konfigurasi enkripsi default bucket tujuan (misalnya, bucket direktori) sebagai SSE-KMS (). `aws:kms` Untuk contoh perintah, lihat Menggunakan AWS CLI. Kemudian, ketika Anda menyalin objek dengan perintah berikut, Amazon S3 menggunakan pengaturan bucket untuk enkripsi default untuk mengenkripsi objek secara default.

```
aws s3api copy-object --copy-source amzn-s3-demo-bucket/example-object-key --
bucket bucket-base-name--zone-id--x-s3 --key example-object-key
```

Menggunakan AWS SDKs

Saat menggunakan AWS SDKs, Anda dapat meminta Amazon S3 untuk digunakan AWS KMS keys untuk enkripsi sisi server. Contoh berikut menunjukkan bagaimana menggunakan SSE-KMS dengan AWS SDKs untuk Java dan .NET. Untuk informasi tentang lainnya SDKs, lihat Contoh kode dan pustaka di Pusat AWS Pengembang.

> **Note**
>
> Saat Anda menggunakan AWS SDKs, for`CreateSession`, token sesi akan diperbarui secara otomatis untuk menghindari gangguan layanan saat sesi berakhir. Itu tidak didukung untuk mengganti nilai pengaturan enkripsi untuk `CreateSession` permintaan. Selain itu, dalam panggilan API titik akhir Zonal (kecuali CopyObjectdan UploadPartCopy), tidak didukung untuk mengganti nilai pengaturan enkripsi dari permintaan. `CreateSession` Untuk mengenkripsi objek baru dalam bucket direktori dengan SSE-KMS, Anda harus menentukan SSE-KMS sebagai konfigurasi enkripsi default bucket direktori dengan kunci KMS (khususnya, kunci yang dikelola pelanggan). Kemudian, ketika sesi dibuat untuk operasi API titik akhir Zonal, objek baru secara otomatis dienkripsi dan didekripsi dengan SSE-KMS dan S3 Bucket Keys selama sesi berlangsung.
>
> Untuk informasi selengkapnya tentang penggunaan AWS SDKs untuk menyetel konfigurasi enkripsi default bucket direktori sebagai SSE-KMS, lihat. Menggunakan AWS SDKs

> **Important**
>
> Saat Anda menggunakan enkripsi sisi server AWS KMS key untuk Amazon S3, Anda harus memilih kunci KMS enkripsi simetris. Amazon S3 hanya mendukung kunci KMS enkripsi simetris. Untuk informasi selengkapnya terkait kunci ini, lihat Membuat kunci enkripsi simetris KMS dalam Panduan Pengembang AWS Key Management Service .

Untuk informasi selengkapnya tentang membuat kunci terkelola pelanggan, lihat Memprogram AWS KMS API di Panduan AWS Key Management Service Pengembang.

Enkripsi bergerak

Bucket direktori menggunakan titik akhir API Regional dan Zonal. Bergantung pada operasi Amazon S3 API yang Anda gunakan, titik akhir Regional atau Zonal diperlukan. Anda dapat mengakses titik akhir Zonal dan Regional melalui titik akhir cloud privat virtual (VPC) gateway. Tidak dikenakan biaya tambahan untuk menggunakan titik akhir gateway. Untuk mempelajari selengkapnya tentang titik akhir API Regional dan Zonal, lihat Jaringan untuk ember direktori.

Penghapusan data

Anda dapat menghapus satu atau beberapa objek langsung dari bucket direktori Anda dengan menggunakan konsol Amazon S3,,AWS CLI() AWS SDKs AWS Command Line Interface , atau Amazon S3 REST API. Karena semua objek dalam direktori bucket Anda dikenakan biaya penyimpanan, sebaiknya hapus objek yang tidak lagi diperlukan.

Menghapus objek yang disimpan dalam sebuah bucket direktori juga secara rekursif menghapus setiap direktori induk, jika direktori induk tersebut tidak berisi objek lain selain objek yang sedang dihapus.

> **Note**
>
> Penghapusan autentikasi multi-faktor (MFA) dan Penentuan Versi S3 tidak didukung untuk S3 Express One Zone.

Mengautentikasi dan mengotorisasi permintaan

Secara default, bucket direktori bersifat pribadi dan hanya dapat diakses oleh pengguna yang secara eksplisit diberikan akses. Batas kontrol akses untuk bucket direktori hanya diatur pada tingkat bucket. Sebaliknya, batas kontrol akses untuk bucket tujuan umum dapat diatur pada tingkat bucket, prefiks, atau tag objek. Perbedaan ini berarti bucket direktori adalah satu-satunya sumber daya yang dapat Anda sertakan dalam kebijakan bucket atau kebijakan identitas IAM untuk akses S3 Express One Zone.

Amazon S3 Express One Zone mendukung otorisasi AWS Identity and Access Management (AWS IAM) dan otorisasi berbasis sesi:

- Untuk menggunakan operasi API titik akhir Regional (operasi tingkat ember, atau bidang kontrol) dengan S3 Express One Zone, Anda menggunakan model otorisasi IAM, yang tidak melibatkan manajemen sesi. Izin diberikan untuk tindakan yang dilakukan secara individual. Untuk informasi selengkapnya, lihat Mengotorisasi operasi API endpoint Regional dengan IAM.
- Untuk menggunakan operasi API titik akhir Zonal (operasi tingkat objek, atau bidang data), kecuali untuk `CopyObject` dan`HeadBucket`, Anda menggunakan operasi `CreateSession` API untuk membuat dan mengelola sesi yang dioptimalkan untuk otorisasi permintaan data latensi rendah. Untuk mengambil dan menggunakan token sesi, Anda harus mengizinkan tindakan `s3express:CreateSession` untuk bucket direktori Anda dalam kebijakan berbasis identitas atau kebijakan bucket. Untuk informasi selengkapnya, lihat Mengotorisasi operasi API endpoint

Regional dengan IAM. Jika Anda mengakses S3 Express One Zone di konsol Amazon S3, melalui AWS Command Line Interface AWS CLI(), atau dengan menggunakan, S3 Express One Zone membuat sesi AWS SDKs atas nama Anda.

Dengan operasi `CreateSession` API, Anda mengautentikasi dan mengotorisasi permintaan melalui mekanisme berbasis sesi baru. Anda dapat menggunakan `CreateSession` untuk meminta kredensial sementara yang menyediakan akses latensi rendah ke bucket Anda. Kredensial sementara ini dicakup ke bucket direktori tertentu.

Untuk bekerja dengan`CreateSession`, kami sarankan menggunakan versi terbaru dari AWS SDKs atau menggunakan AWS Command Line Interface (AWS CLI). Pembentukan sesi yang didukung AWS SDKs dan AWS CLI menangani, penyegaran, dan penghentian atas nama Anda.

Anda menggunakan token sesi dengan operasi Zona (tingkat objek) saja (kecuali untuk `CopyObject` dan`HeadBucket`) untuk mendistribusikan latensi yang terkait dengan otorisasi atas sejumlah permintaan dalam sesi. Untuk operasi API titik akhir Regional (operasi tingkat bucket), Anda menggunakan otorisasi IAM, yang tidak melibatkan pengelolaan sesi. Untuk informasi selengkapnya, lihat Mengotorisasi operasi API endpoint Regional dengan IAM dan Mengotorisasi operasi API titik akhir Zonal dengan `CreateSession`.

Bagaimana operasi API diautentikasi dan diotorisasi

Tabel berikut mencantumkan informasi autentikasi dan otorisasi untuk operasi API bucket direktori. Untuk setiap operasi API, tabel menunjukkan nama operasi API, tindakan kebijakan IAM, tipe titik akhir (Regional atau Zonal), dan mekanisme otorisasi (IAM atau berbasis sesi). Tabel ini juga menunjukkan apakah akses lintas akun didukung. Akses ke tindakan tingkat bucket dapat diberikan hanya dalam kebijakan berbasis identitas IAM (pengguna atau peran), bukan kebijakan bucket.

API	Jenis titik akhir	Tindakan IAM	Akses lintas akun
`CreateBucket`	Regional	`s3express:CreateBucket`	Tidak
`DeleteBucket`	Regional	`s3express:DeleteBucket`	Tidak
`ListDirectoryBuckets`	Regional	`s3express:ListAllMyDirectoryBuckets`	Tidak

API	Jenis titik akhir	Tindakan IAM	Akses lintas akun
PutBucketPolicy	Regional	s3express:PutBucketPolicy	Tidak
GetBucketPolicy	Regional	s3express:GetBucketPolicy	Tidak
DeleteBucketPolicy	Regional	s3express:DeleteBucketPolicy	Tidak
CreateSession	Zona	s3express:CreateSession	Ya
CopyObject	Zona	s3express:CreateSession	Ya
DeleteObject	Zona	s3express:CreateSession	Ya
DeleteObjects	Zona	s3express:CreateSession	Ya
HeadObject	Zona	s3express:CreateSession	Ya
PutObject	Zona	s3express:CreateSession	Ya
RenameObject	Zona	s3express:CreateSession	Tidak
GetObjectAttributes	Zona	s3express:CreateSession	Ya
ListObjectsV2	Zona	s3express:CreateSession	Ya
HeadBucket	Zona	s3express:CreateSession	Ya
CreateMultipartUpload	Zona	s3express:CreateSession	Ya
UploadPart	Zona	s3express:CreateSession	Ya
UploadPartCopy	Zona	s3express:CreateSession	Ya

API	Jenis titik akhir	Tindakan IAM	Akses lintas akun
CompleteMultipartUpload	Zona	s3express:CreateSession	Ya
AbortMultipartUpload	Zona	s3express:CreateSession	Ya
ListParts	Zona	s3express:CreateSession	Ya
ListMultipartUploads	Zona	s3express:CreateSession	Ya
ListAccessPointsForDirectoryBuckets	Zona	s3express:ListAccessPointsForDirectoryBuckets	Ya
GetAccessPointScope	Zona	s3express:GetAccessPointScope	Ya
PutAccessPointScope	Zona	s3express:PutAccessPointScope	Ya
DeleteAccessPointScope	Zona	s3express:DeleteAccessPointScope	Ya

Topik

- Mengotorisasi operasi API endpoint Regional dengan IAM
- Mengotorisasi operasi API titik akhir Zonal dengan CreateSession

Mengotorisasi operasi API endpoint Regional dengan IAM

AWS Identity and Access Management (IAM) adalah sebuah Layanan AWS yang membantu administrator mengontrol akses ke sumber daya dengan aman. AWS Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diotorisasi (memiliki izin) untuk menggunakan sumber daya Amazon S3 dalam bucket direktori dan operasi S3 Express One Zone. Anda dapat menggunakan IAM tanpa biaya tambahan.

Secara default, pengguna tidak memiliki izin untuk bucket direktori. Untuk memberikan izin akses bagi bucket direktori, Anda dapat menggunakan IAM untuk membuat pengguna, grup, atau peran dan melampirkan izin ke identitas tersebut. Untuk informasi selengkapnya tentang IAM, lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.

Untuk memberikan akses, Anda dapat menambahkan izin ke pengguna, grup, atau peran Anda melalui cara berikut:

- Pengguna dan grup di AWS IAM Identity Center - Buat set izin. Ikuti instruksi di Buat rangkaian izin di Panduan Pengguna AWS IAM Identity Center .
- Pengguna yang dikelola di IAM melalui penyedia identitas-Buat peran untuk federasi identitas. Ikuti instruksi dalam Membuat peran untuk penyedia identitas pihak ketiga (federasi) di Panduan Pengguna IAM.
- Peran dan pengguna IAM–Buat peran yang dapat diambil oleh pengguna Anda. Ikuti instruksi dalam Membuat peran untuk mendelegasikan izin ke pengguna IAM di Panduan Pengguna IAM.

Untuk informasi tentang pengaturan IAM untuk S3 Express One Zone, lihat topik berikut.

Topik

- Pengguna utama
- Sumber daya
- Tindakan untuk ember direktori
- Kebijakan berbasis identitas IAM untuk bucket direktori
- Contoh kebijakan bucket untuk bucket direktori

Pengguna utama

Saat membuat kebijakan berbasis sumber daya untuk memberikan akses ke bucket, Anda harus menggunakan elemen `Principal` tersebut untuk menentukan orang atau aplikasi yang dapat

membuat permintaan tindakan atau operasi pada sumber daya tersebut. Untuk kebijakan bucket direktori, Anda dapat menggunakan pengguna utama berikut:

- AWS Akun
- Pengguna IAM
- Peran IAM
- Pengguna gabungan

Untuk informasi selengkapnya, lihat Principal dalam Panduan Pengguna IAM.

Sumber daya

Amazon Resource Names (ARNs) untuk bucket direktori berisi `s3express` namespace, ID AWS akun, dan nama bucket direktori, yang menyertakan ID Zona. Wilayah AWS AWS (Availability Zone atau Local Zone ID).

Untuk mengakses dan melakukan tindakan pada bucket direktori Anda, Anda harus menggunakan format ARN berikut ini:

```
arn:aws:s3express:region:account-id:bucket/base-bucket-name--zone-id--x-s3
```

Untuk mengakses dan melakukan tindakan pada titik akses Anda untuk bucket direktori, Anda harus menggunakan format ARN berikut:

```
arn:aws::s3express:region:account-id:accesspoint/accesspoint-basename--zone-id--xa-s3
```

Untuk informasi selengkapnya ARNs, lihat Amazon Resource Names (ARNs)di Panduan Pengguna IAM. Untuk informasi selengkapnya tentang sumber daya, lihat Elemen Kebijakan JSON IAM: Resource dalam Panduan Pengguna IAM.

Tindakan untuk ember direktori

Dalam kebijakan berbasis identitas atau kebijakan berbasis sumber daya IAM, Anda menentukan tindakan S3 yang diizinkan atau ditolak. Tindakan sesuai dengan operasi API tertentu. Dengan bucket direktori, Anda harus menggunakan namespace S3 Express One Zone untuk memberikan izin, yang disebut. `s3express`

Saat Anda mengizinkan `s3express:CreateSession` izin, operasi `CreateSession` API mengambil token sesi sementara untuk semua operasi Zonal endpoint API (level objek). Token

sesi mengembalikan kredensil yang digunakan untuk semua operasi API titik akhir Zonal lainnya. Akibatnya, Anda tidak memberikan izin akses ke operasi API Zonal dengan kebijakan IAM. Sebagai gantinya, `CreateSession` aktifkan akses untuk semua operasi tingkat objek. Untuk daftar operasi dan izin API Zonal, lihat Mengautentikasi dan mengotorisasi permintaan.

Untuk mempelajari lebih lanjut tentang operasi API `CreateSession`, lihat CreateSession di Referensi API Amazon Simple Storage Service.

Anda dapat menyebutkan tindakan berikut dalam elemen `Action` pernyataan kebijakan IAM. Gunakan kebijakan untuk memberikan izin untuk melaksanakan operasi dalam AWS. Saat Anda menggunakan tindakan dalam kebijakan, Anda biasanya mengizinkan atau menolak akses ke operasi API dengan nama yang sama. Namun, dalam beberapa kasus, satu tindakan mengendalikan akses ke lebih dari satu operasi API. Akses ke tindakan tingkat bucket dapat diberikan hanya dalam kebijakan berbasis identitas IAM (pengguna atau peran) dan bukan kebijakan bucket.

Untuk informasi selengkapnya tentang cara mengonfigurasi kebijakan titik akses, lihatMengkonfigurasi kebijakan IAM untuk menggunakan titik akses untuk bucket direktori.

Untuk informasi selengkapnya, lihat Kunci tindakan, sumber daya, dan kondisi untuk Amazon S3 Express.

Kebijakan berbasis identitas IAM untuk bucket direktori

Sebelum Anda dapat membuat bucket direktori, Anda harus memberikan izin yang diperlukan untuk peran AWS Identity and Access Management (IAM) atau pengguna Anda. Kebijakan contoh ini memungkinkan akses ke operasi API `CreateSession` (untuk digunakan dengan operasi API titik akhir [tingkat objek] Zona) dan semua operasi API titik akhir Regional (tingkat bucket). Kebijakan ini memungkinkan operasi API `CreateSession` untuk digunakan dengan semua bucket direktori, tetapi operasi API titik akhir Regional hanya diizinkan untuk digunakan dengan bucket direktori yang ditentukan. Untuk menggunakan kebijakan contoh ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
{
     "Version": "2012-10-17",
     "Statement": [
         {
             "Sid": "AllowAccessRegionalEndpointAPIs",
             "Effect": "Allow",
             "Action": [
```

```
                "s3express:DeleteBucket",
                "s3express:DeleteBucketPolicy",
                "s3express:CreateBucket",
                "s3express:PutBucketPolicy",
                "s3express:GetBucketPolicy",
                "s3express:ListAllMyDirectoryBuckets"
            ],

            "Resource": "arn:aws:s3express:region:account_id:bucket/bucket-base-
name--zone-id--x-s3/*"
        },
        {
            "Sid": "AllowCreateSession",
            "Effect": "Allow",
            "Action": "s3express:CreateSession",
            "Resource": "*"
        }
    ]
}
```

Contoh kebijakan bucket untuk bucket direktori

Bagian ini memberikan contoh kebijakan bucket direktori. Untuk menggunakan kebijakan ini, ganti *user input placeholders* dengan informasi Anda sendiri.

Contoh kebijakan bucket berikut memungkinkan Akun AWS ID *111122223333* untuk menggunakan operasi `CreateSession` API dengan `ReadWrite` sesi default untuk bucket direktori yang ditentukan. Kebijakan ini memberikan akses ke operasi API titik akhir Zona (tingkat objek).

Example – Kebijakan bucket untuk mengizinkan panggilan **`CreateSession`** dengan sesi **`ReadWrite`** default

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ReadWriteAccess",
            "Effect": "Allow",
            "Resource": "arn:aws:s3express:us-west-2:account-id:bucket/bucket-base-
name--zone-id--x-s3",
            "Principal": {
```

```
                "AWS": [
                    "111122223333"
                ]
            },
            "Action": [
                "s3express:CreateSession"
            ]
        }
    ]
}
```

Example – Kebijakan bucket untuk mengizinkan panggilan **`CreateSession`** dengan sesi **`ReadOnly`**

Contoh kebijakan bucket berikut memungkinkan Akun AWS ID *111122223333* untuk menggunakan operasi `CreateSession` API. Kebijakan ini menggunakan kunci syarat `s3express:SessionMode` dengan nilai `ReadOnly` untuk mengatur sesi hanya baca.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ReadOnlyAccess",
            "Effect": "Allow",
            "Principal": {
                "AWS": "111122223333"
            },
            "Action": "s3express:CreateSession",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "s3express:SessionMode": "ReadOnly"
                }
            }
        }
    ]
}
```

Example – Kebijakan bucket untuk memungkinkan akses lintas akun bagi panggilan **`CreateSession`**

Contoh kebijakan bucket berikut *111122223333* memungkinkan Akun AWS ID menggunakan operasi `CreateSession` API untuk bucket direktori tertentu yang dimiliki oleh Akun AWS ID*444455556666*.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CrossAccount",
            "Effect": "Allow",
            "Principal": {
                "AWS": "111122223333"
            },
            "Action": [
                "s3express:CreateSession"
            ],
            "Resource": "arn:aws:s3express:us-west-2:444455556666:bucket/bucket-base-name--zone-id--x-s3"

        }
    ]
}
```

Mengotorisasi operasi API titik akhir Zonal dengan **`CreateSession`**

Untuk menggunakan operasi API titik akhir Zonal (operasi tingkat objek, atau bidang data), kecuali untuk `CopyObject` dan`HeadBucket`, Anda menggunakan operasi `CreateSession` API untuk membuat dan mengelola sesi yang dioptimalkan untuk otorisasi permintaan data latensi rendah. Untuk mengambil dan menggunakan token sesi, Anda harus mengizinkan tindakan `s3express:CreateSession` untuk bucket direktori Anda dalam kebijakan berbasis identitas atau kebijakan bucket. Untuk informasi selengkapnya, lihat Mengotorisasi operasi API endpoint Regional dengan IAM. Jika Anda mengakses S3 Express One Zone di konsol Amazon S3, melalui AWS Command Line Interface AWS CLI(), atau dengan menggunakan, S3 Express One Zone membuat sesi AWS SDKs atas nama Anda.

Jika Anda menggunakan API REST Amazon S3, Anda dapat menggunakan operasi API `CreateSession` untuk mendapatkan kredensial keamanan sementara yang menyertakan ID kunci akses, kunci akses rahasia, token sesi, dan waktu kedaluwarsa. Kredensial sementara memberikan izin yang sama seperti kredensial keamanan jangka panjang, seperti kredensial pengguna IAM, tetapi kredensial keamanan sementara harus menyertakan token sesi.

Mode Sesi

Mode sesi mendefinisikan ruang lingkup sesi. Dalam kebijakan bucket, Anda dapat menentukan kunci kondisi `s3express:SessionMode` untuk mengontrol siapa yang dapat membuat sesi `ReadWrite` atau `ReadOnly`. Untuk informasi selengkapnya tentang sesi `ReadWrite` atau `ReadOnly`, lihat parameter `x-amz-create-session-mode` untuk CreateSession di Referensi API Amazon S3. Untuk informasi selengkapnya tentang kebijakan bucket yang akan dibuat, lihatContoh kebijakan bucket untuk bucket direktori.

Token Sesi

Saat Anda melakukan panggilan dengan menggunakan kredensial keamanan sementara, panggilan harus menyertakan token sesi. Token sesi dikembalikan bersama dengan kredensial sementara. Token sesi dibatasi pada bucket direktori Anda dan digunakan untuk memverifikasi bahwa kredensial keamanan valid dan belum kedaluwarsa. Untuk melindungi sesi Anda, kredensial keamanan sementara akan kedaluwarsa setelah 5 menit.

`CopyObject` dan `HeadBucket`

Kredensial keamanan sementara dibatasi pada bucket direktori tertentu dan secara otomatis diaktifkan untuk semua panggilan API operasi Zonal (tingkat objek) ke bucket direktori yang diberikan. Tidak seperti operasi API titik akhir Zonal lainnya, `CopyObject` dan `HeadBucket` tidak menggunakan otentikasi `CreateSession`. Semua permintaan `CopyObject` dan `HeadBucket` harus diautentikasi dan ditandatangani dengan menggunakan kredensial IAM. Namun, `CopyObject` dan masih `HeadBucket` diotorisasi oleh `s3express:CreateSession`, seperti operasi API titik akhir Zonal lainnya.

Untuk informasi selengkapnya, lihat CreateSession dalam Referensi API Amazon Simple Storage Service.

Praktik terbaik keamanan untuk bucket direktori

Ada sejumlah fitur keamanan yang perlu dipertimbangkan saat bekerja dengan bucket direktori. Praktik terbaik berikut adalah pedoman umum dan tidak mewakili solusi keamanan yang lengkap.

Karena praktik terbaik ini mungkin tidak sesuai atau tidak memadai untuk lingkungan Anda, perlakukan itu sebagai rekomendasi yang bermanfaat, bukan sebagai resep.

Pengaturan Blokir Akses Publik dan Kepemilikan Objek Default

Bucket direktori mendukung Blokir Akses Publik S3 dan Kepemilikan Objek S3. Fitur S3 ini digunakan untuk mengaudit dan mengelola akses ke bucket dan objek Anda.

Secara default, semua pengaturan Blok Akses Publik untuk bucket direktori diaktifkan. Selain itu, Kepemilikan Objek disetel ke pemilik bucket yang diberlakukan, yang berarti bahwa daftar kontrol akses (ACLs) dinonaktifkan. Pengaturan ini tidak dapat dimodifikasi. Untuk informasi selengkapnya tentang fitur ini, lihat the section called “Memblokir akses publik” dan the section called “Mengontrol kepemilikan objek”.

> **Note**
>
> Anda tidak dapat memberikan akses ke objek yang disimpan dalam bucket direktori. Anda hanya dapat memberikan akses ke bucket direktori Anda. Model otorisasi untuk S3 Express One Zone berbeda dari model otorisasi untuk Amazon S3. Untuk informasi selengkapnya, lihat Mengotorisasi operasi API titik akhir Zonal dengan `CreateSession`.

Autentikasi dan otorisasi

Mekanisme otentikasi dan otorisasi untuk bucket direktori berbeda-beda, tergantung apakah Anda membuat permintaan ke operasi API titik akhir Zonal atau operasi API titik akhir Regional. Operasi API Zona adalah operasi tingkat objek (bidang data). Operasi API regional adalah operasi tingkat bucket (bidang kontrol).

Anda mengautentikasi dan mengotorisasi permintaan ke operasi API titik akhir Zonal melalui mekanisme berbasis sesi baru yang dioptimalkan untuk memberikan latensi terendah. Dengan autentikasi berbasis sesi, AWS SDKs gunakan operasi `CreateSession` API untuk meminta kredensil sementara yang menyediakan akses latensi rendah ke bucket direktori Anda. Kredensial sementara ini dicakup ke bucket direktori tertentu dan kedaluwarsa setelah 5 menit. Anda dapat menggunakan kredensial sementara ini untuk menandatangani panggilan API Zona (tingkat objek). Untuk informasi selengkapnya, lihat Mengotorisasi operasi API titik akhir Zonal dengan `CreateSession`.

Menandatangani permintaan dengan kredensil untuk manajemen bucket direktori

Anda menggunakan kredensi Anda untuk menandatangani permintaan API titik akhir Zonal (level objek) dengan AWS Signature Version 4, dengan nama layanan`s3express`. Saat Anda menandatangani permintaan, gunakan kunci rahasia yang dikembalikan dari `CreateSession` dan berikan juga token sesi dengan `x-amzn-s3session-token header`. Untuk informasi selengkapnya, lihat CreateSession.

Kredensi AWS SDKs kelola yang didukung dan penandatanganan atas nama Anda. Sebaiknya gunakan AWS SDKs untuk menyegarkan kredensil dan menandatangani permintaan untuk Anda.

Menandatangani permintaan dengan kredensial IAM

Semua panggilan API Regional (tingkat bucket) harus diautentikasi dan ditandatangani oleh kredensial AWS Identity and Access Management (IAM), bukan kredensial sesi sementara. Kredensial IAM terdiri atas ID kunci akses dan kunci akses rahasia untuk identitas IAM. Semua permintaan `CopyObject` dan `HeadBucket` juga harus diautentikasi dan ditandatangani menggunakan kredensial IAM.

Untuk mencapai latensi terendah untuk panggilan operasi Zonal (tingkat objek) Anda, sebaiknya gunakan kredensyal yang diperoleh dari panggilan `CreateSession` untuk menandatangani permintaan Anda, kecuali untuk permintaan ke dan. `CopyObject` `HeadBucket`

Gunakan AWS CloudTrail

AWS CloudTrail menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau Layanan AWS di Amazon S3. Anda dapat menggunakan informasi yang dikumpulkan oleh CloudTrail untuk menentukan hal-hal berikut:

- Permintaan yang diajukan ke Amazon S3
- Alamat IP dari mana permintaan itu dibuat
- Siapa yang membuat permintaan
- Kapan permintaan dibuat
- Detail tambahan tentang permintaan

Saat Anda mengatur Akun AWS, acara CloudTrail manajemen diaktifkan secara default. Operasi API titik akhir Regional berikut (tingkat ember, atau bidang kontrol, operasi API) dicatat. CloudTrail

- CreateBucket

- DeleteBucket
- DeleteBucketPolicy
- PutBucketPolicy
- GetBucketPolicy
- ListDirectoryBuckets
- ListMultipartUploads
- PutBucketEncryption
- GetBucketEncryption
- DeleteBucketEncryption

> **Note**
>
> `ListMultipartUploads`adalah operasi API titik akhir Zonal. Namun, itu dicatat CloudTrail sebagai acara manajemen. Untuk informasi selengkapnya, lihat ListMultipartUploads dalam Referensi API Amazon Simple Storage Service.

Secara default, CloudTrail jejak tidak mencatat peristiwa data, tetapi Anda dapat mengonfigurasi jejak untuk mencatat peristiwa data untuk bucket direktori yang Anda tentukan, atau untuk mencatat peristiwa data untuk semua bucket direktori di akun Anda. AWS Operasi API titik akhir Zonal berikut (tingkat objek, atau bidang data, operasi API) dicatat. CloudTrail

- AbortMultipartUpload
- CompleteMultipartUpload
- CreateSession
- CopyObject
- CreateMultipartUpload
- DeleteObject
- DeleteObjects
- GetObject
- GetObjectAttributes
- HeadBucket

- HeadObject
- ListObjectsV2
- ListParts
- PutObject
- UploadPart
- UploadPartCopy

Untuk informasi selengkapnya tentang penggunaan AWS CloudTrail dengan bucket direktori, lihat Logging with AWS CloudTrail for directory bucket.

Melaksanakan pemantauan dengan menggunakan alat AWS pemantauan

Pemantauan adalah bagian penting dalam menjaga keandalan, keamanan, ketersediaan, dan kinerja Amazon S3 dan solusi Anda AWS . AWS menyediakan beberapa alat dan layanan untuk membantu Anda memantau Amazon S3 dan yang lain. Layanan AWS Misalnya, Anda dapat memantau CloudWatch metrik Amazon untuk Amazon S3, khususnya metrik penyimpanan `NumberOfObjects` dan `BucketSizeBytes` penyimpanan.

Objek yang disimpan dalam bucket direktori tidak akan tercermin dalam metrik `NumberOfObjects` penyimpanan `BucketSizeBytes` dan untuk Amazon S3. Namun, metrik `BucketSizeBytes` dan `NumberOfObjects` penyimpanan didukung untuk bucket direktori. Untuk melihat metrik pilihan Anda, Anda dapat membedakan antara kelas penyimpanan Amazon S3 dengan menentukan dimensi. `StorageType` Untuk informasi selengkapnya, lihat Memantau metrik dengan Amazon CloudWatch.

Untuk informasi selengkapnya, lihat Memantau metrik dengan Amazon CloudWatch dan Pencatatan log dan pemantauan di Amazon S3.

Mengelola akses ke kumpulan data bersama dalam bucket direktori dengan titik akses

Poin Akses Amazon S3 menyederhanakan pengelolaan akses data dalam skala besar untuk kumpulan data bersama di Amazon S3. Titik akses adalah nama host unik yang Anda buat untuk menegakkan izin dan kontrol jaringan yang berbeda untuk semua permintaan yang dibuat melalui titik akses. Anda dapat membuat ratusan titik akses per bucket, masing-masing dengan nama dan

izin berbeda yang disesuaikan untuk setiap aplikasi. Setiap titik akses berfungsi bersama dengan kebijakan bucket yang dilampirkan ke bucket yang mendasarinya.

Dalam bucket direktori, nama titik akses terdiri dari nama dasar yang Anda berikan, diikuti dengan ID Zona (AWS Availability Zone atau Local Zone) dari lokasi bucket direktori Anda, dan kemudian`--xa-s3`. Misalnya, *`accesspointname`*`--`*`zoneID`*`--xa-s3`. Setelah Anda membuat titik akses, Anda tidak dapat mengubah nama atau ID Zona.

Dengan titik akses untuk bucket direktori, Anda dapat menggunakan cakupan titik akses untuk membatasi akses ke awalan atau operasi API tertentu. Anda dapat menentukan jumlah awalan, tetapi total panjang karakter dari semua awalan harus kurang dari 256 byte.

Anda dapat mengonfigurasi titik akses apa pun untuk menerima permintaan hanya dari cloud pribadi virtual (VPC). Ini membatasi akses data Amazon S3 ke jaringan pribadi.

Di bagian ini, topik menjelaskan cara menggunakan titik akses untuk bucket direktori. Untuk informasi tentang bucket direktori, lihatBekerja dengan bucket direktori.

Topik

- Titik akses untuk aturan penamaan bucket direktori, pembatasan, dan batasan
- Mereferensikan titik akses untuk bucket direktori
- Operasi objek untuk titik akses untuk bucket direktori
- Mengkonfigurasi kebijakan IAM untuk menggunakan titik akses untuk bucket direktori
- Memantau dan mencatat titik akses untuk bucket direktori
- Membuat titik akses untuk ember direktori
- Mengelola titik akses Anda untuk bucket direktori

Titik akses untuk aturan penamaan bucket direktori, pembatasan, dan batasan

Titik akses menyederhanakan pengelolaan akses data dalam skala besar untuk set data bersama di Amazon S3. Topik berikut memberikan informasi tentang aturan dan batasan dan batasan penamaan titik akses.

Topik

- Aturan penamaan untuk titik akses untuk bucket direktori
- Pembatasan dan batasan untuk titik akses untuk bucket direktori

Aturan penamaan untuk titik akses untuk bucket direktori

Titik akses harus dibuat di zona yang sama dengan tempat bucket berada. Nama titik akses harus unik di dalam zona tersebut.

Nama titik akses harus sesuai dengan DNS dan harus memenuhi ketentuan berikut:

- Harus dimulai dengan angka atau huruf kecil
- Nama dasar yang Anda berikan harus memiliki panjang antara 3 dan 50 karakter
- Tidak dapat diawali atau diakhiri dengan tanda hubung (`-`)
- Tidak dapat berisi garis bawah (`_`), huruf besar, spasi, atau titik (`.`)
- Harus diakhiri dengan sufiks*`zoneid`*`--xa--s3`.

Pembatasan dan batasan untuk titik akses untuk bucket direktori

Titik akses untuk bucket direktori memiliki batasan dan batasan berikut:

- Setiap titik akses dikaitkan dengan satu bucket direktori. Setelah membuat jalur akses, Anda tidak dapat mengaitkannya ke bucket yang berbeda. Namun, Anda dapat menghapus titik akses, lalu membuat yang baru dengan nama yang sama dan mengaitkannya ke bucket yang berbeda.
- Setelah Anda membuat titik akses, Anda tidak dapat mengubah konfigurasi cloud privat virtual (VPC).
- Ukuran kebijakan titik akses dibatasi hingga 20 KB.
- Awalan cakupan titik akses dibatasi hingga 256 byte dalam ukuran total.
- Anda dapat membuat maksimum 10.000 titik akses per Akun AWS per Wilayah AWS. Jika Anda memerlukan lebih dari 10.000 titik akses untuk satu akun di satu Wilayah, Anda dapat meminta peningkatan kuota layanan. Untuk informasi lebih lanjut tentang kuota layanan dan permintaan peningkatan, lihat AWS Service Quotas dalam Referensi Umum AWS.
- Anda hanya dapat menggunakan titik akses untuk melakukan pengoperasian pada objek. Anda tidak dapat menggunakan titik akses untuk menjalankan operasi bucket Amazon S3, seperti memodifikasi atau menghapus bucket. Untuk daftar lengkap operasi yang didukung, lihatOperasi objek untuk titik akses untuk bucket direktori.
- Anda dapat merujuk ke titik akses berdasarkan nama, alias titik akses, atau virtual-hosted-style URI. Anda tidak dapat menangani titik akses oleh ARN. Untuk informasi selengkapnya, lihat Mereferensikan titik akses untuk bucket direktori.

- Operasi API yang mengontrol fungsionalitas titik akses (misalnya, `PutAccessPointPolicy` dan`GetAccessPointPolicy`) harus menentukan AWS akun yang memiliki titik akses.
- Anda harus menggunakan AWS Signature Version 4 saat membuat permintaan ke titik akses dengan menggunakan REST API. Untuk informasi selengkapnya tentang mengautentikasi permintaan, lihat Mengautentikasi Permintaan (Versi AWS Tanda Tangan 4) di Referensi API Amazon Simple Storage Service.
- Titik akses hanya mendukung permintaan melalui HTTPS. Amazon S3 akan secara otomatis merespons dengan pengalihan HTTP untuk setiap permintaan yang dibuat melalui HTTP, untuk meningkatkan permintaan ke HTTPS.
- Titik akses tidak mendukung akses anonim.
- Jika Anda membuat jalur akses ke bucket yang dimiliki oleh akun lain (titik akses lintas akun), jalur akses lintas akun tidak memberi Anda akses ke data hingga pemilik bucket memberi Anda izin untuk mengakses bucket. Pemilik bucket selalu memegang kendali penuh atas akses ke data, dan harus memperbarui kebijakan bucket untuk mengotorisasi permintaan dari titik akses lintas akun. Untuk melihat contoh kebijakan bucket, lihat Mengkonfigurasi kebijakan IAM untuk menggunakan titik akses untuk bucket direktori.

Mereferensikan titik akses untuk bucket direktori

Setelah Anda membuat titik akses, Anda dapat menggunakannya sebagai titik akhir untuk membentuk operasi objek. Untuk titik akses untuk bucket direktori, alias titik akses sama dengan nama titik akses. Anda dapat menggunakan nama titik akses alih-alih nama bucket untuk semua operasi data. Untuk daftar operasi yang didukung ini, lihatOperasi objek untuk titik akses untuk bucket direktori.

Mengacu pada titik akses oleh virtual-hosted-style URIs

Titik akses hanya mendukung virtual-host-style pengalamatan. Titik akses menggunakan format yang sama dengan titik akhir bucket direktori. Untuk informasi selengkapnya, lihat Titik akhir regional dan Zonal untuk bucket direktori.

Jalur akses S3 tidak mendukung akses melalui HTTP. Jalur akses hanya mendukung akses aman melalui HTTPS.

Operasi objek untuk titik akses untuk bucket direktori

Anda dapat menggunakan titik akses untuk mengakses objek menggunakan operasi S3 data berikut.

- AbortMultipartUpload
- CompleteMultipartUpload
- CopyObject
- CreateMultipartUpload
- CreateSession
- DeleteObject
- DeleteObjects
- GetObject
- GetObjectAttributes
- HeadBucket
- HeadObject
- ListMultipartUploads
- ListObjectsV2
- ListParts
- PutObject
- UploadPart
- UploadPartCopy

Mengkonfigurasi kebijakan IAM untuk menggunakan titik akses untuk bucket direktori

Kebijakan sumber daya dukungan titik akses AWS Identity and Access Management (IAM) yang memungkinkan Anda mengontrol penggunaan titik akses berdasarkan sumber daya, pengguna, atau kondisi lainnya. Agar aplikasi atau pengguna dapat mengakses objek melalui titik akses, jalur akses dan kebijakan bucket yang mendasarinya harus mengizinkan permintaan tersebut.

> **Important**
>
> Menambahkan titik akses ke bucket direktori tidak mengubah perilaku bucket saat bucket diakses langsung melalui nama bucket. Semua operasi yang ada terhadap bucket akan terus bekerja seperti sebelumnya. Pembatasan yang Anda sertakan dalam kebijakan jalur akses atau cakupan titik akses hanya berlaku untuk permintaan yang dibuat melalui jalur akses tersebut.

Saat menggunakan kebijakan sumber daya IAM, pastikan untuk menyelesaikan peringatan keamanan, kesalahan, peringatan umum, dan saran AWS Identity and Access Management Access Analyzer sebelum Anda menyimpan kebijakan Anda. Penganalisis Akses IAM menjalankan pemeriksaan kebijakan untuk memvalidasi kebijakan Anda terhadap tata bahasa kebijakan IAM dan praktik terbaik. Pemeriksaan ini menghasilkan temuan dan memberikan rekomendasi untuk membantu Anda membuat kebijakan yang berfungsi dan sesuai dengan praktik terbaik keamanan.

Untuk mempelajari validasi kebijakan menggunakan Penganalisis Akses IAM lebih lanjut, lihat Memvalidasi kebijakan Penganalisis Akses IAM di Panduan Pengguna IAM. Untuk melihat daftar peringatan, kesalahan, dan saran yang ditampilkan oleh Penganalisis Akses IAM, lihat referensi pemeriksaan kebijakan Penganalisis Akses IAM.

Titik akses untuk contoh kebijakan bucket direktori

Kebijakan titik akses berikut menunjukkan cara mengontrol permintaan ke bucket direktori. Kebijakan titik akses memerlukan bucket ARNs atau access point ARNs. Alias titik akses tidak didukung dalam kebijakan. Berikut ini adalah contoh dari titik akses ARN:

```
arn:aws:s3express:region:account-id:accesspoint/myaccesspoint--zoneID--xa-s3
```

Anda dapat melihat titik akses ARN dalam detail titik akses. Untuk informasi selengkapnya, lihat Lihat detail untuk titik akses Anda untuk bucket direktori.

> **Note**
>
> Izin yang diberikan dalam kebijakan titik akses hanya berlaku jika bucket yang mendasarinya juga mengizinkan akses yang sama. Anda dapat melakukan ini menggunakan dua cara:
>
> 1. (Disarankan) Delegasikan kontrol akses dari bucket ke titik akses seperti yang dijelaskan dalam Mendelegasikan kontrol akses ke titik akses.
> 2. Tambahkan izin yang sama yang tercantum dalam kebijakan titik akses ke kebijakan bucket dasar.

Example 1 — Kebijakan kontrol layanan untuk membatasi titik akses ke asal jaringan VPC

Kebijakan kontrol layanan berikut mengharuskan semua titik akses baru dibuat dengan asal jaringan virtual private cloud (VPC). Dengan kebijakan ini, pengguna di organisasi Anda tidak dapat membuat jalur akses apa pun yang dapat diakses dari internet.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Effect": "Deny",
        "Action": "s3express:CreateAccessPoint",
        "Resource": "*",
        "Condition": {
            "StringNotEquals": {
                "s3express:AccessPointNetworkOrigin": "VPC"
            }
        }
    }
  ]
}
```

Example 2 — Kebijakan titik akses untuk membatasi akses bucket ke titik akses dengan asal jaringan VPC

Kebijakan titik akses berikut membatasi semua akses ke bucket *amzn-s3-demo-bucket--zoneID--x-s3* ke titik akses dengan asal jaringan VPC.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Principal": "*",
            "Action": "s3express:CreateSession",
            "Effect": "Deny",
            "Resource": "arn:aws:s3express:region:111122223333:bucket/amzn-s3-demo-
bucket--zoneID--x-s3",
            "Condition": {
                "StringNotEqualsIfExists": {
                    "s3express:AccessPointNetworkOrigin": "VPC"
                }
            }
```

```
        }
    ]
}
```

Kunci syarat

Titik akses untuk bucket direktori memiliki kunci kondisi yang dapat Anda gunakan dalam kebijakan IAM untuk mengontrol akses ke sumber daya Anda. Kunci kondisi berikut hanya mewakili bagian dari kebijakan IAM. Untuk contoh kebijakan lengkap, lihat Titik akses untuk contoh kebijakan bucket direktori, Mendelegasikan kontrol akses ke titik akses, dan Memberikan izin untuk titik akses lintas akun.

`s3express:DataAccessPointArn`

Contoh ini menunjukkan cara memfilter akses dengan nama sumber daya Amazon (ARN) dari titik akses dan mencocokkan semua titik akses untuk Akun AWS *111122223333* di Wilayah: *region*

```
"Condition" : {
    "StringLike": {
        "s3express:DataAccessPointArn":
 "arn:aws:s3express:region:111122223333:accesspoint/*"
    }
}
```

`s3express:DataAccessPointAccount`

Contoh ini menunjukkan operator string yang dapat Anda gunakan untuk mencocokkan ID akun pemilik titik akses. Contoh berikut cocok dengan semua titik akses yang dimiliki oleh Akun AWS *111122223333*.

```
"Condition" : {
    "StringEquals": {
        "s3express:DataAccessPointAccount": "111122223333"
    }
}
```

`s3express:AccessPointNetworkOrigin`

Contoh ini menunjukkan operator string yang dapat Anda gunakan untuk mencocokkan asal jaringan, baik `Internet` atau VPC. Contoh berikut hanya mencocokkan titik akses dengan asal VPC.

```
"Condition" : {
    "StringEquals": {
        "s3express:AccessPointNetworkOrigin": "VPC"
    }
}
```

`s3express:Permissions`

Anda dapat menggunakan `s3express:Permissions` untuk membatasi akses ke operasi API tertentu dalam cakupan titik akses. Operasi API berikut didukung:

- `PutObject`
- `GetObject`
- `DeleteObject`
- `ListBucket`(diperlukan untuk`ListObjectsV2`)
- `GetObjectAttributes`
- `AbortMultipartUpload`
- `ListBucketMultipartUploads`
- `ListMultipartUploadParts`

> **Note**
>
> Saat menggunakan kunci kondisi multi-nilai, kami sarankan Anda menggunakan `ForAllValues` dengan `Allow` pernyataan dan `ForAnyValue` dengan `Deny` pernyataan. Untuk informasi selengkapnya, lihat Kunci konteks Multivalued di Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang menggunakan kunci kondisi dengan Amazon S3, lihat Tindakan, sumber daya, dan kunci kondisi untuk Amazon S3 di Referensi Otorisasi Layanan.

Untuk informasi selengkapnya tentang izin yang diperlukan untuk operasi S3 API menurut jenis sumber daya S3, lihat. Izin yang diperlukan untuk operasi API Amazon S3

Mendelegasikan kontrol akses ke titik akses

Anda dapat mendelegasikan kontrol akses dari kebijakan bucket ke kebijakan titik akses. Kebijakan ini memungkinkan akses penuh ke semua titik akses yang dimiliki oleh akun pemilik bucket. Setelah

menerapkan kebijakan, semua akses ke bucket ini dikendalikan oleh kebijakan titik akses. Kami menyarankan Anda untuk mengonfigurasi bucket dengan cara ini untuk semua kasus penggunaan yang tidak memerlukan akses langsung ke bucket.

Example kebijakan bucket yang mendelegasikan kontrol akses ke titik akses

```
{
    "Version": "2012-10-17",
    "Statement" : [
    {
        "Effect": "Allow",
        "Principal" : { "AWS": "*" },
        "Action" : "*",
        "Resource" : [ "Bucket ARN",
        "Condition": {
            "StringEquals" : { "s3express:DataAccessPointAccount" : "Bucket owner's
 account ID" }
        }
    }]
}
```

Memberikan izin untuk titik akses lintas akun

Untuk membuat titik akses ke bucket yang dimiliki oleh akun lain, Anda harus terlebih dahulu membuat titik akses dengan menentukan nama bucket dan ID pemilik akun. Kemudian, pemilik bucket harus memperbarui kebijakan bucket untuk mengotorisasi permintaan dari titik akses. Membuat titik akses mirip dengan membuat DNS CNAME di mana titik akses tidak menyediakan akses ke konten bucket. Semua akses bucket dikendalikan oleh kebijakan bucket. Contoh kebijakan bucket berikut mengizinkan `GET` dan `LIST` meminta pada bucket dari titik akses yang dimiliki oleh Akun AWS yang tepercaya.

Ganti *`Bucket ARN`* dengan ARN ember.

Example kebijakan bucket yang mendelegasikan izin ke yang lain Akun AWS

```
{
    "Version": "2012-10-17",
    "Statement" : [
    {
        "Effect": "Allow",
        "Principal" : { "AWS": "*" },
```

```
        "Action" : "s3express:CreateSession",
        "Resource" : [ "Bucket ARN" ],
        "Condition": {
            "StringEquals" : {
                "s3express:DataAccessPointAccount": "Access point owner's account ID"
            },
            "ForAllValues:StringEquals": {
                "s3express:Permissions": [
                    "GetObject",
                    "ListBucket"
                ]
            }
        }
    }]
}
```

Memantau dan mencatat titik akses untuk bucket direktori

Anda dapat mencatat permintaan yang dibuat melalui titik akses dan permintaan yang dibuat untuk APIs yang mengelola titik akses, seperti `CreateAccessPoint` dan `GetAccessPointPolicy`, dengan menggunakan AWS CloudTrail. CloudTrail entri log untuk permintaan yang dibuat melalui titik akses termasuk titik akses ARN (yang mencakup nama titik akses) di `resources` bagian log.

Sebagai contoh, anggap Anda memiliki konfigurasi berikut:

- Sebuah bucket bernama *amzn-s3-demo-bucket*`--`*zone-id*`--x-s3` di Region `region` yang berisi objek bernama`my-image.jpg`.
- Titik akses bernama `my-bucket-ap--zoneID--xa-s3` yang terkait dengan *amzn-s3-demo-bucket*`--`*zone-id*`--x-s3`
- Akun AWS ID dari `123456789012`

Contoh berikut menunjukkan `resources` bagian dari entri CloudTrail log untuk konfigurasi sebelumnya:

```
"resources": [
        {"type": "AWS::S3Express::Object",

            "ARN": "arn:aws:s3express-region:123456789012:bucket/amzn-s3-demo-
bucket--zone-id--x-s3/my-image.jpg"
```

```
        },
        {"accountId": "c",
            "type": "AWS::S3Express::DirectoryBucket",
            "ARN": "arn:aws::s3express:region:123456789012:bucket/amzn-s3-demo-
bucket--zone-id--x-s3"
        },
        {"accountId": "123456789012",
            "type": "AWS::S3::AccessPoint",
            "ARN": "arn:aws:s3express:region:123456789012:accesspoint/my-bucket-ap--
zoneID--xa-s3"
        }
    ]
```

Untuk informasi lebih lanjut tentang AWS CloudTrail, lihat Apa itu AWS CloudTrail? dalam AWS CloudTrail User Guide.

Membuat titik akses untuk ember direktori

Seperti bucket Direktori, titik akses dapat dibuat di Availability Zones atau di Dedicated Local Zones. Titik akses harus dibuat di zona yang sama dengan bucket direktori yang terkait dengannya.

Titik akses dikaitkan dengan tepat satu bucket direktori Amazon S3. Jika Anda ingin menggunakan bucket direktori di dalam Akun AWS, Anda harus terlebih dahulu membuat bucket direktori. Untuk informasi selengkapnya tentang membuat bucket direktori, lihat Membuat bucket direktori di Availability Zone atauMembuat bucket direktori di Zona Lokal.

Anda juga dapat membuat titik akses lintas akun yang terkait dengan bucket di bucket lain Akun AWS, selama Anda mengetahui nama bucket dan ID akun pemilik bucket. Namun, membuat titik akses lintas akun tidak memberikan Anda akses ke data di bucket sampai Anda diberikan izin dari pemilik bucket. Pemilik bucket harus memberikan akses akun pemilik titik akses (akun Anda) ke bucket melalui kebijakan bucket. Untuk informasi selengkapnya, lihat Memberikan izin untuk titik akses lintas akun.

Anda dapat membuat titik akses untuk bucket direktori apa pun dengan AWS Management Console, AWS CLI, REST API, atau AWS SDKs. Setiap titik akses dikaitkan dengan satu bucket direktori, dan Anda dapat membuat ratusan titik akses per bucket. Saat membuat titik akses, Anda memilih nama titik akses dan keranjang direktori untuk mengaitkannya. Nama titik akses terdiri dari nama dasar yang Anda berikan dan akhiran yang menyertakan ID Zona lokasi bucket Anda, diikuti oleh`--xa-s3`. Misalnya,*myaccesspoint*`-`*zoneID*`--xa-s3`. Anda juga dapat membatasi akses ke titik akses melalui Virtual Private Cloud (VPC). Kemudian, Anda dapat segera mulai membaca dan menulis data

melalui jalur akses Anda dengan menggunakan namanya, sama seperti Anda menggunakan nama bucket direktori.

Anda dapat menggunakan cakupan titik akses untuk membatasi akses ke bucket direktori melalui titik akses ke awalan atau operasi API tertentu. Jika Anda tidak menambahkan cakupan ke titik akses, semua awalan di bucket direktori dan semua operasi API dapat dilakukan pada objek di bucket saat diakses melalui titik akses. Setelah membuat titik akses, Anda dapat menambahkan, memodifikasi, atau menghapus cakupan menggunakan AWS CLI, AWS SDKs, atau REST API. Untuk informasi selengkapnya, lihat Kelola ruang lingkup titik akses Anda untuk bucket direktori.

Setelah Anda membuat titik akses, Anda dapat mengonfigurasi kebijakan sumber daya IAM titik akses Anda. Untuk informasi selengkapnya, lihat Melihat, mengedit, atau menghapus kebijakan titik akses.

Menggunakan konsol S3

> **Note**
>
> Anda juga dapat membuat titik akses untuk bucket direktori dari layar bucket direktori. Saat Anda melakukan ini, nama bucket direktori disediakan dan Anda tidak perlu memilih bucket saat membuat titik akses. Untuk informasi selengkapnya, lihat Mendaftar bucket direktori.

Untuk membuat titik akses untuk bucket direktori

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi di bagian atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah tempat Anda ingin membuat titik akses. Titik akses harus dibuat di Wilayah yang sama dengan bucket terkait.
3. Di panel navigasi kiri, pilih Titik akses untuk ember direktori.
4. Pada halaman Titik Akses, pilih Buat titik akses.
5. Anda dapat membuat titik akses untuk bucket direktori di akun Anda atau di akun lain. Untuk membuat titik akses untuk bucket direktori di akun lain:

> **Note**
>
> Jika Anda menggunakan bucket di bucket lain Akun AWS, pemilik bucket harus memperbarui kebijakan bucket untuk mengotorisasi permintaan dari titik akses. Untuk contoh kebijakan bucket, lihat Memberikan izin untuk titik akses lintas akun.

a. Di bidang Ember direktori, pilih Tentukan bucket di akun lain.
b. Di bidang ID akun pemilik Bucket, masukkan Akun AWS ID yang memiliki bucket.
c. Di bidang Nama Bucket, masukkan nama bucket, termasuk nama dasar dan ID zona. Misalnya, ***bucket-base-name*--*zone-id*--x-s3**.

6. Untuk membuat titik akses untuk bucket direktori di akun Anda:

a. Di bidang Ember direktori, pilih Pilih ember di akun ini.
b. Di bidang Nama Bucket, masukkan nama bucket, termasuk nama dasar dan ID zona. Misalnya, ***bucket-base-name*--*zone-id*--x-s3**. Untuk memilih bucket dari daftar, pilih Browse S3 dan pilih bucket direktori.

7. Di Nama titik akses, di bidang Nama dasar, masukkan nama dasar untuk titik akses. ID zona dan nama titik akses penuh muncul. Untuk informasi lebih lanjut tentang penamaan titik akses, lihat Aturan penamaan untuk titik akses untuk bucket direktori.
8. Di Network Origin, pilih virtual private cloud (VPC) atau Internet. Jika Anda memilih virtual private cloud (VPC), di bidang ID VPC, masukkan ID VPC yang ingin Anda gunakan dengan titik akses.
9. (Opsional) Dalam cakupan titik akses, untuk menerapkan cakupan ke titik akses ini, pilih Batasi cakupan titik akses ini menggunakan awalan atau izin.

a. Untuk membatasi akses ke awalan di bucket direktori, di Prefiks, masukkan satu atau beberapa awalan. Untuk menambahkan awalan lain, pilih Tambah awalan. Untuk menghapus awalan, pilih Hapus.

> **Note**
>
> Cakupan titik akses memiliki batas karakter 512 karakter total untuk semua awalan. Anda dapat melihat jumlah karakter yang tersisa di bawah Tambahkan awalan.

b. Dalam Izin, pilih satu atau beberapa operasi API yang akan diizinkan oleh jalur akses. Untuk menghapus operasi data, pilih X di sebelah nama operasi data.

10. Untuk tidak menerapkan cakupan ke titik akses dan mengizinkan akses ke semua awalan di bucket direktori dan semua operasi API melalui titik akses, dalam cakupan titik akses, pilih Terapkan akses ke seluruh bucket.
11. Pilih Buat titik akses untuk bucket direktori. Nama titik akses dan informasi lain tentangnya muncul di daftar Access points for directory bucket.

Menggunakan AWS CLI

Contoh berikut membuat Titik Akses Lambda Objek yang diberi nama *example-ap* untuk bucket *amzn-s3-demo-bucket--zone-id--x-s3* di akun *111122223333*.

```
aws s3control create-access-point --name example-ap--zoneID--xa-s3 --account-
id 111122223333 --bucket amzn-s3-demo-bucket--zone-id--x-s3
```

Untuk membatasi akses ke titik akses melalui VPC, sertakan parameter dan `--vpc` ID VPC.

```
aws s3control create-access-point --name example-ap--zoneID--xa-s3 --account-
id 111122223333 --bucket amzn-s3-demo-bucket--zone-id--x-s3 --vpc vpc-id
```

Saat Anda membuat titik akses untuk bucket lintas akun, sertakan `--bucket-account-id` parameternya. Contoh perintah berikut membuat titik akses di Akun AWS *111122223333*, untuk bucket*amzn-s3-demo-bucket--zone-id--x-s3*, yang dimiliki oleh Akun AWS *444455556666*.

```
aws s3control create-access-point --name example-ap--zoneID--xa-s3 --account-
id 111122223333 --bucket amzn-s3-demo-bucket--zone-id--x-s3 --bucket-account-
id 444455556666
```

Untuk informasi dan contoh selengkapnya, lihat create-access-pointdi AWS CLI Command Reference.

Penggunaan API REST

Contoh perintah berikut membuat titik akses yang diberi nama *example-ap* untuk bucket *amzn-s3-demo-bucket--zone-id--x-s3* di akun *111122223333* dan akses dibatasi melalui VPC *vpc-id* (opsional).

```
PUT /v20180820/accesspoint/example-ap--zoneID--xa-s3 HTTP/1.1
Host: s3express-control.region.amazonaws.com
```

```
x-amz-account-id: 111122223333
<?xml version="1.0" encoding="UTF-8"?>
<CreateAccessPointRequest>
   <Bucket>amzn-s3-demo-bucket--zone-id--x-s3s</Bucket>
   <BucketAccountId>111122223333</BucketAccountId>
   <VpcConfiguration>
       <VpcId>vpc-id</VpcId>
   </VpcConfiguration>
</CreateAccessPointRequest>
```

Respons:

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<CreateAccessPointResult>
   <AccessPointArn>
       "arn:aws:s3express:region:111122223333:accesspoint/example-ap--zoneID--xa-s3"
   </AccessPointArn>
   <Alias>example-ap--zoneID--xa-s3</Alias>
</CreateAccessPointResult>
```

Menggunakan AWS SDKs

Anda dapat menggunakan AWS SDKs untuk membuat titik akses. Untuk informasi selengkapnya, lihat daftar yang didukung SDKs di Referensi API Amazon Simple Storage Service.

Mengelola titik akses Anda untuk bucket direktori

Bagian ini menjelaskan cara mengelola titik akses Anda untuk bucket direktori menggunakan AWS Command Line Interface, Amazon S3 REST API, AWS atau SDK.

Topik

- Buat daftar titik akses Anda untuk bucket direktori
- Lihat detail untuk titik akses Anda untuk bucket direktori
- Melihat, mengedit, atau menghapus kebijakan titik akses
- Kelola ruang lingkup titik akses Anda untuk bucket direktori
- Hapus titik akses Anda untuk bucket direktori

Buat daftar titik akses Anda untuk bucket direktori

Bagian ini menjelaskan cara mencantumkan titik akses untuk bucket direktori menggunakan AWS Management Console, AWS Command Line Interface (AWS CLI), REST API, atau AWS SDKs.

Menggunakan konsol S3

Untuk membuat daftar titik akses di Akun AWS

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi di bagian atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah yang ingin Anda daftarkan titik akses.
3. Di panel navigasi di sisi kiri konsol, pilih Titik akses untuk bucket direktori.
4. (Opsional) Cari titik akses berdasarkan nama. Hanya titik akses yang Anda pilih yang Wilayah AWS akan muncul di sini.
5. Pilih nama titik akses yang ingin Anda kelola.

Menggunakan AWS CLI

`list-access-points-for-directory-buckets`Contoh perintah berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk daftar titik akses yang dimiliki oleh AWS akun dan terkait dengan bucket direktori.

Perintah berikut mencantumkan titik akses Akun AWS *111122223333* yang dilampirkan ke bucket*amzn-s3-demo-bucket--zone-id--x-s3*.

```
aws s3control list-access-points-for-directory-buckets --account-id 111122223333 --
directory-bucket amzn-s3-demo-bucket--zone-id--x-s3
```

Untuk informasi dan contoh selengkapnya, lihat list-access-points-for-directory-buckets di Referensi Perintah. AWS CLI

Penggunaan API REST

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan REST API untuk membuat daftar titik akses Anda.

```
GET /v20180820/directoryaccesspoint?directoryBucket=amzn-s3-demo-bucket--zone-id--x-s3
&maxResults=maxResults HTTP/1.1
Host: s3express-control.region.amazonaws.com
x-amz-account-id: 111122223333
```

Example **`ListAccessPointsForDirectoryBuckets`**respon

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<ListDirectoryAccessPointsResult>
    <AccessPointList>
        <AccessPoint>
            <AccessPointArn>arn:aws:s3express:region:111122223333:accesspoint/example-
access-point--zoneID--xa-s3</AccessPointArn>
            <Alias>example-access-point--zoneID--xa-s3</Alias>
            <Bucket>amzn-s3-demo-bucket--zone-id--x-s3</Bucket>
            <BucketAccountId>111122223333</BucketAccountId>
            <Name>example-access-point--zoneID--xa-s3</Name>
            <NetworkOrigin>VPC</NetworkOrigin>
            <VpcConfiguration>
                <VpcId>VPC-1</VpcId>
            </VpcConfiguration>
        </AccessPoint>
    </AccessPointList>
</ListDirectoryAccessPointsResult>
```

Menggunakan AWS SDKs

Anda dapat menggunakan AWS SDKs untuk daftar titik akses Anda. Untuk informasi selengkapnya, lihat daftar yang didukung SDKs di Referensi API Amazon Simple Storage Service.

Lihat detail untuk titik akses Anda untuk bucket direktori

Bagian ini menjelaskan cara melihat detail titik akses Anda untuk bucket direktori menggunakan AWS Management Console,, AWS CLI AWS SDKs, atau REST API.

Menggunakan konsol S3

Lihat detail titik akses untuk bucket direktori untuk melihat informasi berikut tentang titik akses dan bucket direktori terkait:

- Properti:
  - Nama ember direktori
  - ID akun pemilik bucket direktori
  - Wilayah AWS
  - Jenis lokasi bucket direktori
  - Nama lokasi bucket direktori
  - Tanggal pembuatan titik akses
  - Asal jaringan
  - VPC ID
  - URI S3
  - Titik Akses ARN
  - Alias titik akses
- Izin:
  - Temuan penganalisis akses eksternal IAM
  - Ruang lingkup titik akses
  - Kebijakan titik akses

Untuk melihat detail titik akses Anda di Akun AWS

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi di bagian atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah yang ingin Anda daftarkan titik akses.
3. Di panel navigasi di sisi kiri konsol, pilih Titik akses untuk bucket direktori.
4. (Opsional) Cari titik akses berdasarkan nama. Hanya titik akses yang Anda pilih yang Wilayah AWS akan muncul di sini.
5. Pilih nama titik akses yang ingin Anda kelola.
6. Pilih tab Properties atau tab Izin.

Menggunakan AWS CLI

`get-access-point`Contoh perintah berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk melihat rincian untuk titik akses Anda.

Perintah berikut mencantumkan rincian untuk titik akses *my-access-point--zoneID--xa-s3* untuk Akun AWS *111122223333*.

```
aws s3control get-access-point --name my-access-point--zoneID--xa-s3 --account-
id 111122223333
```

Example output dari **get-access-point** perintah

```
{
    "Name": "example-access-point--zoneID--xa-s3",
    "Bucket": "amzn-s3-demo-bucket--zone-id--x-s3",
    "NetworkOrigin": "Internet",
    "PublicAccessBlockConfiguration": {
        "BlockPublicAcls": true,
        "IgnorePublicAcls": true,
        "BlockPublicPolicy": true,
        "RestrictPublicBuckets": true
    },
    "CreationDate": "2025-04-23T18:26:22.146000+00:00",
    "Alias": "example-access-point--zoneID--xa-s3",
    "AccessPointArn": "arn:aws:s3express:region:111122223333:accesspoint/example-
access-point--zoneID--xa-s3",
    "BucketAccountId": "296805379465"
}
```

Untuk informasi dan contoh selengkapnya, lihat get-access-point di AWS CLI Referensi Perintah.

Penggunaan API REST

Anda dapat menggunakan REST API untuk melihat detail titik akses Anda. Untuk informasi selengkapnya, lihat GetAccessPoint dalam Referensi API Amazon Simple Storage Service.

Menggunakan AWS SDKs

Anda dapat menggunakan AWS SDKs untuk melihat detail titik akses Anda. Untuk informasi selengkapnya, lihat daftar yang didukung SDKs di Referensi API Amazon Simple Storage Service.

Melihat, mengedit, atau menghapus kebijakan titik akses

Anda dapat menggunakan kebijakan titik akses AWS Identity and Access Management (IAM) untuk mengontrol prinsipal dan sumber daya yang dapat mengakses titik akses. Cakupan titik akses

mengelola awalan dan izin API untuk titik akses. Anda dapat membuat, mengedit, dan menghapus kebijakan titik akses menggunakan AWS Command Line Interface, REST API, atau AWS SDKs. Untuk informasi selengkapnya tentang cakupan titik akses, lihatKelola ruang lingkup titik akses Anda untuk bucket direktori.

> **Note**
>
> Karena bucket direktori menggunakan otorisasi berbasis sesi, kebijakan Anda harus selalu menyertakan tindakan. `s3express:CreateSession`

Menggunakan konsol S3

Untuk melihat, mengedit, atau menghapus kebijakan titik akses

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi di bagian atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah yang ingin Anda daftarkan titik akses.
3. Di panel navigasi di sisi kiri konsol, pilih Titik akses untuk bucket direktori.
4. (Opsional) Cari titik akses berdasarkan nama. Hanya titik akses yang Anda pilih yang Wilayah AWS akan muncul di sini.
5. Pilih nama titik akses yang ingin Anda kelola.
6. Pilih tab Izin.
7. Untuk membuat atau mengedit kebijakan titik akses, dalam kebijakan Titik akses, pilih Edit. Edit kebijakan. Pilih Simpan.
8. Untuk menghapus kebijakan titik akses, dalam Kebijakan titik akses, pilih Hapus. Di jendela Hapus kebijakan titik akses, ketik **`confirm`** dan pilih Hapus.

Menggunakan AWS CLI

Anda dapat menggunakan `delete-access-point-policy` perintah `get-acccess-point-policyput-access-point-policy`,, dan untuk melihat, mengedit, atau menghapus kebijakan titik akses. Untuk informasi selengkapnya, lihat get-access-point-policyput-access-point-policy, atau delete-access-point-policydi Referensi AWS CLI Perintah.

Penggunaan API REST

Anda dapat menggunakan REST API`GetAccessPointPolicy`,`DeleteAccessPointPolicy`, dan `PutAccessPointPolicy` operasi untuk melihat, menghapus, atau mengedit kebijakan titik akses. Untuk informasi selengkapnya, lihat PutAccessPointPolicyGetAccessPointPolicy, atau DeleteAccessPointPolicydi Referensi API Amazon Simple Storage Service.

Menggunakan AWS SDKs

Anda dapat menggunakan kebijakan AWS SDKs untuk melihat, menghapus, atau mengedit jalur akses. Untuk informasi selengkapnya, lihat daftar yang didukung SDKs untuk GetAccessControlPolicy, DeleteAccessControlPolicy, dan PutAccessControlPolicydi Referensi API Layanan Penyimpanan Sederhana Amazon.

Kelola ruang lingkup titik akses Anda untuk bucket direktori

Bagian ini menjelaskan cara melihat dan memodifikasi cakupan titik akses Anda untuk bucket direktori menggunakan AWS Command Line Interface, REST API, atau AWS SDKs. Anda dapat menggunakan cakupan titik akses untuk membatasi akses ke awalan atau operasi API tertentu.

Topik

- Lihat cakupan titik akses Anda untuk bucket direktori
- Ubah cakupan titik akses Anda untuk bucket direktori
- Hapus cakupan titik akses Anda untuk bucket direktori

Lihat cakupan titik akses Anda untuk bucket direktori

Anda dapat menggunakan AWS Management Console, AWS Command Line Interface, REST API, atau AWS SDKs untuk melihat cakupan titik akses Anda untuk bucket direktori.

Menggunakan konsol S3

Menggunakan AWS CLI

`get-access-point-scope`Contoh perintah berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk melihat ruang lingkup titik akses Anda.

Perintah berikut menunjukkan ruang lingkup titik akses *`my-access-point`* -- *`zoneID`* --xa-s3 untuk. Akun AWS *`111122223333`*

```
aws s3control get-access-point-scope --name my-access-point--zoneID--xa-s3 --account-
id 111122223333
```

Untuk informasi dan contoh selengkapnya, lihat get-access-point-scopedi AWS CLI Command Reference.

Example hasil dari **`get-access-point-scope`**

```
{
    "Scope": {
        "Permissions": [
            "ListBucket",
            "PutObject"
        ]
  "Prefixes": [
            "Prefix": "MyPrefix1*",
            "Prefix": "MyObjectName.csv"
        ]
    }
}
```

Penggunaan API REST

`GetAccessPointScope`Contoh permintaan berikut menunjukkan bagaimana Anda dapat menggunakan REST API untuk melihat cakupan titik akses Anda.

Permintaan berikut menunjukkan ruang lingkup titik akses *`my-access-point`* -- *`region`* - *`zoneID`* --xa-s3 untuk. Akun AWS *`111122223333`*

```
GET /v20180820/accesspoint/my-access-point--zoneID--xa-s3/scope HTTP/1.1
Host: s3express-control.region.amazonaws.com
x-amz-account-id: 111122223333
```

Example hasil dari **`GetAccessPointScope`**

```
      HTTP/1.1 200
```

```
<?xml version="1.0" encoding="UTF-8"?>
<GetAccessPointScopeResult>
    <Scope>
        <Prefixes>
            <Prefix>MyPrefix1*</Prefix>
            <Prefix>MyObjectName.csv</Prefix>
        </Prefixes>
        <Permissions>
            <Permission>ListBucket</Permission>
            <Permission>PutObject</Permission>
        </Permissions>
        <Scope>
</GetAccessPointScopeResult>
```

Menggunakan AWS SDKs

Anda dapat menggunakan AWS SDKs untuk melihat ruang lingkup titik akses Anda. Untuk informasi selengkapnya, lihat daftar yang didukung SDKs di Referensi API Amazon Simple Storage Service.

Ubah cakupan titik akses Anda untuk bucket direktori

Anda dapat menggunakan AWS Command Line Interface, REST API, atau AWS SDKs untuk memodifikasi cakupan titik akses Anda untuk bucket direktori. Cakupan titik akses digunakan untuk membatasi akses ke awalan tertentu, operasi API, atau kombinasi keduanya.

Anda dapat menyertakan satu atau beberapa operasi API berikut sebagai izin:

- `PutObject`
- `GetObject`
- `DeleteObject`
- `ListBucket`(diperlukan untuk`ListObjectsV2`)
- `GetObjectAttributes`
- `AbortMultipartUploads`
- `ListBucketMultipartUploads`
- `ListMultipartUploadParts`

> **Note**
>
> - Anda dapat menentukan jumlah awalan, tetapi panjang total karakter dari semua awalan harus berukuran kurang dari 256 byte.
> - Saat Anda memodifikasi cakupan titik akses, Anda mengganti cakupan yang ada.

Menggunakan AWS CLI

`put-access-point-scope`Contoh perintah berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk memodifikasi ruang lingkup titik akses Anda.

Perintah berikut memodifikasi cakupan titik akses *`my-access-point`* -- *`zoneID`* --xa-s3 untuk. Akun AWS *`111122223333`*

> **Note**
>
> Anda dapat menggunakan wildcard dalam awalan dengan menggunakan karakter asterisk (*). Jika Anda ingin menggunakan karakter tanda bintang sebagai literal, tambahkan karakter garis miring terbalik (\) sebelum menghindarinya.
> Juga, semua awalan memiliki akhiran '*' implisit, yang berarti semua jalur dengan awalan akan disertakan.

```
aws s3control put-access-point-scope --name my-access-point--zoneID--xa-s3 --account-
id 111122223333 --scope Prefixes=string,Permissions=string
```

Untuk informasi dan contoh selengkapnya, lihat put-access-point-scopedi AWS CLI Command Reference.

Penggunaan API REST

`PutAccessPointScope`Contoh permintaan berikut menunjukkan bagaimana Anda dapat menggunakan REST API untuk memodifikasi cakupan titik akses Anda.

Permintaan berikut memodifikasi cakupan titik akses *`my-access-point`* -- *`zoneID`* --xa-s3 untuk. Akun AWS *`111122223333`*

> **Note**
>
> Anda dapat menggunakan wildcard dalam awalan dengan menggunakan karakter asterisk (*). Jika Anda ingin menggunakan karakter tanda bintang sebagai literal, tambahkan karakter garis miring terbalik (\) sebelum menghindarinya.
> Juga, semua awalan memiliki akhiran '*' implisit, yang berarti semua jalur dengan awalan akan disertakan.

```
PUT /v20180820/accesspoint/my-access-point--zoneID--xa-s3/scope HTTP/1.1
Host: s3express-control.region.amazonaws.com
x-amz-account-id: 111122223333
<?xml version="1.0" encoding="UTF-8"?>
<PutAccessPointScopeRequest>
        <Scope>
            <Prefixes>
                <Prefix>Jane/*</Prefix>
            </Prefixes>
            <Permissions>
                <Permission>PutObject</Permission>
                <Permission>GetObject</Permission>
            </Permissions>
            <Scope>
    </PutAccessPointScopeRequest>
```

Menggunakan AWS SDKs

Anda dapat menggunakan AWS CLI, AWS SDKs, atau REST API untuk memodifikasi cakupan titik akses Anda. Untuk informasi selengkapnya, lihat daftar yang didukung SDKs di Referensi API Amazon Simple Storage Service.

Hapus cakupan titik akses Anda untuk bucket direktori

Anda dapat menggunakan AWS Command Line Interface, REST API, atau AWS SDKs untuk menghapus cakupan titik akses Anda untuk bucket direktori.

> **Note**
>
> Saat Anda menghapus cakupan titik akses, semua awalan dan izin dihapus.

Menggunakan AWS CLI

`delete-access-point-scope`Contoh perintah berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk menghapus ruang lingkup titik akses Anda.

Perintah berikut menghapus ruang lingkup titik akses *my-access-point* -- *zoneID* --xa-s3 untuk. Akun AWS *111122223333*

```
aws s3control delete-access-point-scope --name my-access-point--region-zoneID--xa-s3 --
account-id 111122223333
```

Untuk informasi dan contoh selengkapnya, lihat delete-access-point-scopedi AWS CLI Command Reference.

Penggunaan API REST

Permintaan berikut menghapus cakupan titik akses *my-access-point* -- *zoneID* --xa-s3 untuk. Akun AWS *111122223333*

```
DELETE /v20180820/accesspoint/my-access-point--zoneID--xa-s3/scope HTTP/1.1
Host: s3express-control.region.amazonaws.com
x-amz-account-id: 111122223333
```

Menggunakan AWS SDKs

Anda dapat menggunakan AWS SDKs untuk menghapus ruang lingkup jalur akses Anda. Untuk informasi selengkapnya, lihat daftar yang didukung SDKs di Referensi API Amazon Simple Storage Service.

Hapus titik akses Anda untuk bucket direktori

Bagian ini menjelaskan cara menghapus jalur akses Anda menggunakan AWS Management Console, AWS Command Line Interface, REST API, atau AWS SDKs.

> **Note**
>
> Sebelum Anda dapat menghapus ember direktori yang dilampirkan ke titik akses, Anda harus menghapus titik akses.

Menggunakan konsol S3

Untuk menghapus titik akses untuk bucket direktori di Akun AWS

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi di bagian atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah yang ingin Anda daftarkan titik akses.
3. Di panel navigasi di sisi kiri konsol, pilih Titik akses untuk bucket direktori.
4. (Opsional) Cari titik akses berdasarkan nama. Hanya titik akses yang Anda pilih yang Wilayah AWS akan muncul di sini.
5. Pilih nama titik akses yang ingin Anda hapus.
6. Pilih Hapus.
7. Untuk mengonfirmasi penghapusan, ketik **`confirm`** dan pilih Hapus.

Menggunakan AWS CLI

`delete-access-point`Contoh perintah berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk menghapus titik akses Anda.

Perintah berikut menghapus titik akses *`my-access-point`* -- *`zoneID`* --xa-s3 untuk. Akun AWS *`111122223333`*

```
aws s3control delete-access-point --name my-access-point--zoneID--xa-s3 --account-
id 111122223333
```

Untuk informasi dan contoh selengkapnya, lihat delete-access-point di AWS CLI Referensi Perintah.

Penggunaan API REST

Anda dapat menggunakan REST API untuk menghapus jalur akses Anda. Untuk informasi selengkapnya, lihat DeleteAccessPoint dalam Referensi API Amazon Simple Storage Service.

Menggunakan AWS SDKs

Anda dapat menggunakan AWS SDKs untuk menghapus titik akses Anda. Untuk informasi selengkapnya, lihat daftar yang didukung SDKs di Referensi API Amazon Simple Storage Service.

Masuk dengan AWS CloudTrail untuk ember direktori

Amazon S3 terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau layanan. AWS CloudTrail menangkap semua panggilan API untuk Amazon S3 sebagai peristiwa. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat ke Amazon S3, alamat IP dari mana permintaan dibuat, kapan dibuat, dan detail tambahan. Saat aktivitas peristiwa yang didukung terjadi di Amazon S3, aktivitas tersebut direkam dalam suatu CloudTrail peristiwa. Anda dapat menggunakan AWS CloudTrail jejak untuk mencatat peristiwa manajemen dan peristiwa data untuk bucket direktori. Untuk informasi selengkapnya, lihat CloudTrail acara Amazon S3 dan Apa itu? AWS CloudTrail dalam AWS CloudTrail User Guide.

CloudTrail acara manajemen untuk ember direktori

Secara default, CloudTrail mencatat tindakan tingkat ember untuk bucket direktori sebagai peristiwa manajemen. Acara `eventsource` untuk CloudTrail manajemen untuk bucket direktori adalah`s3express.amazonaws.com`. Saat Anda mengatur AWS akun, acara CloudTrail manajemen diaktifkan secara default. Operasi API titik akhir Regional berikut (tingkat ember, atau bidang kontrol, operasi API) dicatat. CloudTrail

- `CreateBucket`
- `DeleteBucket`
- `DeleteBucketPolicy`
- `PutBucketPolicy`
- `GetBucketPolicy`
- `ListDirectoryBuckets`
- `ListMultipartUploads`
- `GetBucketEncryption`
- `PutBucketEncryption`
- `DeleteBucketEncryption`

> **Note**
>
> `ListMultipartUploads`adalah operasi API titik akhir Zonal. Namun, operasi API ini dicatat CloudTrail sebagai peristiwa manajemen. Untuk informasi selengkapnya, lihat `ListMultipartUploads` dalam Referensi API Amazon Simple Storage Service.

Untuk informasi selengkapnya tentang peristiwa CloudTrail manajemen, lihat peristiwa manajemen log di Panduan AWS CloudTrail Pengguna.

CloudTrail peristiwa data untuk ember direktori

Peristiwa data memberikan informasi tentang operasi sumber daya yang dilakukan pada atau di sumber daya (misalnya, membaca atau menulis ke objek Amazon S3). Ini juga dikenal sebagai operasi bidang data. Peristiwa data seringkali merupakan aktivitas volume tinggi. Secara default, CloudTrail jejak tidak mencatat peristiwa data, tetapi Anda dapat mengonfigurasi jejak untuk mencatat peristiwa data untuk objek yang disimpan dalam bucket tujuan umum dan bucket direktori. Untuk informasi selengkapnya, lihat Mengaktifkan logging untuk objek dalam bucket menggunakan konsol.

Saat Anda mencatat peristiwa data untuk jejak CloudTrail, Anda dapat memilih untuk menggunakan pemilih acara lanjutan atau pemilih acara dasar. Untuk mencatat peristiwa data untuk objek yang disimpan dalam bucket direktori, Anda harus menggunakan pemilih peristiwa lanjutan. Saat mengonfigurasi pemilih sumber daya tingkat lanjut, Anda akan memilih atau menentukan jenis sumber daya yang ada. `AWS::S3Express::Object`

Operasi API titik akhir Zonal berikut (tingkat objek, atau. bidang data, operasi API) dicatat. CloudTrail

- `AbortMultipartUpload`
- `CompleteMultipartUpload`
- `CreateSession`
- `CopyObject`
- `CreateMultipartUpload`
- `DeleteObject`
- `DeleteObjects`
- `GetObject`
- `GetObjectAttributes`

- HeadBucket
- HeadObject
- ListObjectsV2
- ListParts
- PutObject
- RenameObject
- UploadPart
- UploadPartCopy

Untuk informasi selengkapnya tentang peristiwa CloudTrail data, lihat Mencatat peristiwa data di Panduan AWS CloudTrail Pengguna.

Untuk informasi tambahan tentang CloudTrail peristiwa untuk bucket direktori, lihat topik berikut:

Topik

- CloudTrail contoh file log untuk ember direktori

CloudTrail contoh file log untuk ember direktori

File CloudTrail log mencakup informasi tentang operasi API yang diminta, tanggal dan waktu operasi, parameter permintaan, dan sebagainya. Topik ini menampilkan contoh peristiwa CloudTrail data dan peristiwa manajemen untuk bucket direktori.

Topik

- CloudTrail contoh file log peristiwa data untuk ember direktori

CloudTrail contoh file log peristiwa data untuk ember direktori

Contoh berikut menunjukkan contoh file CloudTrail log yang menunjukkan. CreateSession

```
{
    "eventVersion": "1.09",
    "userIdentity": {
      "type": "AssumedRole",
      "principalId": "AROAIDPPEZS35WEXAMPLE:AssumedRoleSessionName",
      "arn": "arn:aws:sts::111122223333assumed-role/RoleToBeAssumed/MySessionName",
```

```
          "accountId": "111122223333",
          "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
          "sessionContext": {
            "sessionIssuer": {
              "type": "Role",
              "principalId": "AROAIDPPEZS35WEXAMPLE",
              "arn": "arn:aws:iam::111122223333:role/RoleToBeAssumed",
              "accountId": "111122223333",
              "userName":"RoleToBeAssumed
            },

            "attributes": {
              "creationDate": "2024-07-02T00:21:16Z",
            "mfaAuthenticated": "false"
            }
          }
        },
        "eventTime": "2024-07-02T00:22:11Z",
        "eventSource": "s3express.amazonaws.com",
        "eventName": "CreateSession",
        "awsRegion": "us-west-2",
        "sourceIPAddress": "72.21.198.68",
        "userAgent": "aws-sdk-java/2.20.160-SNAPSHOT
 Linux/5.10.216-225.855.amzn2.x86_64 OpenJDK_64-Bit_Server_VM/11.0.23+9-LTS
 Java/11.0.23 vendor/Amazon.com_Inc. md/internal exec-env/AWS_Lambda_java11 io/sync
 http/Apache cfg/retry-mode/standard",
        "requestParameters": {
          "bucketName": "bucket-base-name--usw2-az1--x-s3".
            "host": "bucket-base-name--usw2-az1--x-s3.s3express-usw2-az1.us-
west-2.amazonaws.com",
            "x-amz-create-session-mode": "ReadWrite"
        },
        "responseElements": {
            "credentials": {
                "accessKeyId": "AKIAI44QH8DHBEXAMPLE"
                "expiration": ""Mar 20, 2024, 11:16:09 PM",
                "sessionToken": "<session token string>"
           },
        },
        "additionalEventData": {
            "SignatureVersion": "SigV4",
            "cipherSuite": "TLS_AES_128_GCM_SHA256",
            "bytesTransferredIn": 0,
            "AuthenticationMethod": "AuthHeader",
```

```
            "xAmzId2": "q6xhNJYmhg",
            "bytesTransferredOut": 1815,
            "availabilityZone": "usw2-az1"
          },
          "requestID": "28d2faaf-3319-4649-998d-EXAMPLE72818",
          "eventID": "694d604a-d190-4470-8dd1-EXAMPLEe20c1",
          "readOnly": true,
          "resources": [
            {
              "type": "AWS::S3Express::Object",
              "ARNPrefix": "arn:aws:s3express:us-west-2:111122223333:bucket-base-name--
usw2-az1--x-s3"
            },
            {
              "accountId": "111122223333"
              "type": "AWS::S3Express::DirectoryBucket",
              "ARN": "arn:aws:s3express:us-west-2:111122223333:bucket-base-name--usw2-
az1--x-s3"
             }
           ],
           "eventType": "AwsApiCall",
           "managementEvent": false,
           "recipientAccountId": "111122223333",
           "eventCategory": "Data",
           "tlsDetails": {
             "tlsVersion": "TLSv1.3",
             "cipherSuite": "TLS_AES_128_GCM_SHA256",
             "clientProvidedHostHeader": "bucket-base-name--usw2-az1--x-s3.s3express-
usw2-az1.us-west-2.amazonaws.com"
            }
          }
```

Untuk menggunakan operasi API titik akhir Zonal (operasi tingkat objek, atau bidang data), Anda dapat menggunakan operasi `CreateSession` API untuk membuat dan mengelola sesi yang dioptimalkan untuk otorisasi permintaan data latensi rendah. Anda juga dapat menggunakan `CreateSession` untuk mengurangi jumlah logging. Untuk mengidentifikasi operasi Zonal API mana yang dilakukan selama sesi, Anda dapat mencocokkan bagian `accessKeyId` bawah `responseElements` dalam file `CreateSession` log Anda dengan `accessKeyId` file log operasi API Zonal lainnya. Untuk informasi selengkapnya, lihat `CreateSessionotorisasi`.

Contoh berikut menunjukkan contoh file CloudTrail log yang menunjukkan operasi `GetObject`API yang diautentikasi oleh. `CreateSession`

```
    {
        "eventVersion": "1.09",
        "userIdentity": {
          "type": "AssumedRole",
          "principalId": "AROAIDPPEZS35WEXAMPLE:AssumedRoleSessionName",
          "arn": "arn:aws:sts::111122223333assumed-role/RoleToBeAssumed/MySessionName",
          "accountId": "111122223333",
          "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
          "sessionContext": {
            "attributes": {
              "creationDate": "2024-07-02T00:21:49Z"
            }
          }
        },
        "eventTime": "2024-07-02T00:22:01Z",
        "eventSource": "s3express.amazonaws.com",
        "eventName": "GetObject",
        "awsRegion": "us-west-2",
        "sourceIPAddress": "72.21.198.68",
        "userAgent": "aws-sdk-java/2.25.66 Linux/5.10.216-225.855.amzn2.x86_64
 OpenJDK_64-Bit_Server_VM/17.0.11+9-LTS Java/17.0.11 vendor/Amazon.com_Inc. md/internal
 exec-env/AWS_Lambda_java17 io/sync http/Apache cfg/retry-mode/legacy",
        "requestParameters": {
          "bucketName": "bucket-base-name--usw2-az1--x-s3",
          "x-amz-checksum-mode": "ENABLED",
          "Host": "bucket-base-name--usw2-az1--x-s3.s3express-usw2-az1.us-
west-2.amazonaws.com",
          "key": "test-get-obj-with-checksum"
        },
        "responseElements": null,
        "additionalEventData": {
          "SignatureVersion": "Sigv4",
          "CipherSuite": "TLS_AES_128_GCM_SHA256",
          "bytesTransferredIn": 0,
          "AuthenticationMethod": "AuthHeader",
          "x-amz-id-2": "oOy6w8K7LFsyFN",
          "bytesTransferredOut": 9,
          "availabilityZone": "usw2-az1",
          "sessionModeApplied": "ReadWrite"
         },
          "requestID": "28d2faaf-3319-4649-998d-EXAMPLE72818",
          "eventID": "694d604a-d190-4470-8dd1-EXAMPLEe20c1",
          "readOnly": true,
```

```
          "resources": [
            {
              "type": "AWS::S3Express::Object",
              "ARNPrefix": "arn:aws:s3express:us-west-2:111122223333:bucket-base-name--
usw2-az1--x-s3"
            },
            {
              "accountId": "111122223333",
              "type": "AWS::S3Express::DirectoryBucket",
              "ARN": "arn:aws:s3express:us-west-2:111122223333:bucket-base-name--usw2-
az1--x-s3"
             }
           ],
           "eventType": "AwsApiCall",
           "managementEvent": false,
           "recipientAccountId": "111122223333",
           "eventCategory": "Data",
           "tlsDetails": {
             "tlsVersion": "TLSv1.3",
             "cipherSuite": "TLS_AES_128_GCM_SHA256",
             "clientProvidedHostHeader": "bucket-base-name--usw2-az1--x-s3.s3express-
usw2-az1.us-west-2.amazonaws.com"
            }
          }
```

Dalam contoh file `GetObject` log di atas, `accessKeyId` (AKIAI44QH8DHBEXAMPLE) cocok dengan `accessKeyId` di bawah `responseElements` dalam contoh file log. CreateSession Pencocokan `accessKeyId` menunjukkan sesi di mana `GetObject` operasi dilakukan.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan `DeleteObjects` tindakan pada bucket direktori, dipanggil oleh Siklus Hidup S3. Untuk informasi selengkapnya, lihat Working with S3 Lifecycle for directory buckets.

```
eventVersion:"1.09",
  userIdentity:{

    type:"AWSService",
    invokedBy:"lifecycle.s3.amazonaws.com"
  },
  eventTime:"2024-09-11T00:55:54Z",
  eventSource:"s3express.amazonaws.com",
  eventName:"DeleteObjects",
  awsRegion:"us-east-2",
```

```
  sourceIPAddress:"lifecycle.s3.amazonaws.com",
  userAgent:"gamma.lifecycle.s3.amazonaws.com",
  requestParameters:{

    bucketName:"amzn-s3-demo-bucket--use2-az2--x-s3",
    'x-amz-expected-bucket-owner':"637423581905",
    Host:"amzn-s3-demo-bucket--use2-az2--x-s3.gamma.use2-az2.express.s3.aws.dev",
    delete:"",
    'x-amz-sdk-checksum-algorithm':"CRC32C"
  },
  responseElements:null,
  additionalEventData:{

    SignatureVersion:"Sigv4",
    CipherSuite:"TLS_AES_128_GCM_SHA256",
    bytesTransferredIn:41903,
    AuthenticationMethod:"AuthHeader",
    'x-amz-id-2':"9H5YWZY0",
    bytesTransferredOut:35316,
    availabilityZone:"use2-az2",
    sessionModeApplied:"ReadWrite"
  },
  requestID:"011eeadd04000191",
  eventID:"d3d8b116-219d-4ee6-a072-5f9950733c74",
  readOnly:false,
  resources:[

    {

      type:"AWS::S3Express::Object",
      ARNPrefix:"arn:aws:s3express:us-east-2:637423581905:bucket/amzn-s3-demo-bucket--
use2-az2--x-s3/"
    },
    {

      accountId:"637423581905",
      type:"AWS::S3Express::DirectoryBucket",
      ARN:"arn:aws:s3express:us-east-2:637423581905:bucket/amzn-s3-demo-bucket--use2-
az2--x-s3"
    }
  ],
  eventType:"AwsApiCall",
  managementEvent:false,
  recipientAccountId:"637423581905",
```

```
  sharedEventID:"59f877ac-1dd9-415d-b315-9bb8133289ce",
  eventCategory:"Data"
}
```

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan `Access Denied` permintaan pada `CreateSession` tindakan yang dipanggil oleh Siklus Hidup S3. Untuk informasi selengkapnya, lihat CreateSession.

```
{
    "eventVersion": "1.09",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "gamma.lifecycle.s3.amazonaws.com"
    },
    "eventTime": "2024-09-11T18:13:08Z",
    "eventSource": "s3express.amazonaws.com",
    "eventName": "CreateSession",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "gamma.lifecycle.s3.amazonaws.com",
    "userAgent": "gamma.lifecycle.s3.amazonaws.com",
    "errorCode": "AccessDenied",
    "errorMessage": "Access Denied",
    "requestParameters": {
        "bucketName": "amzn-s3-demo-bucket--use2-az2--x-s3",
        "Host": "amzn-s3-demo-bucket--use2-az2--x-s3.gamma.use2-
az2.express.s3.aws.dev",
        "x-amz-create-session-mode": "ReadWrite",
        "x-amz-server-side-encryption": "AES256"
    },
    "responseElements": null,
    "additionalEventData": {
        "SignatureVersion": "Sigv4",
        "CipherSuite": "TLS_AES_128_GCM_SHA256",
        "bytesTransferredIn": 0,
        "AuthenticationMethod": "AuthHeader",
        "x-amz-id-2": "zuDDC1VNbC4LoNwUIc5",
        "bytesTransferredOut": 210,
        "availabilityZone": "use2-az2"
    },
    "requestID": "010932f174000191e24a0",
    "eventID": "dce7cc46-4cd3-46c0-9a47-d1b8b70e301c",
    "readOnly": true,
    "resources": [{
```

```
            "type": "AWS::S3Express::Object",
            "ARNPrefix": "arn:aws:s3express:us-east-2:637423581905:bucket/amzn-s3-demo-
bucket--use2-az2--x-s3/"
        },
        {
            "accountId": "637423581905",
            "type": "AWS::S3Express::DirectoryBucket",
            "ARN": "arn:aws:s3express:us-east-2:637423581905:bucket/amzn-s3-demo-
bucket--use2-az2--x-s3"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "637423581905",
    "sharedEventID": "da96b5bd-6066-4a8d-ad8d-f7f427ca7d58",
    "eventCategory": "Data"
}
```

Mengoptimalkan kinerja bucket direktori

Untuk mendapatkan kinerja terbaik saat menggunakan bucket direktori, kami merekomendasikan pedoman berikut.

Untuk informasi selengkapnya tentang praktik terbaik untuk S3 Express One Zone, lihatPraktik terbaik untuk mengoptimalkan kinerja S3 Express One Zone.

Gunakan autentikasi berbasis sesi

Bucket direktori mendukung mekanisme otorisasi berbasis sesi baru untuk mengautentikasi dan mengotorisasi permintaan ke bucket direktori. Dengan otentikasi berbasis sesi, secara AWS SDKs otomatis menggunakan operasi `CreateSession` API untuk membuat token sesi sementara yang dapat digunakan untuk otorisasi latensi rendah permintaan data ke bucket direktori.

AWS SDKs Gunakan operasi `CreateSession` API untuk meminta kredensyal sementara, lalu secara otomatis membuat dan menyegarkan token untuk Anda atas nama Anda setiap 5 menit. Untuk memanfaatkan manfaat kinerja bucket direktori, sebaiknya gunakan file AWS SDKs untuk memulai dan mengelola permintaan `CreateSession` API. Untuk informasi lebih lanjut tentang model berbasis sesi ini, lihat Mengotorisasi operasi API titik akhir Zonal dengan `CreateSession`.

Praktik terbaik checksum tambahan S3

Bucket direktori menawarkan opsi untuk memilih algoritma checksum yang digunakan untuk memvalidasi data Anda saat mengunggah atau mengunduh. Anda dapat memilih salah satu algoritma pemeriksaan integritas data Secure Hash Algorithms (SHA) atau Cyclic Redundancy Check (CRC) berikut:, C, SHA-1, dan SHA-256. CRC32 CRC32 MD5checksum berbasis tidak didukung dengan kelas penyimpanan S3 Express One Zone.

CRC32 adalah checksum default yang digunakan oleh AWS SDKs saat mengirimkan data ke atau dari ember direktori. Kami merekomendasikan penggunaan CRC32 dan CRC32 C untuk kinerja terbaik dengan bucket direktori.

Gunakan versi terbaru dari AWS SDKs pustaka runtime umum

Beberapa di antaranya AWS SDKs juga menyediakan pustaka AWS Common Runtime (CRT) untuk lebih mempercepat kinerja di klien S3. Ini SDKs termasuk AWS SDK for Java 2.x, AWS SDK untuk C++, dan AWS SDK untuk Python (Boto3). Klien S3 berbasis CRT mentransfer objek ke dan dari bucket direktori dengan kinerja dan keandalan yang ditingkatkan dengan secara otomatis menggunakan operasi API unggahan multibagian dan pengambilan rentang byte untuk mengotomatiskan koneksi penskalaan horizontal.

Untuk mencapai kinerja tertinggi dengan bucket direktori, sebaiknya gunakan versi terbaru AWS SDKs yang menyertakan pustaka CRT atau menggunakan (). AWS Command Line Interface AWS CLI

Berkembang dengan ember direktori

Setelah Anda membuat bucket direktori Anda, Anda kemudian dapat segera memulai membaca dan menulis latensi sangat rendah. Anda dapat berkomunikasi dengan bucket direktori menggunakan koneksi titik akhir melalui cloud privat virtual (VPC), atau menggunakan operasi API Zonal dan Regional untuk mengelola objek dan bucket direktori Anda. Anda dapat bekerja dengan bucket direktori dengan menggunakan konsol Amazon S3 AWS SDKs, Command Line Interface (AWS CLI) AWS , dan Amazon S3 REST. APIs

Topik

- Titik akhir regional dan Zonal untuk bucket direktori
- Bekerja dengan bucket direktori dengan menggunakan konsol S3,, dan AWS CLIAWS SDKs
- Operasi API bucket direktori

Titik akhir regional dan Zonal untuk bucket direktori

Untuk mengakses titik akhir Regional dan Zonal untuk bucket direktori dari virtual private cloud (VPC), Anda dapat menggunakan titik akhir VPC gateway. Setelah membuat titik akhir gateway, Anda dapat menambahkannya sebagai target di tabel rute untuk lalu lintas yang ditujukan dari VPC ke bucket Anda. Tidak dikenakan biaya tambahan untuk menggunakan titik akhir gateway. Untuk informasi selengkapnya tentang cara mengonfigurasi titik akhir VPC gateway, lihat Jaringan untuk ember direktori.

Operasi API tingkat ember (bidang kontrol) tersedia melalui titik akhir Regional dan disebut sebagai operasi API titik akhir Regional. Contoh operasi API titik akhir Regional adalah `CreateBucket` dan `DeleteBucket`.

Anda menggunakan Zonal (tingkat objek, atau operasi API titik akhir bidang data) untuk mengunggah dan mengelola objek Anda. Operasi API titik akhir Zona tersedia melalui titik akhir Zona. Contoh operasi API Zona adalah `PutObject` dan `CopyObject`.

Untuk informasi selengkapnya tentang titik akhir Regional dan Zonal untuk bucket direktori di Availability Zones, lihat. Titik akhir regional dan Zonal untuk bucket direktori di Availability Zone

Untuk informasi selengkapnya tentang titik akhir Regional dan Zonal untuk bucket direktori di Local Zones, lihat. Konsep untuk bucket direktori di Local Zones

Bekerja dengan bucket direktori dengan menggunakan konsol S3,, dan AWS CLIAWS SDKs

Anda dapat bekerja dengan kelas penyimpanan S3 Express One Zone dan bucket direktori dengan menggunakan, konsol Amazon S3 AWS SDKs,AWS CLI() AWS Command Line Interface , dan Amazon S3 REST API.

Konsol S3

Untuk mulai menggunakan konsol S3, ikuti langkah-langkah ini:

- Membuat bucket direktori di Availability Zone
- Mengosongkan bucket direktori
- Menghapus bucket direktori

Untuk tutorial selengkapnya, lihat Tutorial: Memulai dengan S3 Express One Zone.

AWS SDKs

S3 Express One Zone mendukung hal berikut: AWS SDKs

- AWS SDK untuk C++
- AWS SDK untuk Go v2
- AWS SDK for Java 2.x
- AWS SDK untuk JavaScript v3
- AWS SDK untuk .NET
- AWS SDK untuk PHP
- AWS SDK untuk Python (Boto3)
- AWS SDK untuk Ruby
- AWS SDK untuk Kotlin
- AWS SDK for Rust

Saat Anda bekerja dengan S3 Express One Zone, kami sarankan untuk menggunakan versi terbaru. AWS SDKs Yang didukung AWS SDKs untuk S3 Express One Zone menangani pembentukan sesi, penyegaran, dan penghentian atas nama Anda. Ini berarti bahwa Anda dapat segera mulai menggunakan operasi API setelah Anda men-download dan menginstal AWS SDKs dan mengkonfigurasi izin IAM yang diperlukan. Untuk informasi selengkapnya, lihat Mengotorisasi operasi API endpoint Regional dengan IAM.

Untuk informasi tentang AWS SDKs, termasuk cara mengunduh dan menginstalnya, lihat Alat untuk Dibangun AWS.

Untuk contoh AWS SDK, lihat berikut ini:

- Membuat bucket direktori di Availability Zone
- Mengosongkan bucket direktori
- Menghapus bucket direktori

AWS Command Line Interface (AWS CLI)

Anda dapat menggunakan AWS Command Line Interface (AWS CLI) untuk membuat bucket direktori dan menggunakan operasi API titik akhir Regional dan Zonal yang didukung untuk S3 Express One Zone.

Untuk memulai AWS CLI, lihat Memulai dengan AWS CLI di AWS CLI Command Reference.

> **Note**
>
> Untuk menggunakan bucket direktori dengan `aws s3` perintah tingkat tinggi, perbarui AWS CLI ke versi terbaru. Untuk informasi selengkapnya tentang cara menginstal dan mengonfigurasi AWS CLI, lihat Menginstal atau memperbarui versi terbaru dari Referensi AWS CLI Perintah. AWS CLI

Sebagai AWS CLI contoh, lihat yang berikut ini:

- Membuat bucket direktori di Availability Zone
- Mengosongkan bucket direktori
- Menghapus bucket direktori

Operasi API bucket direktori

Untuk mengelola bucket direktori, Anda dapat menggunakan operasi API titik akhir Regional (tingkat bucket, atau bidang kontrol). Untuk mengelola objek di bucket direktori, Anda dapat menggunakan operasi API titik akhir Zonal (level objek, atau bidang data). Untuk informasi selengkapnya, lihat Jaringan untuk ember direktori dan Titik akhir dan titik akhir VPC gateway.

Operasi API titik akhir regional

Operasi API titik akhir Regional berikut didukung untuk bucket direktori:

- CreateAccessPoint
- CreateBucket
- DeleteAccessPoint
- DeleteAccessPointPolicy
- DeleteAccessPointScope
- DeleteBucket
- DeleteBucketLifecycle
- DeleteBucketPolicy
- GetAccessPoint

- GetAccessPointPolicy
- GetAccessPointScope
- GetBucketLifecycleConfiguration
- GetBucketPolicy
- ListAccessPointsForDirectoryBuckets
- ListDirectoryBuckets
- ListTagsForResource
- PutAccessPointPolicy
- PutAccessPointScope
- PutBucketLifecycleConfiguration
- PutBucketPolicy
- DeleteBucketEncryption
- GetBucketEncryption
- PutBucketEncryption
- TagResource
- UntagResource

Operasi API titik akhir zonal

Operasi API titik akhir Zonal berikut didukung untuk bucket direktori:

- CreateSession
- CopyObject
- DeleteObject
- DeleteObjects
- GetObject
- GetObjectAttributes
- HeadBucket
- HeadObject
- ListObjectsV2
- PutObject
- RenameObject

- AbortMultipartUpload
- CompleteMultiPartUpload
- CreateMultipartUpload
- ListMultipartUploads
- ListParts
- UploadPart
- UploadPartCopy

Menggunakan tag dengan ember direktori S3

AWS Tag adalah pasangan nilai kunci yang menyimpan metadata tentang sumber daya, dalam hal ini bucket direktori Amazon S3. Anda dapat menandai bucket direktori S3 saat Anda membuatnya atau mengelola tag pada bucket direktori yang ada. Tidak ada biaya tambahan untuk menggunakan tag pada bucket direktori di luar tarif permintaan API S3 standar. Untuk informasi selengkapnya, lihat Harga Amazon S3.

Bagaimana tag bekerja

Bucket direktori Amazon S3 mendukung dua jenis tag:

- AWS tag yang dihasilkan: AWS secara otomatis menerapkan tag ini, dan Anda tidak dapat memodifikasinya atau menghapusnya. Untuk mempelajari lebih lanjut tentang tag AWS-generated, lihat Menggunakan tag AWS-generated.
- Tag yang ditentukan pengguna: Anda menerapkan tag ini ke bucket direktori S3 atau sumber daya lainnya dan mengelolanya.

Tag yang ditentukan pengguna

Tag yang ditentukan pengguna adalah pasangan nilai kunci tag yang Anda gunakan untuk memberi label sumber daya Anda. Tag yang ditentukan pengguna terdiri dari kunci yang diperlukan dan nilai opsional. Ini adalah komponen utama dari tag yang ditentukan pengguna:

Kunci tag

Kunci tag adalah nama tag yang wajib diisi. Misalnya, `project` adalah kunci tag dalam pasangan nilai kunci tag berikut:

Kunci	Nilai
`project`	`Trinity`

Kunci tag adalah string case-sensitive yang harus berisi antara 1 dan 128 karakter Unicode.

Nilai tag

Nilai tag adalah string opsional. Misalnya, `Trinity` adalah nilai tag dalam pasangan nilai kunci tag ini:

Kunci	Nilai
`project`	`Trinity`

Nilai tag adalah string case-sensitive yang dapat berisi antara 0 dan 256 karakter Unicode.

Untuk detail tentang karakter yang diizinkan dalam tag yang ditentukan pengguna dan batasan lainnya, lihat Pembatasan Tag yang Ditentukan Pengguna di Panduan Pengguna.AWS Manajemen Penagihan dan Biaya

Set tag

Setiap bucket direktori S3 memiliki satu set tag yang berisi semua kunci tag dan pasangan nilai yang ditetapkan ke bucket tersebut. Satu set tag dapat berisi sebanyak 50 tag yang ditentukan pengguna, atau bisa kosong.

Meskipun setiap kunci harus unik dalam satu set tag, Anda dapat menggunakan nilai yang sama beberapa kali. Misalnya, Anda dapat memiliki nilai yang sama,`Trinity`, untuk mengikuti dua kunci tag:

Kunci	Nilai
`project`	`Trinity`
`cost-center`	`Trinity`

Dalam bucket, saat Anda menambahkan tag yang memiliki kunci yang sama dengan tag yang ada, nilai baru akan menimpa nilai lama.

AWS tidak menerapkan arti semantik apa pun pada tag Anda. Kami menafsirkan tanda hanya sebagai string karakter.

Untuk menambahkan, membuat daftar, memodifikasi, atau menghapus tag, Anda dapat menggunakan konsol Amazon S3, Antarmuka Baris AWS Perintah (AWS CLI), atau Amazon S3 API.

Cara umum untuk menggunakan tag

Gunakan tag pada bucket direktori S3 Anda untuk:

1. Alokasi biaya — Lacak biaya penyimpanan dengan tag bucket in AWS Manajemen Penagihan dan Biaya.
2. Attribute based access control (ABAC) — Skala izin akses dan berikan akses ke bucket direktori S3 berdasarkan tag mereka.

> **Note**
>
> Anda dapat menggunakan tag yang sama untuk alokasi biaya dan kontrol akses.

Menggunakan tag untuk alokasi biaya

Lacak biaya penyimpanan Amazon S3 Anda dengan menerapkan tag ke bucket direktori S3 dan mengaktifkan tag ini untuk alokasi biaya.

Untuk mulai melacak biaya:

1. Tambahkan tag ke bucket direktori S3 Anda atau gunakan tag yang ada. Untuk informasi selengkapnya tentang cara menambahkan tag yang ditentukan pengguna ke bucket direktori Anda, lihat. Bekerja dengan tag Misalnya, Anda dapat memberi label bucket dengan tag yang mengidentifikasi proyek atau grup proyek.
2. Aktifkan tag untuk alokasi biaya di AWS Manajemen Penagihan dan Biaya konsol. Lihat Mengaktifkan tag alokasi biaya yang ditentukan pengguna di Panduan Pengguna.AWS Manajemen Penagihan dan Biaya Anda dapat mengaktifkan tag yang ditentukan pengguna atau AWS yang dihasilkan. Untuk informasi selengkapnya, lihat Mengatur dan melacak AWS biaya menggunakan tag alokasi biaya.

AWS menggunakan tag yang diaktifkan untuk mengatur biaya sumber daya Anda di berbagai alat penagihan dan manajemen biaya, seperti:

- Laporan alokasi biaya

  Memberikan tampilan biaya tingkat tinggi yang diatur oleh tag aktif Anda. Untuk informasi selengkapnya, lihat Menggunakan laporan alokasi biaya bulanan di Panduan Pengguna AWS Penagihan.

- Laporan Biaya dan Penggunaan (CUR)

  Menyediakan kumpulan data AWS biaya dan penggunaan yang paling rinci, termasuk rincian biaya berbasis tag. Untuk informasi selengkapnya, lihat Apa itu Laporan AWS Biaya dan Penggunaan? dalam Panduan Pengguna Ekspor AWS Data.

Menggunakan tag untuk kontrol akses berbasis atribut (ABAC)

Attribute-based access control (ABAC) adalah strategi otorisasi yang mendefinisikan izin berdasarkan atribut, yaitu tag. Anda dapat melampirkan tag ke entitas AWS Identity and Access Management (IAM) (pengguna atau peran) dan ke AWS sumber daya, seperti bucket direktori Amazon S3. Kemudian, Anda mengontrol izin ke sumber daya ini menggunakan kondisi berbasis tag dalam kebijakan kontrol akses untuk mengizinkan atau menolak operasi saat kondisi ini terpenuhi.

ABAC untuk ember direktori S3

Bucket direktori Amazon S3 mendukung kontrol akses berbasis atribut (ABAC) menggunakan tag. Gunakan kunci kondisi berbasis tag di AWS organisasi, IAM, dan kebijakan bucket direktori S3 Anda. Untuk perusahaan, ABAC di Amazon S3 mendukung otorisasi di beberapa akun. AWS

Dalam kebijakan IAM, Anda dapat mengontrol akses ke bucket direktori S3 berdasarkan tag bucket dengan menggunakan kunci kondisi global berikut:

- `aws:ResourceTag/key-name`
  - Gunakan kunci ini untuk membandingkan pasangan nilai kunci tag yang Anda tentukan dalam kebijakan dengan pasangan nilai kunci yang dilampirkan ke sumber daya. Misalnya, Anda dapat meminta agar akses ke sumber daya hanya diperbolehkan jika sumber daya memiliki kunci tanda yang dilampirkan `Dept` dengan nilai `Marketing`. Untuk informasi selengkapnya, lihat Mengontrol akses ke AWS sumber daya.
- `aws:RequestTag/key-name`

  - Gunakan kunci ini untuk membandingkan pasangan nilai kunci tanda yang diteruskan dalam permintaan dengan pasangan tanda yang Anda sebutkan dalam kebijakan. Misalnya, Anda dapat memeriksa apakah permintaan tersebut menyertakan kunci tanda `Dept` dan memiliki nilai `Accounting`. Untuk informasi selengkapnya, lihat Mengontrol akses selama AWS permintaan. Anda dapat menggunakan kunci kondisi ini untuk membatasi pasangan nilai kunci tag mana yang dapat diteruskan selama operasi `TagResource` dan `CreateBucket` API.
- `aws:TagKeys`
  - Gunakan kunci ini untuk membandingkan kunci tanda dalam permintaan dengan kunci yang Anda sebutkan dalam kebijakan. Sebaiknya saat Anda menggunakan kebijakan untuk mengontrol akses menggunakan tag, gunakan tombol `aws:TagKeys` kondisi untuk menentukan kunci tag apa yang diizinkan. Misalnya kebijakan dan informasi selengkapnya, lihat Mengontrol akses berdasarkan kunci tag. Anda dapat membuat bucket direktori S3 dengan tag. Untuk mengizinkan penandaan selama operasi `CreateBucket` API, Anda harus membuat kebijakan yang mencakup `s3express:CreateBucket` tindakan `s3express:TagResource` dan tindakan. Anda kemudian dapat menggunakan kunci `aws:TagKeys` kondisi untuk menerapkan menggunakan tag tertentu dalam `CreateBucket` permintaan.
- `s3express:BucketTag/tag-key`
  - Gunakan kunci kondisi ini untuk memberikan izin ke data tertentu dalam bucket direktori menggunakan tag. Saat mengakses bucket direktori dengan menggunakan titik akses, kunci kondisi ini mereferensikan tag pada bucket direktori baik saat mengotorisasi terhadap titik akses dan bucket direktori, sedangkan `aws:ResourceTag/tag-key` akan mereferensikan tag hanya dari sumber daya yang diotorisasi.

Contoh kebijakan

Lihat contoh kebijakan ABAC berikut untuk bucket direktori Amazon S3.

1.1 - Kebijakan IAM untuk membuat atau memodifikasi bucket dengan tag tertentu

Dalam kebijakan IAM ini, pengguna atau peran dengan kebijakan ini hanya dapat membuat bucket direktori S3 jika mereka menandai bucket dengan kunci tag `project` dan nilai tag `Trinity` dalam permintaan pembuatan bucket. Mereka juga dapat menambahkan atau memodifikasi tag pada bucket direktori S3 yang ada selama `TagResource` permintaan tersebut menyertakan pasangan nilai kunci tag. `project:Trinity` Kebijakan ini tidak memberikan izin baca, tulis, atau hapus pada bucket atau objeknya.

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CreateBucketWithTags",
      "Effect": "Allow",
      "Action": [
        "s3express:CreateBucket",
        "s3express:TagResource"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/project": [
            "Trinity"
          ]
        }
      }
    }
  ]
}
```

1.2 - Kebijakan bucket untuk membatasi operasi pada bucket menggunakan tag

Dalam kebijakan bucket ini, prinsipal IAM (pengguna dan peran) dapat melakukan operasi menggunakan `CreateSession` tindakan pada bucket hanya jika nilai tag bucket cocok dengan nilai `project` tag prinsipal. `project`

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowObjectOperations",
      "Effect": "Allow",
      "Principal": {
        "AWS": "111122223333"
      },
      "Action": "s3express:CreateSession",
      "Resource": "arn:aws:s3express:us-west-2:111122223333:bucket/amzn-s3-demo-
bucket--usw2-az1--x-s3",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/project": "${aws:PrincipalTag/project}"
        }
```

```
      }
    }
  ]
}
```

1.3 - Kebijakan IAM untuk memodifikasi tag pada sumber daya yang ada mempertahankan tata kelola penandaan

Dalam kebijakan IAM ini, prinsipal IAM (pengguna atau peran) dapat memodifikasi tag pada bucket hanya jika nilai tag bucket cocok dengan nilai `project` tag prinsipal. `project` Hanya empat tag`project`,`environment`,`owner`, dan `cost-center` ditentukan dalam kunci `aws:TagKeys` kondisi yang diizinkan untuk bucket direktori ini. Ini membantu menegakkan tata kelola tag, mencegah modifikasi tag yang tidak sah, dan menjaga skema penandaan tetap konsisten di seluruh bucket Anda.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EnforceTaggingRulesOnModification",
      "Effect": "Allow",
      "Action": [
        "s3express:TagResource"
      ],
      "Resource": "arn:aws:s3express:*:*:bucket/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/project": "${aws:PrincipalTag/project}"
        },
        "ForAllValues:StringEquals": {
          "aws:TagKeys": [
            "project",
            "environment",
            "owner",
            "cost-center"
          ]
        }
      }
    }
  ]
}
```

1.4 - Menggunakan kunci kondisi s3express: BucketTag

Dalam kebijakan IAM ini, pernyataan kondisi mengizinkan akses ke data bucket hanya jika bucket memiliki kunci tag `Environment` dan nilai `Production` tag.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAccessToSpecificAccessPoint",
      "Effect": "Allow",
      "Action": "*",
      "Resource": "arn:aws:s3express:*:*:accesspoint/*",
      "Condition": {
        "StringEquals": {
          "s3express:BucketTag/Environment": "Production"
        }
      }
    }
  ]
}
```

Bekerja dengan tag

Anda dapat menambahkan atau mengelola tag untuk bucket direktori S3 menggunakan Konsol Amazon S3, Antarmuka Baris AWS Perintah (CLI), AWS SDKs atau menggunakan S3:,, dan. APIs TagResourceUntagResourceListTagsForResource Untuk informasi selengkapnya, lihat:

Topik

- Membuat ember direktori dengan tag
- Menambahkan tag ke bucket direktori
- Melihat tag bucket direktori
- Menghapus tag dari bucket direktori

Membuat ember direktori dengan tag

Anda dapat menandai bucket direktori Amazon S3 saat Anda membuatnya. Tidak ada biaya tambahan untuk menggunakan tag pada bucket direktori di luar tarif permintaan API S3 standar.

Untuk informasi selengkapnya, lihat Harga Amazon S3. Untuk informasi selengkapnya tentang menandai bucket direktori, lihat. Menggunakan tag dengan ember direktori S3

Izin

Untuk membuat bucket direktori dengan tag, Anda harus memiliki izin berikut:

- `s3express:CreateBucket`
- `s3express:TagResource`

Memecahkan masalah kesalahan

Jika Anda mengalami kesalahan saat mencoba membuat bucket direktori dengan tag, Anda dapat melakukan hal berikut:

- Verifikasi bahwa Anda memiliki kebutuhan Izin untuk membuat bucket direktori dan menambahkan tag ke dalamnya.
- Periksa kebijakan pengguna IAM Anda untuk kondisi kontrol akses berbasis atribut (ABAC) apa pun. Anda mungkin diminta untuk memberi label pada bucket direktori Anda hanya dengan kunci dan nilai tag tertentu. Untuk informasi selengkapnya, lihat Menggunakan tag untuk kontrol akses berbasis atribut (ABAC).

Langkah-langkah

Anda dapat membuat bucket direktori dengan tag yang diterapkan menggunakan konsol Amazon S3, AWS Command Line Interface (AWS CLI), Amazon S3 REST API, dan. AWS SDKs

Menggunakan konsol S3

Untuk membuat bucket direktori dengan tag menggunakan konsol Amazon S3:

1. Masuk ke konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih bucket direktori.
3. Pilih create bucket untuk membuat bucket direktori baru.
4. Anda dapat membuat dua jenis bucket direktori:

   Buat bucket direktori di Availability Zone untuk beban kerja berkinerja tinggi. Untuk informasi selengkapnya, lihat Beban kerja kinerja tinggi.

Buat bucket direktori di Zona Lokal untuk beban kerja residensi data. Untuk informasi selengkapnya, lihat Beban kerja residensi data.

5. Untuk kedua jenis bucket direktori, pada halaman Buat bucket, Tag adalah opsi saat membuat bucket direktori baru.
6. Masukkan nama untuk ember. Untuk informasi selengkapnya, lihat

Saat membuat bucket direktori di Amazon S3, aturan penamaan bucket berikut berlaku. Untuk aturan penamaan bucket bertujuan umum, lihat Aturan penamaan bucket tujuan umum.

Nama bucket direktori terdiri dari nama dasar yang Anda berikan, dan akhiran yang berisi ID AWS Zona (Availability Zone atau Local Zone) tempat bucket Anda berada dan`--x-s3`. *`zone-id`*Bisa berupa ID Availability Zone atau Local Zone.

```
base-name--zoneid--x-s3
```

Misalnya, nama bucket direktori berikut berisi ID Zona Ketersediaan `usw2-az1`:

```
bucket-base-name--usw2-az1--x-s3
```

> **Note**
>
> Saat Anda membuat bucket direktori menggunakan konsol, akhiran secara otomatis ditambahkan ke nama dasar yang Anda berikan. Akhiran ini mencakup ID Zona Zona (Availability Zone atau Local Zone) yang Anda pilih.

> Saat membuat bucket direktori menggunakan API, Anda harus memberikan akhiran lengkap, termasuk ID Zone, dalam permintaan Anda. Untuk daftar Zone IDs, lihatTitik akhir.

Aturan penamaan berikut berlaku untuk bucket direktori.

- Jadilah unik dalam Zona yang dipilih (AWS Availability Zone atau AWS Local Zone).
- Panjang nama harus antara 3 (menit) dan 63 (maks) karakter, termasuk akhiran.
- Hanya terdiri dari huruf kecil, angka, tanda hubung (-).
- Dimulai dan diakhiri dengan huruf atau angka.
- Harus menyertakan akhiran berikut:`--`*`zone-id`*`--x-s3`.

- Nama bucket tidak boleh dimulai dengan prefiks `xn--`.

- Nama bucket tidak boleh dimulai dengan prefiks `sthree-`.
- Nama bucket tidak boleh dimulai dengan prefiks `sthree-configurator`.
- Nama bucket tidak boleh dimulai dengan prefiks `amzn-s3-demo-`.
- Nama bucket tidak boleh diakhiri dengan sufiks `-s3alias`. Sufiks ini dicadangkan untuk nama alias titik akses. Untuk informasi selengkapnya, lihat Alias titik akses.
- Nama bucket tidak boleh diakhiri dengan sufiks `--ol-s3`. Sufiks ini dicadangkan untuk nama alias Titik Akses Lambda Objek. Untuk informasi selengkapnya, lihat Cara menggunakan alias gaya bucket untuk bucket S3 Anda Titik Akses Lambda Objek.
- Nama bucket tidak boleh diakhiri dengan sufiks `.mrap`. Akhiran ini dicadangkan untuk nama Titik Akses Multi-Wilayah. Untuk informasi selengkapnya, lihat Aturan penamaan Titik Akses Multi-Wilayah Amazon S3.

.

7. Pilih Tambahkan Tag baru untuk membuka editor Tag dan masukkan pasangan nilai kunci tag. Kunci tag diperlukan, tetapi nilainya opsional.
8. Untuk menambahkan tag lain, pilih Tambah Tag baru lagi. Anda dapat memasukkan hingga 50 pasangan nilai kunci tag.
9. Setelah Anda menyelesaikan menentukan opsi untuk bucket direktori baru Anda, pilih Buat bucket.

Menggunakan AWS SDKs

SDK for Java 2.x

Contoh ini menunjukkan cara membuat bucket direktori dengan tag dengan menggunakan AWS SDK for Java 2.x. Untuk menggunakan perintah ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.BucketInfo;
import software.amazon.awssdk.services.s3.model.BucketType;
import software.amazon.awssdk.services.s3.model.CreateBucketConfiguration;
import software.amazon.awssdk.services.s3.model.CreateBucketRequest;
import software.amazon.awssdk.services.s3.model.CreateBucketResponse;
import software.amazon.awssdk.services.s3.model.DataRedundancy;
```

```
import software.amazon.awssdk.services.s3.model.LocationInfo;
import software.amazon.awssdk.services.s3.model.LocationType;
import software.amazon.awssdk.services.s3.model.Tag;

public class CreateBucketWithTagsExample {
    public static void createBucketWithTagsExample() {
        S3Client s3 = S3Client.builder().region(Region.US_WEST_2).build();

        CreateBucketConfiguration bucketConfiguration =
 CreateBucketConfiguration.builder()
                .location(LocationInfo.builder()
                        .type(LocationType.AVAILABILITY_ZONE)
                        .name("usw2-az1").build())
                .bucket(BucketInfo.builder()
                        .type(BucketType.DIRECTORY)
                        .dataRedundancy(DataRedundancy.SINGLE_AVAILABILITY_ZONE)
                        .build())
                .tags(Tag.builder().key("MyTagKey").value("MyTagValue").build())
                .build();

        CreateBucketRequest createBucketRequest = CreateBucketRequest.builder()
                .bucket("amzn-s3-demo-bucket--usw2-az1--x-s3--usw2-az1--x-s3")
                .createBucketConfiguration(bucketConfiguration)
                .build();

        CreateBucketResponse response = s3.createBucket(createBucketRequest);
        System.out.println("Status code (should be 200):");
        System.out.println(response.sdkHttpResponse().statusCode());
    }
}
```

Penggunaan API REST

Untuk informasi tentang dukungan Amazon S3 REST API untuk membuat bucket direktori dengan tag, lihat bagian berikut di Referensi API Layanan Penyimpanan Sederhana Amazon:

- CreateBucket

Menggunakan AWS CLI

Untuk menginstal AWS CLI, lihat Menginstal AWS CLI di Panduan Pengguna.AWS Command Line Interface

Contoh CLI berikut menunjukkan cara membuat bucket direktori dengan tag dengan menggunakan. AWS CLI Untuk menggunakan perintah ganti *user input placeholders* dengan informasi Anda sendiri.

Saat membuat bucket direktori, Anda harus memberikan detail konfigurasi dan menggunakan konvensi penamaan berikut: *bucket-base-name*--*zone-id*--x-s3

Permintaan:

```
aws s3api create-bucket \
--bucket bucket-base-name--zone-id--x-s3 \
--create-bucket-configuration "Location={Type=AvailabilityZone,Name=zone-
id},Bucket={DataRedundancy=SingleAvailabilityZone,Type=Directory},Tags=[{Key=mykey1,Value=myval
 {Key=mykey2,Value=myvalue2}]"
```

Tanggapan:

```
{
  "Location": "http://bucket--use1-az4--x-s3.s3express-use1-az4.us-
east-1.amazonaws.com/"
}
```

Menambahkan tag ke bucket direktori

Anda dapat menambahkan tag ke bucket direktori Amazon S3 dan memodifikasi tag ini. Tidak ada biaya tambahan untuk menggunakan tag pada bucket direktori di luar tarif permintaan API S3 standar. Untuk informasi selengkapnya, lihat Harga Amazon S3. Untuk informasi selengkapnya tentang menandai bucket direktori, lihat. Menggunakan tag dengan ember direktori S3

Izin

Untuk menambahkan tag ke bucket direktori, Anda harus memiliki izin berikut:

- `s3express:TagResource`

Memecahkan masalah kesalahan

Jika Anda mengalami kesalahan saat mencoba menambahkan tag ke bucket direktori, Anda dapat melakukan hal berikut:

- Verifikasi bahwa Anda memiliki persyaratan Izin untuk menambahkan tag ke bucket direktori.
- Jika Anda mencoba menambahkan kunci tag yang dimulai dengan awalan AWS cadangan`aws:`, ubah kunci tag dan coba lagi.

Langkah-langkah

Anda dapat menambahkan tag ke bucket direktori dengan menggunakan konsol Amazon S3, Command Line AWS Interface (AWS CLI), Amazon S3 REST API, dan. AWS SDKs

Menggunakan konsol S3

Untuk menambahkan tag ke bucket direktori menggunakan konsol Amazon S3:

1. Masuk ke konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih bucket direktori.
3. Pilih nama bucket.
4. Pilih tab Properti.
5. Gulir ke bagian Tag dan pilih Tambahkan Tag baru.
6. Ini membuka halaman Tambah Tag. Anda dapat memasukkan hingga 50 pasangan nilai kunci tag.
7. Jika Anda menambahkan tag baru dengan nama kunci yang sama dengan tag yang ada, nilai tag baru akan menggantikan nilai tag yang ada.
8. Anda juga dapat mengedit nilai tag yang ada di halaman ini.
9. Setelah Anda menambahkan tag, pilih Simpan perubahan.

Menggunakan AWS SDKs

SDK for Java 2.x

Contoh ini menunjukkan cara menambahkan tag ke bucket direktori dengan menggunakan file AWS SDK for Java 2.x. Untuk menggunakan perintah ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.Tag;
import software.amazon.awssdk.services.s3control.model.TagResourceRequest;
```

```
import software.amazon.awssdk.services.s3control.model.TagResourceResponse;

public class TagResourceExample {
    public static void tagResourceExample() {
        S3ControlClient s3Control =
 S3ControlClient.builder().region(Region.US_WEST_2).build();

        TagResourceRequest tagResourceRequest = TagResourceRequest.builder()
                .resourceArn("arn:aws:s3express:us-west-2:111122223333:bucket/my-
directory-bucket--usw2-az1--x-s3")
                .accountId("111122223333")
                .tags(Tag.builder().key("MyTagKey").value("MyTagValue").build())
                .build();

        TagResourceResponse response = s3Control.tagResource(tagResourceRequest);
        System.out.println("Status code (should be 204):");
        System.out.println(response.sdkHttpResponse().statusCode());
    }
}
```

Penggunaan API REST

Untuk informasi tentang dukungan Amazon S3 REST API untuk menambahkan tag ke bucket direktori, lihat bagian berikut di Referensi API Layanan Penyimpanan Sederhana Amazon:

- TagResource

Menggunakan AWS CLI

Untuk menginstal AWS CLI, lihat Menginstal AWS CLI di Panduan Pengguna.AWS Command Line Interface

Contoh CLI berikut menunjukkan cara menambahkan tag ke bucket direktori dengan menggunakan. AWS CLI Untuk menggunakan perintah ganti *user input placeholders* dengan informasi Anda sendiri.

Permintaan:

```
aws s3control tag-resource \
--account-id 111122223333 \
--resource-arn arn:aws:s3express:us-east-1:444455556666:bucket/prefix--use1-az4--x-s3 \
```

```
--tags "Key=mykey,Value=myvalue"
```

Tanggapan:

```
{
  "ResponseMetadata": {
      "RequestId": "EXAMPLE123456789",
      "HTTPStatusCode": 200,
      "HTTPHeaders": {
          "date": "Wed, 19 Jun 2025 10:30:00 GMT",
          "content-length": "0"
      },
      "RetryAttempts": 0
  }
}
```

Melihat tag bucket direktori

Anda dapat melihat atau mencantumkan tag yang diterapkan ke bucket direktori S3. Untuk informasi selengkapnya tentang tag, lihat Menggunakan tag dengan ember direktori S3.

Izin

Untuk melihat tag yang diterapkan ke bucket direktori, Anda harus memiliki izin berikut:

- `s3express:ListTagsForResource`

Memecahkan masalah kesalahan

Jika Anda mengalami kesalahan saat mencoba membuat daftar atau melihat tag bucket direktori, Anda dapat melakukan hal berikut:

- Verifikasi bahwa Anda memiliki persyaratan Izin untuk melihat atau mencantumkan tag bucket direktori.

Langkah-langkah

Anda dapat melihat tag yang diterapkan ke bucket direktori dengan menggunakan konsol Amazon S3,AWS CLI(), AWS Command Line Interface Amazon S3 REST API, dan. AWS SDKs

Menggunakan konsol S3

Untuk melihat tag yang diterapkan ke bucket direktori menggunakan konsol Amazon S3:

1. Masuk ke konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih bucket direktori.
3. Pilih nama bucket.
4. Pilih tab Properti.
5. Gulir ke bagian Tag untuk melihat semua tag yang diterapkan ke bucket direktori.
6. Bagian Tag menunjukkan tag yang ditentukan pengguna secara default. Anda dapat memilih tab tag AWS yang dihasilkan untuk melihat tag yang diterapkan ke bucket direktori menurut AWS layanan.

Menggunakan AWS SDKs

Bagian ini memberikan contoh cara melihat tag yang diterapkan ke bucket direktori dengan menggunakan AWS SDKs.

SDK for Java 2.x

Contoh ini menunjukkan cara melihat tag yang diterapkan ke bucket direktori dengan menggunakan AWS SDK for Java 2.x.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.ListTagsForResourceRequest;
import software.amazon.awssdk.services.s3control.model.ListTagsForResourceResponse;

public class ListTagsForResourceExample {
    public static void listTagsForResourceExample() {
        S3ControlClient s3Control =
 S3ControlClient.builder().region(Region.US_WEST_2).build();

        ListTagsForResourceRequest listTagsForResourceRequest =
 ListTagsForResourceRequest.builder()
                .resourceArn("arn:aws:s3express:us-west-2:111122223333:bucket/my-
directory-bucket--usw2-az1--x-s3")
                .accountId("111122223333")
                .build();
```

```
        ListTagsForResourceResponse response =
 s3Control.listTagsForResource(listTagsForResourceRequest);
        System.out.println("Tags on my resource:");
        System.out.println(response.toString());
    }
}
```

Penggunaan API REST

Untuk informasi tentang dukungan Amazon S3 REST API untuk melihat tag yang diterapkan ke bucket direktori, lihat bagian berikut di Referensi API Layanan Penyimpanan Sederhana Amazon:

- ListTagsforResource

Menggunakan AWS CLI

Untuk menginstal AWS CLI, lihat Menginstal AWS CLI di Panduan Pengguna.AWS Command Line Interface

Contoh CLI berikut menunjukkan cara melihat tag yang diterapkan ke bucket direktori. Untuk menggunakan perintah ganti *user input placeholders* dengan informasi Anda sendiri.

Permintaan:

```
aws s3control list-tags-for-resource \
--account-id 111122223333 \
--resource-arn arn:aws:s3express:us-east-1:444455556666:bucket/prefix--use1-az4--x-s3 \
```

Respons - tag hadir:

```
{
  "Tags": [
      {
          "Key": "MyKey1",
          "Value": "MyValue1"
      },
      {
          "Key": "MyKey2",
          "Value": "MyValue2"
      },
      {
```

```
            "Key": "MyKey3",
            "Value": "MyValue3"
        }
    ]
}
```

Tanggapan - tidak ada tag yang ada:

```
{
  "Tags": []
}
```

Menghapus tag dari bucket direktori

Anda dapat menghapus tag dari ember direktori S3. AWS Tag adalah pasangan nilai kunci yang menyimpan metadata tentang sumber daya, dalam hal ini bucket direktori Amazon S3. Untuk informasi selengkapnya tentang tag, lihat Menggunakan tag dengan ember direktori S3.

> **Note**
>
> Jika Anda menghapus tag dan kemudian mengetahui bahwa tag tersebut digunakan untuk melacak biaya atau untuk kontrol akses, Anda dapat menambahkan tag kembali ke bucket direktori.

Izin

Untuk menghapus tag dari bucket direktori, Anda harus memiliki izin berikut:

- `s3express:UntagResource`

Langkah-langkah

Anda dapat menghapus tag dari bucket direktori dengan menggunakan konsol Amazon S3,AWS CLI(), AWS Command Line Interface Amazon S3 REST API, dan. AWS SDKs

Menggunakan konsol S3

Untuk menghapus tag dari bucket direktori menggunakan konsol Amazon S3:

1. Masuk ke konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih bucket direktori.
3. Pilih nama bucket.
4. Pilih tab Properti.
5. Gulir ke bagian Tag dan pilih kotak centang di sebelah tag atau tag yang ingin Anda hapus.
6. Pilih Hapus.
7. Pop-up Hapus tag yang ditentukan pengguna muncul dan meminta Anda untuk mengonfirmasi tag atau tag yang Anda pilih.
8. Pilih Hapus untuk mengonfirmasi.

Menggunakan AWS SDKs

SDK for Java 2.x

Contoh ini menunjukkan cara menghapus tag dari bucket direktori dengan menggunakan file AWS SDK for Java 2.x. Untuk menggunakan perintah ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.UntagResourceRequest;
import software.amazon.awssdk.services.s3control.model.UntagResourceResponse;

public class UntagResourceExample {
    public static void untagResourceExample() {
        S3ControlClient s3Control =
 S3ControlClient.builder().region(Region.US_WEST_2).build();

        UntagResourceRequest untagResourceRequest = UntagResourceRequest.builder()
                .resourceArn("arn:aws:s3express:us-west-2:111122223333:bucket/my-
directory-bucket--usw2-az1--x-s3")
                .accountId("111122223333")
                .tagKeys("myTagKey")
                .build();

        UntagResourceResponse response =
 s3Control.untagResource(untagResourceRequest);
        System.out.println("Status code (should be 204):");
        System.out.println(response.sdkHttpResponse().statusCode());
```

```
    }
}
```

Penggunaan API REST

Untuk informasi tentang dukungan Amazon S3 REST API untuk menghapus tag dari bucket direktori, lihat bagian berikut di Referensi API Layanan Penyimpanan Sederhana Amazon:

- UnTagResource

Menggunakan AWS CLI

Untuk menginstal AWS CLI, lihat Menginstal AWS CLI di Panduan Pengguna.AWS Command Line Interface

Contoh CLI berikut menunjukkan cara menghapus tag dari bucket direktori dengan menggunakan. AWS CLI Untuk menggunakan perintah ganti *user input placeholders* dengan informasi Anda sendiri.

Permintaan:

```
aws s3control untag-resource \
--account-id 111122223333 \
--resource-arn arn:aws:s3express:us-east-1:444455556666:bucket/prefix--use1-az4--x-s3 \
--tag-keys "tagkey1" "tagkey2"
```

Tanggapan:

```
{
  "ResponseMetadata": {
    "RequestId": "EXAMPLE123456789",
    "HTTPStatusCode": 204,
    "HTTPHeaders": {
        "date": "Wed, 19 Jun 2025 10:30:00 GMT",
        "content-length": "0"
    },
    "RetryAttempts": 0
  }
}
```

Bekerja dengan Tabel Amazon S3 dan ember meja

Tabel Amazon S3 menyediakan penyimpanan S3 yang dioptimalkan untuk beban kerja analitik, dengan fitur yang dirancang untuk terus meningkatkan kinerja kueri dan mengurangi biaya penyimpanan untuk tabel. Tabel S3 dibuat khusus untuk menyimpan data tabular, seperti transaksi pembelian harian, data sensor streaming, atau tayangan iklan. Data tabular mewakili data dalam kolom dan baris, seperti dalam tabel database.

Data dalam Tabel S3 disimpan dalam tipe bucket baru: bucket tabel, yang menyimpan tabel sebagai subresource. Bucket meja mendukung penyimpanan tabel di Apache Iceberg format. Menggunakan pernyataan SQL standar, Anda dapat menanyakan tabel Anda dengan mesin kueri yang mendukung Iceberg, seperti Amazon Athena, Amazon Redshift, dan Apache Spark.

Topik

- Fitur Tabel S3
- Layanan terkait
- Tutorial: Memulai dengan Tabel S3
- Ember meja
- Pemeliharaan Tabel S3
- Ruang nama tabel
- Tabel dalam ember meja S3
- Mengakses data tabel
- Tabel S3 Wilayah AWS, titik akhir, dan kuota layanan
- Keamanan untuk Tabel S3
- Logging dengan AWS CloudTrail untuk Tabel S3

Fitur Tabel S3

Penyimpanan yang dibuat khusus untuk tabel

Bucket meja S3 dirancang khusus untuk tabel. Bucket tabel menyediakan transaksi per detik (TPS) yang lebih tinggi dan throughput kueri yang lebih baik dibandingkan dengan tabel yang dikelola sendiri di bucket tujuan umum S3. Bucket meja memberikan daya tahan, ketersediaan, dan skalabilitas yang sama dengan jenis bucket Amazon S3 lainnya.

Dukungan bawaan untuk Apache Iceberg

Tabel di ember meja Anda disimpan di Apache Icebergformat. Anda dapat menanyakan tabel ini menggunakan SQL standar di mesin kueri yang mendukung Iceberg. Iceberg memiliki berbagai fitur untuk mengoptimalkan kinerja kueri, termasuk evolusi skema dan evolusi partisi.

Dengan Iceberg, Anda dapat mengubah bagaimana data Anda diatur sehingga dapat berkembang dari waktu ke waktu tanpa mengharuskan Anda untuk menulis ulang kueri Anda atau membangun kembali struktur data Anda. Iceberg dirancang untuk membantu memastikan konsistensi dan keandalan data melalui dukungannya untuk transaksi. Untuk membantu Anda memperbaiki masalah atau melakukan kueri perjalanan waktu, Anda dapat melacak bagaimana data berubah dari waktu ke waktu dan memutar kembali ke versi historis.

Optimalisasi tabel otomatis

Untuk mengoptimalkan tabel Anda untuk kueri, S3 terus melakukan operasi pemeliharaan otomatis, seperti pemadatan, manajemen snapshot, dan penghapusan file yang tidak direferensikan. Operasi ini meningkatkan kinerja tabel dengan memadatkan objek yang lebih kecil menjadi file yang lebih sedikit dan lebih besar. Operasi pemeliharaan juga mengurangi biaya penyimpanan Anda dengan membersihkan benda yang tidak terpakai. Pemeliharaan otomatis ini merampingkan pengoperasian data lake dalam skala besar dengan mengurangi kebutuhan akan pemeliharaan meja manual. Untuk setiap keranjang meja dan meja, Anda dapat menyesuaikan konfigurasi pemeliharaan.

Manajemen akses dan keamanan

Anda dapat mengelola akses untuk bucket tabel dan tabel individual dengan AWS Identity and Access Management (IAM) dan Kebijakan Kontrol Layanan di. AWS OrganizationsTabel S3 menggunakan namespace layanan yang berbeda dari Amazon S3: namespace s3tables. Oleh karena itu, Anda dapat merancang kebijakan khusus untuk layanan Tabel S3 dan sumber dayanya. Anda dapat merancang kebijakan untuk memberikan akses ke tabel individual, semua tabel dalam namespace tabel, atau seluruh bucket tabel. Semua pengaturan Akses Publik Blok Amazon S3 selalu diaktifkan untuk bucket tabel dan tidak dapat dinonaktifkan.

Integrasi dengan layanan AWS analitik

Anda dapat secara otomatis mengintegrasikan bucket tabel Amazon S3 Anda dengan Amazon SageMaker Lakehouse melalui konsol S3. Integrasi ini memungkinkan layanan AWS analitik untuk secara otomatis menemukan dan mengakses data tabel Anda melalui AWS Glue Data Catalog. Setelah integrasi, Anda dapat bekerja dengan tabel Anda menggunakan layanan analitik seperti Amazon Athena, Amazon Redshift QuickSight, dan banyak lagi. Untuk informasi

selengkapnya tentang cara kerja integrasi, lihatMenggunakan Tabel Amazon S3 dengan layanan analitik AWS.

Layanan terkait

Anda dapat menggunakan yang berikut ini Layanan AWS dengan Tabel S3 untuk mendukung aplikasi analitik spesifik Anda.

- Amazon Athena - Athena adalah layanan kueri interaktif yang dapat Anda gunakan untuk menganalisis data secara langsung di Amazon S3 dengan menggunakan SQL standar. Anda juga dapat menggunakan Athena untuk menjalankan analisis data secara interaktif dengan menggunakan Apache Spark tanpa harus merencanakan, mengkonfigurasi, atau mengelola sumber daya. Saat Anda berlari Apache Spark aplikasi di Athena, Anda kirimkan Spark kode untuk memproses dan menerima hasilnya secara langsung.
- AWS Glue— AWS Glue adalah layanan integrasi data tanpa server yang memungkinkan Anda menemukan, menyiapkan, memindahkan, dan mengintegrasikan data dari berbagai sumber. Anda dapat menggunakan AWS Glue untuk analitik, pembelajaran mesin (ML), dan pengembangan aplikasi. AWS Glue juga mencakup produktivitas tambahan dan perkakas operasi data untuk menulis, menjalankan pekerjaan, dan mengimplementasikan alur kerja bisnis.
- Amazon EMR - Amazon EMR adalah platform cluster terkelola yang menyederhanakan menjalankan kerangka kerja data besar, seperti Apache Hadoop and Apache Spark, AWS untuk memproses dan menganalisis sejumlah besar data.
- Amazon Redshift — Amazon Redshift adalah layanan gudang data skala petabyte di cloud. Anda dapat menggunakan Amazon Redshift Tanpa Server untuk mengakses dan menganalisis data tanpa semua konfigurasi gudang data yang disediakan. Sumber daya secara otomatis disediakan dan kapasitas gudang data diskalakan secara cerdas untuk memberikan kinerja yang cepat bahkan untuk beban kerja yang paling menuntut dan tidak dapat diprediksi. Anda tidak dikenakan biaya saat gudang data menganggur, jadi Anda hanya membayar untuk apa yang Anda gunakan. Anda dapat memuat data dan mulai melakukan kueri segera di editor kueri Amazon Redshift v2 atau di alat intelijen bisnis (BI) favorit Anda.
- QuickSight— QuickSight adalah layanan analisis bisnis untuk membangun visualisasi, melakukan analisis ad hoc, dan dengan cepat mendapatkan wawasan bisnis dari data Anda. QuickSight menemukan sumber AWS data dengan mulus dan memberikan kinerja kueri yang cepat dan responsif dengan menggunakan QuickSight Super-cepat, Paralel, Dalam Memori, Mesin Perhitungan (SPICE).

- AWS Lake Formation— Lake Formation adalah layanan terkelola yang merampingkan proses untuk mengatur, mengamankan, dan mengelola danau data Anda. Lake Formation membantu Anda menemukan sumber data Anda dan kemudian membuat katalog, membersihkan, dan mengubah data. Dengan Lake Formation, Anda dapat mengelola kontrol akses berbutir halus untuk data lake data Anda di Amazon S3 dan metadatanya di. AWS Glue Data Catalog

Tutorial: Memulai dengan Tabel S3

Dalam tutorial ini, Anda membuat keranjang tabel dan mengintegrasikan bucket tabel di Wilayah Anda dengan layanan AWS analitik. Selanjutnya, Anda akan menggunakan AWS CLI untuk membuat namespace dan tabel pertama Anda di ember meja Anda. Kemudian, Anda gunakan AWS Lake Formation untuk memberikan izin di meja Anda, sehingga Anda dapat mulai menanyakan tabel Anda dengan Athena.

> **Tip**
>
> Jika Anda memigrasikan data tabular dari bucket tujuan umum ke bucket tabel, Perpustakaan AWS Solusi memiliki solusi terpandu untuk membantu Anda. Solusi ini mengotomatiskan pemindahan Apache Iceberg dan Apache Hive tabel yang terdaftar AWS Glue Data Catalog dan disimpan dalam ember tujuan umum ke ember meja dengan menggunakan dan AWS Step Functions Amazon EMR dengan. Apache Spark Untuk informasi selengkapnya, lihat Panduan untuk Memigrasi Data Tabular dari Amazon S3 ke Tabel S3 di Pustaka Solusi. AWS

Topik

- Langkah 1: Buat keranjang meja dan integrasikan dengan layanan AWS analitik
- Langkah 2: Buat namespace tabel dan tabel
- (Opsional) Langkah 3: Berikan izin Lake Formation di meja Anda
- Langkah 4: Kueri data dengan SQL di Athena

Langkah 1: Buat keranjang meja dan integrasikan dengan layanan AWS analitik

Pada langkah ini, Anda menggunakan konsol Amazon S3 untuk membuat bucket tabel pertama Anda. Untuk cara lain untuk membuat keranjang meja, lihat Membuat ember meja.

> **Note**
>
> Secara default, konsol Amazon S3 secara otomatis mengintegrasikan bucket tabel Anda dengan SageMaker Amazon Lakehouse, yang AWS memungkinkan layanan analitik menemukan dan mengakses data Tabel S3 Anda secara otomatis. Jika Anda membuat bucket tabel pertama secara terprogram menggunakan AWS Command Line Interface (AWS CLI),, atau REST API AWS SDKs, Anda harus menyelesaikan integrasi layanan AWS analitik secara manual. Untuk informasi selengkapnya, lihat Menggunakan Tabel Amazon S3 dengan layanan analitik AWS.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi di bagian atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah tempat Anda ingin membuat keranjang tabel.
3. Di panel navigasi kiri, pilih Bucket tabel.
4. Pilih Buat ember meja.
5. Di bawah Konfigurasi umum, masukkan nama untuk bucket tabel Anda.

   Nama keranjang meja harus:

   - Jadilah unik di dalam untuk Anda Akun AWS di Wilayah saat ini.
   - Panjangnya antara 3 hingga 63 karakter.
   - Hanya terdiri dari huruf kecil, angka, dan tanda hubung (). -
   - Dimulai dan diakhiri dengan huruf atau angka.

   Setelah Anda membuat bucket tabel, Anda tidak dapat mengubah namanya. Akun AWS Yang menciptakan ember meja memilikinya. Untuk informasi selengkapnya tentang penamaan bucket tabel, lihatAturan penamaan ember meja.
6. Di bagian Integrasi dengan layanan AWS analitik, pastikan kotak centang Aktifkan integrasi dipilih.

   Jika Aktifkan integrasi dipilih saat Anda membuat bucket tabel pertama menggunakan konsol, Amazon S3 mencoba mengintegrasikan bucket tabel Anda dengan layanan AWS analitik. Integrasi ini memungkinkan Anda menggunakan layanan AWS analitik untuk mengakses semua

tabel di Wilayah saat ini. Untuk informasi selengkapnya, lihat Menggunakan Tabel Amazon S3 dengan layanan analitik AWS.

7. Pilih Buat bucket.

Langkah 2: Buat namespace tabel dan tabel

Untuk langkah ini, Anda membuat namespace di bucket tabel Anda, dan kemudian membuat tabel baru di bawah namespace itu. Anda dapat membuat namespace tabel dan tabel dengan menggunakan konsol atau. AWS CLI

> **Important**
>
> Saat membuat tabel, pastikan Anda menggunakan semua huruf kecil dalam nama tabel dan definisi tabel Anda. Misalnya, pastikan bahwa nama kolom Anda semua huruf kecil. Jika nama tabel atau definisi tabel berisi huruf kapital, tabel tidak didukung oleh AWS Lake Formation atau definisi tabel AWS Glue Data Catalog. Dalam hal ini, tabel Anda tidak akan terlihat oleh layanan AWS analitik seperti Amazon Athena, meskipun bucket tabel Anda terintegrasi dengan AWS layanan analitik.
>
> Jika definisi tabel berisi huruf kapital, Anda menerima pesan galat berikut saat menjalankan `SELECT` kueri di Athena: “GENERIC_INTERNAL_ERROR: Get table request failed: com.amazonaws.services.glue.model. ValidationException: Sumber Daya Federasi Tidak Didukung - Nama tabel atau kolom tidak valid.

Menggunakan konsol S3 dan Amazon Athena

Prosedur berikut menggunakan konsol Amazon S3 untuk membuat namespace dan tabel dengan Amazon Athena.

Untuk membuat namespace tabel dan tabel

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tabel.
3. Pada halaman Bucket Table, pilih bucket tabel tempat Anda ingin membuat tabel.
4. Pada halaman detail bucket tabel, pilih Buat tabel dengan Athena.

5. Dalam kotak dialog Buat tabel dengan Athena, pilih Buat namespace, lalu masukkan nama di bidang Nama Namespace. Nama namespace harus 1 hingga 255 karakter dan unik di dalam keranjang tabel. Karakter yang valid adalah a—z, 0-9, dan garis bawah (). _ Underscore tidak diizinkan pada awal nama namespace.
6. Pilih Buat namespace.
7. Pilih Buat tabel dengan Athena.
8. Konsol Amazon Athena terbuka dan editor kueri Athena muncul. Editor kueri diisi dengan kueri sampel yang dapat Anda gunakan untuk membuat tabel. Ubah kueri untuk menentukan nama tabel dan kolom yang Anda ingin tabel Anda miliki.
9. Setelah selesai memodifikasi kueri, pilih Jalankan untuk membuat tabel Anda.

Jika pembuatan tabel Anda berhasil, nama tabel baru Anda muncul di daftar tabel di Athena. Saat Anda menavigasi kembali ke konsol Amazon S3, tabel baru akan muncul di daftar Tabel di halaman detail untuk bucket tabel setelah Anda menyegarkan daftar.

Menggunakan AWS CLI

Untuk menggunakan AWS CLI contoh perintah berikut untuk membuat namespace di bucket tabel Anda, lalu buat tabel baru dengan skema di bawah namespace itu, ganti nilainya dengan milik Anda. *user input placeholder*

Prasyarat

- Lampirkan AmazonS3TablesFullAccesskebijakan ke identitas IAM Anda.
- Instal AWS CLI versi 2.23.10 atau lebih tinggi. Untuk informasi selengkapnya, lihat Menginstal atau memperbarui versi terbaru dari Panduan AWS Command Line Interface Pengguna. AWS CLI

1. Buat namespace baru di bucket tabel Anda dengan menjalankan perintah berikut:

```
aws s3tables create-namespace \
--table-bucket-arn arn:aws:s3tables:us-east-1:111122223333:bucket/amzn-s3-demo-table-bucket \
--namespace my_namespace
```

   - Konfirmasikan bahwa namespace Anda berhasil dibuat dengan menjalankan perintah berikut:

```
aws s3tables list-namespaces \
--table-bucket-arn arn:aws:s3tables:us-east-1:111122223333:bucket/amzn-s3-demo-table-bucket
```

2. Buat tabel baru dengan skema tabel dengan menjalankan perintah berikut:

```
aws s3tables create-table --cli-input-json file://mytabledefinition.json
```

Untuk `mytabledefinition.json` file, gunakan definisi tabel contoh berikut:

```
{
    "tableBucketARN": "arn:aws:s3tables:us-east-1:111122223333:bucket/amzn-s3-demo-table-bucket",
    "namespace": "my_namespace",
    "name": "my_table",
    "format": "ICEBERG",
    "metadata": {
        "iceberg": {
            "schema": {
                "fields": [
                     {"name": "id", "type": "int","required": true},
                     {"name": "name", "type": "string"},
                     {"name": "value", "type": "int"}
                ]
            }
        }
    }
}
```

(Opsional) Langkah 3: Berikan izin Lake Formation di meja Anda

Untuk langkah ini, Anda memberikan izin Lake Formation pada tabel baru Anda ke prinsipal IAM lainnya. Izin ini memungkinkan prinsipal selain Anda mengakses sumber daya bucket tabel dengan menggunakan Athena dan layanan analitik lainnya. AWS Untuk informasi selengkapnya, lihat Memberikan izin pada tabel atau database. Jika Anda satu-satunya pengguna yang akan mengakses tabel Anda, Anda dapat melewati langkah ini.

1. Buka AWS Lake Formation konsol dihttps://console.aws.amazon.com/lakeformation/, dan masuk sebagai administrator danau data. Untuk informasi selengkapnya tentang cara membuat administrator data lake, lihat Membuat administrator data lake.
2. Di panel navigasi, pilih Izin data lalu pilih Hibah.
3. Pada halaman Hibah Izin, di bawah Prinsipal, pilih pengguna dan peran IAM dan pilih pengguna IAM atau peran yang ingin Anda izinkan untuk menjalankan kueri di tabel Anda.
4. Di bawah LF-tag atau sumber katalog, pilih Sumber daya Katalog Data Bernama.
5. Lakukan salah satu hal berikut, tergantung pada apakah Anda ingin memberikan akses ke semua tabel di akun Anda atau apakah Anda ingin memberikan akses hanya ke sumber daya dalam keranjang tabel yang Anda buat:
   - Untuk Katalog, pilih katalog tingkat akun yang Anda buat saat mengintegrasikan bucket tabel. Misalnya, *111122223333*`:s3tablescatalog`.
   - Untuk Katalog, pilih subkatalog untuk ember meja Anda. Misalnya, *111122223333*`:s3tablescatalog/`*amzn-s3-demo-table-bucket*.
6. (Opsional) Jika Anda memilih subkatalog untuk bucket tabel Anda, lakukan salah satu atau kedua hal berikut:
   - Untuk Database, pilih namespace bucket tabel yang Anda buat.
   - Untuk Tabel, pilih tabel yang Anda buat di keranjang tabel, atau pilih Semua tabel.
7. Bergantung pada apakah Anda memilih katalog atau subkatalog dan bergantung pada apakah Anda kemudian memilih database atau tabel, Anda dapat mengatur izin di tingkat katalog, database, atau tabel. Untuk informasi selengkapnya tentang izin Lake Formation, lihat Mengelola izin Lake Formation di Panduan AWS Lake Formation Pengembang.

   Lakukan salah satu tindakan berikut:
   - Untuk izin Katalog, pilih Super untuk memberikan prinsipal lainnya semua izin di katalog Anda, atau pilih izin yang lebih halus, seperti Jelaskan.
   - Untuk izin Database, Anda tidak dapat memilih Super untuk memberikan prinsipal lainnya semua izin pada database Anda. Sebagai gantinya, pilih lebih banyak izin berbutir halus, seperti Deskripsikan.
   - Untuk izin Tabel, pilih Super untuk memberikan prinsipal lainnya semua izin di tabel Anda, atau pilih izin yang lebih halus, seperti Pilih atau Jelaskan.

> **Note**
>
> Saat Anda memberikan izin Lake Formation pada sumber daya Katalog Data ke akun eksternal atau langsung ke kepala IAM di akun lain, Lake Formation menggunakan layanan AWS Resource Access Manager (AWS RAM) untuk membagikan sumber daya. Jika akun penerima hibah berada di organisasi yang sama dengan akun pemberi hibah, sumber daya bersama segera tersedia untuk penerima hibah. Jika akun penerima hibah tidak berada di organisasi yang sama, AWS RAM kirimkan undangan ke akun penerima hibah untuk menerima atau menolak hibah sumber daya. Kemudian, untuk membuat sumber daya bersama tersedia, administrator data lake di akun penerima hibah harus menggunakan AWS RAM konsol atau AWS CLI untuk menerima undangan. Untuk informasi selengkapnya tentang berbagi data lintas akun, lihat Berbagi data lintas akun di Lake Formation di Panduan AWS Lake Formation Pengembang.

8. PilihIzin.

Langkah 4: Kueri data dengan SQL di Athena

Anda dapat menanyakan tabel Anda dengan SQL di Athena. Athena mendukung kueri Data Definition Language (DDL), Data Manipulation Language (DHTML), dan Data Query Language (DQL) untuk Tabel S3.

Anda dapat mengakses kueri Athena baik dari konsol Amazon S3 atau melalui konsol Amazon Athena.

Menggunakan konsol S3 dan Amazon Athena

Prosedur berikut menggunakan konsol Amazon S3 untuk mengakses editor kueri Athena sehingga Anda dapat menanyakan tabel dengan Amazon Athena.

Untuk menanyakan tabel

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tabel.
3. Pada halaman bucket Table, pilih bucket tabel yang berisi tabel yang ingin Anda kueri.

4. Pada halaman detail keranjang tabel, pilih tombol opsi di sebelah nama tabel yang ingin Anda kueri.
5. Pilih tabel Query dengan Athena.
6. Konsol Amazon Athena terbuka dan editor kueri Athena muncul dengan contoh `SELECT` kueri yang dimuat untuk Anda. Ubah kueri ini sesuai kebutuhan untuk kasus penggunaan Anda.
7. Untuk menjalankan kueri, pilih Jalankan.

Menggunakan konsol Amazon Athena

Untuk menanyakan tabel

1. Buka konsol Athena di https://console.aws.amazon.com/athena/.
2. Kueri tabel Anda. Berikut ini adalah contoh query yang dapat Anda modifikasi. Pastikan untuk mengganti *`user input placeholders`* dengan informasi Anda sendiri.

```
SELECT * FROM "s3tablescatalog/amzn-s3-demo-table-bucket"."my_namespace"."my_table"
 LIMIT 10
```

3. Untuk menjalankan kueri, pilih Jalankan.

Ember meja

Bucket tabel Amazon S3 adalah jenis bucket S3 yang dapat Anda gunakan untuk membuat dan menyimpan tabel sebagai sumber daya S3. Bucket tabel digunakan untuk menyimpan data tabular dan metadata sebagai objek untuk digunakan dalam beban kerja analitik. S3 melakukan perawatan di bucket meja Anda secara otomatis untuk membantu mengurangi biaya penyimpanan meja Anda. Untuk informasi selengkapnya, lihat Pemeliharaan untuk ember meja.

Untuk berinteraksi dengan tabel yang disimpan di dalam bucket tabel, Anda dapat mengintegrasikan bucket tabel Anda dengan aplikasi analitik yang mendukung Apache Iceberg. Bucket tabel terintegrasi dengan layanan AWS analitik melalui. AWS Glue Data Catalog Untuk informasi selengkapnya, lihat Menggunakan Tabel Amazon S3 dengan layanan analitik AWS. Anda juga dapat berinteraksi dengan tabel menggunakan mesin kueri sumber terbuka menggunakan Katalog Tabel Amazon S3 untuk Apache IcebergUntuk informasi selengkapnya, lihat Mengakses tabel menggunakan Tabel Amazon S3 Iceberg REST titik akhir.

Setiap keranjang tabel memiliki Nama Sumber Daya Amazon (ARN) unik dan kebijakan sumber daya yang dilampirkan padanya. Table bucket ARNs ikuti format ini:

```
arn:aws:s3tables:Region:OwnerAccountID:bucket/bucket-name
```

Semua ember meja dan meja bersifat pribadi dan tidak dapat dipublikasikan. Sumber daya ini hanya dapat diakses oleh pengguna yang secara eksplisit diberikan akses. Untuk memberikan akses, Anda dapat menggunakan kebijakan berbasis sumber daya IAM untuk bucket dan tabel tabel, serta kebijakan berbasis identitas IAM untuk pengguna dan peran.

Secara default, Anda dapat membuat hingga 10 bucket tabel per Wilayah AWS dalam file Akun AWS. Untuk meminta peningkatan kuota untuk bucket atau tabel tabel, hubungi. Dukungan

Ada beberapa jenis bucket Amazon S3. Sebelum membuat bucket, pastikan Anda memilih jenis bucket yang paling sesuai dengan aplikasi dan persyaratan kinerja Anda. Untuk informasi lebih lanjut tentang berbagai jenis bucket dan kasus penggunaan yang sesuai untuk masing-masing, lihatBucket.

Topik

- Aturan penamaan bucket, tabel, dan namespace Amazon S3
- Membuat ember meja
- Menghapus ember meja
- Melihat detail tentang ember meja Amazon S3
- Mengelola kebijakan bucket tabel

Aturan penamaan bucket, tabel, dan namespace Amazon S3

Saat Anda membuat keranjang tabel, Anda memilih nama bucket dan Wilayah AWS, nama tersebut harus unik untuk akun Anda di Wilayah yang dipilih. Setelah membuat bucket tabel, Anda tidak dapat mengubah nama bucket atau Region. Nama keranjang tabel harus mengikuti aturan penamaan tertentu. Untuk informasi selengkapnya tentang aturan penamaan untuk bucket tabel dan tabel serta ruang nama di dalamnya, lihat topik berikut.

Topik

- Aturan penamaan ember meja
- Aturan penamaan untuk tabel dan ruang nama

Aturan penamaan ember meja

Saat membuat bucket tabel Amazon S3, Anda menentukan nama bucket tabel. Seperti jenis bucket lainnya, bucket meja tidak dapat diganti namanya. Tidak seperti jenis bucket lainnya, bucket tabel tidak berada dalam namespace global, jadi setiap nama bucket di akun Anda harus unik hanya di Region Anda saat ini. AWS

Untuk aturan penamaan bucket bertujuan umum, lihat Aturan penamaan bucket tujuan umum. Untuk aturan penamaan bucket direktori, lihatAturan penamaan bucket direktori.

Aturan penamaan berikut berlaku untuk bucket tabel.

- Panjang nama bucket harus antara 3 dan 63 karakter.
- Nama bucket hanya dapat terdiri dari huruf kecil, angka, dan tanda hubung (). -
- Nama bucket harus diawali dan juga diakhiri dengan huruf atau, nomor.
- Nama bucket tidak boleh mengandung garis bawah (_) atau periode (.).
- Nama bucket tidak boleh dimulai dengan awalan cadangan berikut:
  - `xn--`
  - `sthree-`
  - `amzn-s3-demo-`
- Nama bucket tidak boleh diakhiri dengan salah satu sufiks cadangan berikut:
  - `-s3alias`
  - `--ol-s3`
  - `--x-s3`
  - `--table-s3`

Aturan penamaan untuk tabel dan ruang nama

Aturan penamaan berikut berlaku untuk tabel dan ruang nama dalam bucket tabel:

- Panjang nama harus antara 1 dan 255 karakter.
- Nama hanya dapat terdiri dari huruf kecil, angka, dan garis bawah (). _ Underscore tidak diizinkan di awal atau akhir nama namespace.
- Nama harus dimulai dan diakhiri dengan huruf atau angka.
- Nama tidak boleh mengandung tanda hubung (-) atau titik (.).

- Nama tabel harus unik dalam namespace.
- Namespace harus unik di dalam ember meja.
- Nama namespace tidak boleh dimulai dengan awalan cadangan. `aws` Misalnya, Anda tidak dapat menggunakan `aws_s3_metadata` sebagai namespace. `aws_s3_metadata`adalah namespace yang dicadangkan untuk tabel metadata. Lihat informasi yang lebih lengkap di Mempercepat penemuan data dengan Metadata S3.

Membuat ember meja

Bucket tabel Amazon S3 adalah jenis bucket S3 yang dapat Anda gunakan untuk membuat dan menyimpan tabel sebagai sumber daya S3. Untuk mulai menggunakan Tabel S3, Anda membuat keranjang meja tempat Anda menyimpan dan mengelola tabel. Saat Anda membuat ember meja, Anda memilih nama ember dan Wilayah AWS. Nama bucket tabel harus unik untuk akun Anda di Wilayah yang dipilih. Setelah membuat bucket tabel, Anda tidak dapat mengubah nama bucket atau Region. Untuk informasi tentang penamaan ember tabel, lihatAturan penamaan bucket, tabel, dan namespace Amazon S3.

Bucket tabel memiliki format Amazon Resource Name (ARN) berikut:

```
arn:aws:s3tables:region:owner-account-id:bucket/bucket-name
```

Secara default, Anda dapat membuat hingga 10 bucket tabel per Wilayah dalam file Akun AWS. Untuk meminta peningkatan kuota untuk bucket atau tabel tabel, hubungi. Dukungan

Saat membuat bucket tabel, Anda dapat menentukan jenis enkripsi yang akan digunakan untuk mengenkripsi tabel yang Anda buat di bucket tersebut. Untuk informasi selengkapnya tentang opsi enkripsi bucket, lihatthe section called “Enkripsi”.

Prasyarat untuk membuat ember meja

Untuk membuat ember meja, Anda harus terlebih dahulu melakukan hal berikut:

- Pastikan Anda memiliki izin AWS Identity and Access Management (IAM) untuk. `s3tables:CreateTableBucket`

> **Note**
>
> Jika Anda memilih SSE-KMS sebagai jenis enkripsi default, Anda harus memiliki izin untuk`s3tables:PutTableBucketEncryption`, dan memiliki `DescribeKey` izin pada kunci yang dipilih. AWS KMS Selain itu AWS KMS kunci yang Anda gunakan perlu memberikan izin Tabel S3 untuk melakukan pemeliharaan tabel otomatis. Untuk informasi selengkapnya, lihat Persyaratan izin untuk enkripsi SSE-KMS Tabel S3

Untuk membuat bucket tabel, Anda dapat menggunakan konsol Amazon S3, Amazon S3 REST API,AWS CLI() AWS Command Line Interface , atau. AWS SDKs

Menggunakan konsol S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi di bagian atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah tempat Anda ingin membuat ember.
3. Di panel navigasi kiri, pilih Bucket tabel.
4. Pilih Buat bucket tabel untuk membuka halaman Bucket Create table.
5. Di bawah Properties, masukkan nama untuk bucket tabel Anda.

   Nama keranjang meja harus:

   - Jadilah unik untuk akun Anda di Wilayah saat ini.
   - Panjangnya antara 3 dan 63 karakter
   - Hanya terdiri dari huruf kecil, angka, dan tanda hubung (-).
   - Dimulai dan diakhiri dengan huruf atau angka.

   Setelah membuat bucket, Anda tidak dapat mengubah namanya. Akun AWS Yang menciptakan ember memilikinya. Untuk informasi tentang penamaan ember tabel, lihatAturan penamaan bucket, tabel, dan namespace Amazon S3.
6. Jika Anda ingin mengintegrasikan bucket tabel Anda dengan layanan AWS analitik, pastikan Aktifkan integrasi dipilih di bawah Integrasi dengan layanan AWS analitik.

> **Note**
>
> Saat Anda membuat bucket tabel pertama menggunakan konsol dengan opsi Aktifkan integrasi yang dipilih, Amazon S3 mencoba mengintegrasikan bucket tabel Anda secara otomatis dengan layanan AWS analitik. Integrasi ini memungkinkan Anda menggunakan layanan AWS analitik untuk menanyakan semua tabel di Wilayah saat ini. Untuk informasi selengkapnya, lihat Menggunakan Tabel Amazon S3 dengan layanan analitik AWS.

7. Untuk mengonfigurasi enkripsi default, di bawah Jenis enkripsi, pilih salah satu dari berikut ini:

   - Enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3)
   - Enkripsi sisi server dengan AWS Key Management Service kunci (SSE-KMS)

   Untuk informasi selengkapnya tentang opsi enkripsi untuk data tabel, lihatMelindungi data tabel S3 dengan enkripsi.
8. Pilih Buat bucket.

Menggunakan AWS CLI

Contoh ini menunjukkan cara membuat bucket tabel dengan menggunakan AWS CLI. Untuk menggunakan contoh ini, ganti *user input placeholders*dengan informasi Anda sendiri.

```
aws s3tables create-table-bucket \
    --region us-east-2 \
    --name amzn-s3-demo-bucket1
```

Secara default bucket tabel S3 menggunakan SSE-S3 sebagai pengaturan enkripsi default mereka, namun, Anda dapat menggunakan `--encryption-configuration` parameter opsional untuk menentukan jenis enkripsi yang berbeda. Contoh berikut menunjukkan cara membuat bucket yang menggunakan enkripsi SSE-KMS. Untuk informasi selengkapnya tentang setelan enkripsi untuk bucket tabel, lihatMelindungi data tabel S3 dengan enkripsi.

```
aws s3tables create-table-bucket \
    --region us-east-2 \
    --name amzn-s3-demo-bucket1
```

```
    --encryption-configuration '{
                    "sseAlgorithm": "aws:kms",
                    "kmsKeyArn":
 "arn:aws:kms:Region:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab" }'
```

Menghapus ember meja

Anda dapat menggunakan Amazon S3 APIs, AWS Command Line Interface, atau AWS SDKs untuk menghapus keranjang meja. Sebelum menghapus keranjang tabel, Anda harus menghapus semua ruang nama dan tabel di dalam bucket.

Menggunakan AWS CLI

Contoh ini menunjukkan cara menghapus bucket tabel dengan menggunakan file AWS CLI. Untuk menggunakan contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3tables delete-table-bucket \
    --region us-east-2 \
    --table-bucket-arn arn:aws:s3tables:us-east-1:111122223333:bucket/amzn-s3-demo-
bucket1
```

Melihat detail tentang ember meja Amazon S3

Anda dapat melihat detail bucket tabel Amazon S3 secara terprogram menggunakan REST API S3 Tables, atau. AWS CLI AWS SDKs

Menggunakan AWS CLI

Contoh ini menunjukkan cara mendapatkan detail tentang ember meja dengan menggunakan AWS CLI. Untuk menggunakan contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3tables get-table-bucket --table-bucket-arn arn:aws:s3tables:us-
east-1:111122223333:bucket/amzn-s3-demo-bucket1
```

Mengelola kebijakan bucket tabel

Anda dapat menambahkan, menghapus, memperbarui, dan melihat kebijakan bucket untuk bucket tabel Amazon S3 dengan menggunakan Amazon S3 REST API AWS SDKs, dan (). AWS Command Line Interface AWS CLI Untuk informasi selengkapnya, lihat topik berikut.

Untuk informasi selengkapnya, lihat topik berikut. Untuk informasi selengkapnya tentang tindakan dan kunci kondisi yang didukung AWS Identity and Access Management (IAM) untuk Tabel Amazon S3, lihat. Manajemen akses untuk Tabel S3 Misalnya kebijakan bucket untuk bucket tabel, lihatKebijakan berbasis sumber daya untuk Tabel S3.

Menambahkan kebijakan bucket tabel

Untuk menambahkan kebijakan bucket ke bucket tabel, gunakan AWS CLI contoh berikut.

Menggunakan AWS CLI

Contoh ini menunjukkan cara membuat kebijakan bucket tabel dengan menggunakan AWS CLI. Untuk menggunakan perintah, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3tables put-table-bucket-policy \
    --table-bucket-arn arn:aws:s3tables:us-east-1:111122223333:bucket/amzn-s3-demo-bucket1  \
    --resource-policy your-policy-JSON
```

Melihat kebijakan keranjang tabel

Untuk melihat kebijakan bucket yang dilampirkan ke bucket tabel, gunakan AWS CLI contoh berikut.

Menggunakan AWS CLI

Contoh ini menunjukkan cara melihat kebijakan yang dilampirkan ke bucket tabel dengan menggunakan AWS CLI. Untuk menggunakan perintah, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3tables get-table-bucket-policy --table-bucket-arn arn:aws:s3tables:us-east-1:111122223333:bucket/amzn-s3-demo-bucket1
```

Menghapus kebijakan bucket tabel

Untuk menghapus kebijakan bucket yang dilampirkan ke bucket tabel, gunakan AWS CLI contoh berikut.

Menggunakan AWS CLI

Contoh ini menunjukkan cara menghapus kebijakan bucket tabel dengan menggunakan AWS CLI. Untuk menggunakan perintah, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3tables delete-table-bucket-policy --table-bucket-arn arn:aws:s3tables:us-
east-1:111122223333:bucket/amzn-s3-demo-bucket1
```

Pemeliharaan Tabel S3

Amazon S3 secara otomatis melakukan pemeliharaan untuk meningkatkan kinerja tabel Anda di bucket tabel S3. Pemeliharaan dilakukan di ember meja dan tingkat meja individu dan meliputi yang berikut:

Perawatan tingkat ember meja:

- Penghapusan file yang tidak direferensikan - Membersihkan file yatim piatu untuk mengoptimalkan penggunaan dan biaya penyimpanan

Perawatan tingkat meja:

- Pemadatan file — Mengkonsolidasikan file kecil untuk meningkatkan kinerja kueri dan mengurangi biaya penyimpanan.
- Manajemen snapshot — Mengontrol riwayat versi tabel dan mencegah pertumbuhan metadata yang berlebihan.

Opsi ini diaktifkan secara default. Anda dapat mengedit atau menonaktifkan operasi ini melalui file konfigurasi pemeliharaan.

Topik

- Status pekerjaan pemeliharaan Tabel S3
- Pemeliharaan untuk ember meja
- Pemeliharaan untuk tabel
- Pertimbangan dan batasan untuk pekerjaan pemeliharaan

Status pekerjaan pemeliharaan Tabel S3

Pekerjaan pemeliharaan Tabel S3 berjalan secara berkala untuk tabel S3 atau ember meja Anda. Anda dapat menanyakan status pekerjaan ini dengan `GetTableMaintenanceJobStatus` API.

Untuk mendapatkan status pekerjaan pemeliharaan Anda dengan menggunakan AWS CLI

Contoh berikut akan mendapatkan status pekerjaan pemeliharaan menggunakan `GetTableMaintenanceJobStatus` API.

```
aws s3tables get-table-maintenance-job-status \
   --table-bucket-arn="arn:aws:s3tables:arn:aws::111122223333:bucket/amzn-s3-demo-bucket1" \
   --namespace="mynamespace" \
   --name="testtable"
```

Untuk informasi selengkapnya, lihat get-table-maintenance-job-status dalam AWS CLI Referensi Perintah.

Pekerjaan pemeliharaan Tabel S3 dapat bertransisi di antara empat kemungkinan status:

- `Successful`
- `Failed`
- `Disabled`
- `Not_Yet_Run`

Pekerjaan dengan status gagal akan menyertakan pesan kegagalan. Daftar berikut menjelaskan kemungkinan pesan kegagalan.

- Mengalami pengecualian validasi Iceberg saat mencoba membaca tabel. Pastikan tabel Anda dapat dibaca, mematuhi spesifikasi Iceberg, dan hanya berisi jalur S3 yang dimulai dengan alias Tabel S3 Anda.
- Manajemen Iceberg Snapshot saat ini tidak mendukung tag atau referensi yang ditentukan pengguna.
- Konfigurasi pemeliharaan tabel Iceberg tidak kompatibel dengan 'history.expire. max-snapshot-age-ms' dan 'history.expire. min-snapshots-to-keep' properti tabel.
- Manajemen snapshot gunung es dan penghapusan file yang tidak direferensikan tidak didukung ketika properti tabel 'gc.enabled' salah. Pastikan properti ini tidak disetel atau disetel secara eksplisit ke true.
- Gagal melakukan komit karena metadata kedaluwarsa. Pemeliharaan akan dicoba lagi pada kesempatan berikutnya yang tersedia.

- Akses yang tidak memadai untuk melakukan pemeliharaan meja. Pastikan kunci yang digunakan untuk mengenkripsi tabel aktif, ada, dan memiliki kebijakan sumber daya yang memberikan akses ke prinsipal layanan S3. `maintenance.s3tables.amazonaws.com`

> **Note**
>
> Untuk informasi selengkapnya tentang AWS KMS izin untuk Tabel S3, lihat. Persyaratan izin untuk enkripsi SSE-KMS Tabel S3

- Kesalahan internal

Pemeliharaan untuk ember meja

Amazon S3 menawarkan operasi pemeliharaan untuk meningkatkan manajemen dan kinerja bucket meja Anda. Opsi berikut diaktifkan secara default untuk semua bucket tabel. Anda dapat mengedit atau menonaktifkan opsi ini dengan menentukan file konfigurasi pemeliharaan untuk bucket tabel Anda.

Mengedit konfigurasi ini memerlukan `s3tables:PutTableBucketMaintenanceConfiguration` izin.

Topik

- Penghapusan file yang tidak direferensikan
- Pertimbangan dan keterbatasan

Penghapusan file yang tidak direferensikan

Penghapusan file yang tidak direferensikan mengidentifikasi dan menghapus semua objek yang tidak direferensikan oleh snapshot tabel apa pun. Sebagai bagian dari kebijakan penghapusan file yang tidak direferensikan, Anda dapat mengonfigurasi dua properti: `unreferencedDays` (3 hari secara default) dan `nonCurrentDays` (10 hari secara default).

Untuk objek apa pun yang tidak direferensikan oleh tabel Anda dan lebih tua dari `unreferencedDays` properti, S3 menandai objek sebagai noncurrent. S3 menghapus objek noncurrent setelah jumlah hari yang ditentukan oleh properti. `nonCurrentDays`

> **Note**
>
> Penghapusan objek noncurrent bersifat permanen tanpa cara untuk memulihkan objek ini.

Untuk melihat atau memulihkan objek yang telah ditandai sebagai noncurrent, Anda harus menghubungi AWS Dukungan. Untuk informasi tentang menghubungi AWS Dukungan, lihat Kontak AWS atau AWS Dukungan Dokumentasi.

Penghapusan file yang tidak direferensikan menentukan objek yang akan dihapus dari tabel Anda dengan referensi hanya ke tabel itu. Referensi apa pun yang dibuat untuk objek-objek ini di luar tabel tidak akan mencegah penghapusan file yang tidak direferensikan dari menghapus objek.

Jika Anda menonaktifkan penghapusan file yang tidak direferensikan, pekerjaan apa pun yang sedang berlangsung tidak akan terpengaruh. Konfigurasi baru akan berlaku untuk pekerjaan berikutnya setelah perubahan konfigurasi. Untuk informasi selengkapnya, lihat informasi harga di harga Amazon S3.

Anda hanya dapat mengonfigurasi penghapusan file yang tidak direferensikan di tingkat bucket tabel. Konfigurasi ini akan berlaku untuk setiap tabel di bucket Anda.

Untuk mengkonfigurasi penghapusan file yang tidak direferensikan dengan menggunakan AWS CLI

Contoh berikut akan mengatur `unreferencedDays` ke 4 hari dan `nonCurrentDays` ke 10 hari menggunakan `PutTableBucketMaintenanceConfiguration` API.

```
aws s3tables put-table-bucket-maintenance-configuration \
   --table-bucket-arn arn:aws:s3tables:us-east-1:111122223333:bucket/amzn-s3-demo-
table-bucket \
   --type icebergUnreferencedFileRemoval \
   --value '{"status":"enabled","settings":{"icebergUnreferencedFileRemoval":
{"unreferencedDays":4,"nonCurrentDays":10}}}'
```

Untuk informasi selengkapnya, lihat put-table-bucket-maintenance-configuration dalam AWS CLI Referensi Perintah.

Pertimbangan dan keterbatasan

Untuk mempelajari lebih lanjut tentang pertimbangan dan batasan tambahan untuk penghapusan file yang tidak direferensikan, lihat. the section called “Pertimbangan dan batasan”

Pemeliharaan untuk tabel

Tabel S3 menawarkan operasi pemeliharaan untuk meningkatkan manajemen dan kinerja tabel individual Anda. Opsi berikut diaktifkan secara default untuk semua tabel di bucket tabel. Anda dapat mengedit atau menonaktifkannya dengan menentukan file konfigurasi pemeliharaan untuk tabel S3 Anda.

Mengedit konfigurasi ini membutuhkan `s3tables:PutTableMaintenanceConfiguration` izin `s3tables:GetTableMaintenanceConfiguration` dan.

Topik

- Pemadatan
- Manajemen snapshot
- Pertimbangan dan keterbatasan

Pemadatan

Pemadatan dikonfigurasi pada tingkat tabel dan menggabungkan beberapa objek yang lebih kecil menjadi objek yang lebih sedikit dan lebih besar untuk meningkatkan kinerja Apache Iceberg kueri. Saat menggabungkan objek, pemadatan juga menerapkan efek penghapusan tingkat baris di tabel Anda.

Pemadatan diaktifkan secara default untuk semua tabel, dengan ukuran file target default 512MB, atau nilai kustom yang Anda tentukan antara 64MB hingga 512MB. File yang dipadatkan ditulis sebagai snapshot terbaru dari tabel Anda.

Strategi Pemadatan

Anda dapat memilih dari beberapa strategi pemadatan yang selanjutnya dapat meningkatkan kinerja kueri tergantung pada pola kueri dan urutan urutan tabel Anda.

Tabel S3 mendukung strategi pemadatan ini untuk tabel:

- Otomatis (Default)
  - Amazon S3 memilih strategi pemadatan terbaik berdasarkan urutan pengurutan tabel Anda. Ini adalah strategi pemadatan default untuk semua tabel.
  - Untuk tabel dengan urutan pengurutan yang ditentukan dalam metadata mereka, secara otomatis `auto` akan menerapkan `sort` pemadatan.

  - Untuk tabel tanpa urutan pengurutan, `auto` akan default menggunakan `binpack` pemadatan.
- Binpack
  - Menggabungkan file kecil menjadi file yang lebih besar, biasanya menargetkan ukuran lebih dari 100MB, sambil menerapkan penghapusan yang tertunda. Ini adalah strategi pemadatan default untuk tabel yang tidak disortir.
- Urutkan
  - Mengatur data berdasarkan kolom tertentu yang diurutkan secara otomatis berdasarkan hierarki selama pemadatan, meningkatkan kinerja kueri untuk operasi yang difilter. Strategi ini direkomendasikan ketika kueri Anda sering memfilter pada kolom tertentu. Saat Anda menggunakan strategi ini, Tabel S3 secara otomatis menerapkan pengurutan hierarkis pada kolom saat a `sort_order` didefinisikan dalam properti tabel.
- Urutan Z
  - Mengoptimalkan organisasi data dengan memadukan beberapa atribut ke dalam nilai skalar tunggal yang dapat digunakan untuk menyortir, memungkinkan kueri yang efisien di berbagai dimensi. Strategi ini direkomendasikan ketika Anda perlu melakukan kueri data di beberapa dimensi secara bersamaan. Strategi ini mengharuskan Anda untuk menentukan urutan pengurutan di properti tabel Iceberg Anda menggunakan properti `sort_order` tabel.

Pemadatan akan dikenakan biaya tambahan. Untuk informasi selengkapnya, lihat informasi harga di harga Amazon S3.

Contoh Pemadatan

Contoh berikut menunjukkan konfigurasi untuk pemadatan tabel.

Untuk mengkonfigurasi ukuran file target pemadatan dengan menggunakan AWS CLI

Ukuran file pemadatan target minimum adalah 64MB; maksimum adalah 512MB.

Contoh berikut akan mengubah ukuran file target menjadi 256MB menggunakan API. `PutTableMaintenanceConfiguration`

```
aws s3tables put-table-maintenance-configuration \
   --table-bucket-arn arn:aws:s3tables:us-east-1:111122223333:bucket/amzn-s3-demo-
bucket1 \
   --type icebergCompaction \
   --namespace mynamespace \
   --name testtable \
```

```
   --value='{"status":"enabled","settings":{"icebergCompaction":
{"targetFileSizeMB":256}}}'
```

Untuk informasi selengkapnya, lihat put-table-maintenance-configuration dalam AWS CLI Referensi Perintah.

Untuk mengkonfigurasi strategi pemadatan dengan menggunakan AWS CLI

Contoh berikut akan mengubah strategi pemadatan untuk `sort` menggunakan `PutTableMaintenanceConfiguration` API. Saat mengatur pemadatan, Anda dapat memilih dari strategi pemadatan berikut:`auto`,,`binpack`, `sort` atau `z-order`

> **Note**
>
> Untuk mengatur strategi pemadatan ke `sort` atau `z-order` Anda memerlukan prasyarat berikut:
>
> - Urutan pengurutan didefinisikan dalam properti Iceberg tabel Anda.
> - `s3tables:GetTableData`Izin.

```
aws s3tables put-table-maintenance-configuration \
   --table-bucket-arn arn:aws:s3tables:us-east-1:111122223333:bucket/amzn-s3-demo-
table-bucket \
   --type icebergCompaction \
   --namespace mynamespace \
   --name testtable \
   --value='{"status":"enabled","settings":{"icebergCompaction":
{"strategy":"sort"}}}'
```

Untuk informasi selengkapnya, lihat put-table-maintenance-configuration dalam AWS CLI Referensi Perintah.

Untuk menonaktifkan pemadatan dengan menggunakan AWS CLI

Contoh berikut akan menonaktifkan pemadatan menggunakan `PutTableMaintenanceConfiguration` API.

```
aws s3tables put-table-maintenance-configuration \
   --table-bucket-arn arn:aws:s3tables:us-east-1:111122223333:bucket/amzn-s3-demo-
table-bucket \
```

```
    --type icebergCompaction \
    --namespace mynamespace \
    --name testtable \
    --value='{"status":"disabled","settings":{"icebergCompaction":
{"targetFileSizeMB":256}}}'
```

Untuk informasi selengkapnya, lihat put-table-maintenance-configuration dalam AWS CLI Referensi Perintah.

Manajemen snapshot

Manajemen snapshot menentukan jumlah snapshot aktif untuk tabel Anda. Ini didasarkan pada `MinimumSnapshots` (1 secara default) dan `MaximumSnapshotAge` (120 jam secara default). Manajemen snapshot kedaluwarsa dan menghapus snapshot tabel berdasarkan konfigurasi ini.

Saat snapshot kedaluwarsa, Amazon S3 menandai objek apa pun yang hanya direferensikan oleh snapshot itu sebagai noncurrent. Objek noncurrent ini dihapus setelah jumlah hari yang ditentukan oleh `NoncurrentDays` properti dalam kebijakan penghapusan file yang tidak direferensikan.

> **Note**
>
> Penghapusan objek noncurrent bersifat permanen tanpa cara untuk memulihkan objek ini.

Untuk melihat atau memulihkan objek yang telah ditandai sebagai noncurrent, Anda harus menghubungi AWS Dukungan. Untuk informasi tentang menghubungi AWS Dukungan, lihat Kontak AWS atau AWS Dukungan Dokumentasi.

Manajemen snapshot menentukan objek yang akan dihapus dari tabel Anda dengan referensi hanya ke tabel itu. Referensi ke objek-objek ini dari luar tabel tidak akan mencegah manajemen snapshot menghapusnya.

> **Note**
>
> Manajemen snapshot tidak mendukung nilai retensi yang Anda konfigurasikan sebagai properti tabel Iceberg dalam `metadata.json` file atau melalui perintah `ALTER TABLE SET TBLPROPERTIES` SQL, termasuk retensi berbasis cabang atau tag. Manajemen snapshot dinonaktifkan saat Anda mengonfigurasi kebijakan penyimpanan berbasis cabang atau tag, atau mengonfigurasi kebijakan penyimpanan pada `metadata.json` file yang lebih panjang

dari nilai yang dikonfigurasi melalui API. `PutTableMaintenanceConfiguration` Dalam kasus ini S3 tidak akan kedaluwarsa atau menghapus snapshot dan Anda perlu menghapus snapshot secara manual atau menghapus properti dari tabel Iceberg Anda untuk menghindari biaya penyimpanan.

Anda hanya dapat mengonfigurasi manajemen snapshot di tingkat tabel. Untuk informasi selengkapnya, lihat informasi harga di harga Amazon S3.

Contoh Manajemen Snapshot

Contoh berikut menunjukkan konfigurasi untuk manajemen snapshot tabel.

Untuk mengkonfigurasi manajemen snapshot dengan menggunakan AWS CLI

Contoh berikut akan mengatur `MinimumSnapshots` ke 10 dan 2500 jam menggunakan `PutTableMaintenanceConfiguration` API. `MaximumSnapshotAge`

```
aws s3tables put-table-maintenance-configuration \
--table-bucket-arn arn:aws:s3tables:us-east-1:111122223333:bucket/amzn-s3-demo-
table-bucket \
--namespace my_namespace \
--name my_table \
--type icebergSnapshotManagement \
--value '{"status":"enabled","settings":{"icebergSnapshotManagement":
{"minSnapshotsToKeep":10,"maxSnapshotAgeHours":2500}}}'
```

Untuk menonaktifkan manajemen snapshot dengan menggunakan AWS CLI

Contoh berikut akan menonaktifkan manajemen snapshot menggunakan `PutTableMaintenanceConfiguration` API.

```
aws s3tables put-table-maintenance-configuration \
--table-bucket-arn arn:aws:s3tables:us-east-1:111122223333:bucket/amzn-s3-demo-
table-bucket \
--namespace my_namespace \
--name my_table \
--type icebergSnapshotManagement \
--value '{"status":"disabled","settings":{"icebergSnapshotManagement":
{"minSnapshotsToKeep":1,"maxSnapshotAgeHours":120}}}'
```

Untuk informasi selengkapnya, lihat put-table-maintenance-configuration dalam AWS CLI Referensi Perintah.

Pertimbangan dan keterbatasan

Untuk mempelajari lebih lanjut tentang pertimbangan dan batasan tambahan untuk manajemen pemadatan dan snapshot, lihat. the section called “Pertimbangan dan batasan”

> **Note**
>
> Tabel S3 menerapkan row-group-default ukuran parket 128 MB.

Pertimbangan dan batasan untuk pekerjaan pemeliharaan

Amazon S3 menawarkan operasi pemeliharaan untuk meningkatkan kinerja tabel S3 atau bucket meja Anda. Opsi ini adalah pemadatan file, manajemen snapshot, dan penghapusan file yang tidak direferensikan. Berikut ini adalah batasan dan pertimbangan untuk opsi manajemen ini.

Topik

- Pertimbangan untuk pemadatan
- Pertimbangan untuk manajemen snapshot
- Pertimbangan untuk penghapusan file yang tidak direferensikan
- Batas untuk pemeliharaan ember meja dan meja S3

Pertimbangan untuk pemadatan

Pertimbangan berikut berlaku untuk pemadatan. Untuk informasi lebih lanjut tentang pemadatan, lihatthe section called “Pemeliharaan meja”.

- Pemadatan hanya didukung pada jenis Apache Parquet file.
- Pemadatan tidak mendukung tipe data: Tetap.
- Pemadatan tidak mendukung jenis kompresi:`brotli`,`lz4`.

Pertimbangan untuk manajemen snapshot

Pertimbangan berikut berlaku untuk manajemen snapshot. Untuk informasi selengkapnya tentang manajemen snapshot, lihatthe section called “Pemeliharaan meja”.

- Snapshot akan dipertahankan hanya jika kedua kriteria terpenuhi: jumlah minimum snapshot yang harus disimpan dan periode retensi yang ditentukan.
- Manajemen snapshot menghapus metadata snapshot kedaluwarsa dari Apache Iceberg, mencegah kueri perjalanan waktu untuk snapshot kedaluwarsa dan secara opsional menghapus file data terkait.
- Manajemen snapshot tidak mendukung nilai retensi yang Anda konfigurasikan sebagai properti tabel Iceberg dalam `metadata.json` file atau melalui perintah `ALTER TABLE SET TBLPROPERTIES` SQL, termasuk retensi berbasis cabang atau tag. Manajemen snapshot dinonaktifkan saat Anda mengonfigurasi kebijakan penyimpanan berbasis cabang atau tag, atau mengonfigurasi kebijakan penyimpanan pada `metadata.json` file yang lebih panjang dari nilai yang dikonfigurasi melalui API. `PutTableMaintenanceConfiguration` Dalam kasus ini S3 tidak akan kedaluwarsa atau menghapus snapshot dan Anda perlu menghapus snapshot secara manual atau menghapus properti dari tabel Iceberg Anda untuk menghindari biaya penyimpanan.

Pertimbangan untuk penghapusan file yang tidak direferensikan

Pertimbangan berikut berlaku untuk penghapusan file yang tidak direferensikan. Untuk informasi selengkapnya tentang penghapusan file yang tidak direferensikan, lihat. the section called “Perawatan ember meja”

- Penghapusan file yang tidak direferensikan menghapus data dan file metadata yang tidak lagi direferensikan oleh metadata Iceberg jika waktu pembuatannya sebelum periode retensi.

Batas untuk pemeliharaan ember meja dan meja S3

Operasi pemeliharaan	Properti	Dapat dikonfigurasi pada tingkat ember meja?	Dapat dikonfigurasi di tingkat tabel?	Nilai default	Nilai minimum
Pemadatan	targetFileSizeMB	Tidak	Ya	512MB	64MB
Manajemen snapshot	MinimumSnapshots	Tidak	Ya	1	1
Manajemen snapshot	maximumSnapshotAge	Tidak	Ya	120 jam	1 jam
Penghapusan file yang tidak direferensikan	Hari yang tidak direferensikan	Ya	Tidak	3 hari	1 hari
Penghapusan file yang tidak direferensikan	nonCurrentDays	Ya	Tidak	10 hari	1 hari

> **Note**
>
> Tabel S3 menerapkan row-group-default ukuran parket 128 MB.

Ruang nama tabel

Saat membuat tabel dalam keranjang tabel Amazon S3, Anda mengaturnya ke dalam pengelompokan logis yang disebut ruang nama. Tidak seperti tabel S3 dan bucket tabel, ruang nama bukanlah sumber daya. Namespaces adalah konstruksi yang membantu Anda mengatur dan mengelola tabel Anda dengan cara yang terukur. Misalnya, semua tabel milik departemen sumber daya manusia di perusahaan dapat dikelompokkan di bawah nilai namespace umum. `hr`

Untuk mengontrol akses ke ruang nama tertentu, Anda dapat menggunakan kebijakan sumber daya bucket tabel. Untuk informasi selengkapnya, lihat the section called “Kebijakan berbasis sumber daya”.

Aturan berikut berlaku untuk ruang nama tabel:

- Setiap namespace harus unik di dalam ember meja.
- Anda dapat membuat hingga 10.000 ruang nama per keranjang meja.
- Setiap nama tabel harus unik dalam namespace.
- Setiap tabel hanya dapat memiliki satu tingkat ruang nama. Ruang nama tidak dapat disarangkan.
- Setiap tabel milik satu namespace.
- Anda dapat memindahkan tabel Anda di antara ruang nama.

Topik

- Membuat namespace
- Hapus namespace

Membuat namespace

Namespace tabel adalah konstruksi logis yang Anda kelompokkan tabel di bawah keranjang tabel Amazon S3. Setiap tabel milik satu namespace. Sebelum membuat tabel di keranjang tabel, Anda harus membuat namespace untuk mengelompokkan tabel di bawahnya. Anda dapat membuat namespace menggunakan konsol Amazon S3 AWS Command Line Interface ,AWS CLI(), Amazon S3 REST API AWS SDKs, atau mesin kueri terintegrasi.

Nama namespace

Aturan penamaan berikut berlaku untuk ruang nama:

- Panjang nama harus antara 1 dan 255 karakter.
- Nama hanya dapat terdiri dari huruf kecil, angka, dan garis bawah (). _ Underscore tidak diizinkan di awal atau akhir nama namespace.
- Nama harus dimulai dan diakhiri dengan huruf atau angka.
- Nama tidak boleh mengandung tanda hubung (-) atau titik (.).
- Namespace harus unik di dalam ember meja.

- Nama namespace tidak boleh dimulai dengan awalan cadangan. `aws` Misalnya, Anda tidak dapat menggunakan `aws_s3_metadata` sebagai namespace. `aws_s3_metadata`adalah namespace yang dicadangkan untuk tabel metadata. Untuk informasi selengkapnya, lihat Mempercepat penemuan data dengan Metadata S3.

Untuk informasi selengkapnya tentang nama namespace yang valid, lihat. Aturan penamaan untuk tabel dan ruang nama

Menggunakan konsol S3 dan Amazon Athena

Prosedur berikut menggunakan alur kerja Buat tabel dengan Athena untuk membuat namespace di konsol Amazon S3. Jika Anda tidak ingin juga menggunakan Amazon Athena untuk membuat tabel di namespace Anda, Anda dapat membatalkan alur kerja setelah membuat namespace Anda.

Untuk membuat namespace

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tabel.
3. Pada halaman bucket Table, pilih bucket tempat Anda ingin membuat namespace.
4. Pada halaman detail bucket, pilih Buat tabel dengan Athena.
5. Dalam kotak dialog Buat tabel dengan Athena, pilih Buat namespace, lalu pilih Buat namespace.
6. Masukkan nama di bidang nama Namespace. Nama namespace harus 1 hingga 255 karakter dan unik di dalam keranjang tabel. Karakter yang valid adalah a—z, 0-9, dan garis bawah (). _ Underscore tidak diizinkan di awal atau akhir nama namespace.
7. Pilih Buat namespace.
8. Jika Anda juga ingin membuat tabel, pilih Buat tabel dengan Athena. Untuk informasi lebih lanjut tentang membuat tabel dengan Athena, lihat. the section called “Menggunakan konsol S3 dan Amazon Athena” Jika Anda tidak ingin membuat tabel sekarang, pilih Batal.

Menggunakan AWS CLI

Contoh ini menunjukkan cara membuat namespace tabel dengan menggunakan. AWS CLI Untuk menggunakan contoh ini, ganti *`user input placeholders`*dengan informasi Anda sendiri.

```
aws s3tables create-namespace \
```

```
    --table-bucket-arn arn:aws:s3tables:us-east-1:111122223333:bucket/amzn-s3-demo-
bucket1 \
    --namespace example_namespace
```

Menggunakan mesin kueri

Anda dapat membuat namespace dalam Apache Spark sesi yang terhubung ke bucket tabel Amazon S3 Anda.

Contoh ini menunjukkan cara membuat tabel dengan menggunakan `CREATE` pernyataan di mesin kueri yang terintegrasi dengan Tabel S3. Untuk menggunakan contoh ini, ganti *`user input placeholders`*dengan informasi Anda sendiri.

```
spark.sql("CREATE NAMESPACE IF NOT EXISTS s3tablesbucket.my_namespace")
```

Hapus namespace

Sebelum menghapus namespace tabel dari bucket tabel Amazon S3, Anda harus menghapus semua tabel di dalam namespace, atau memindahkannya di bawah namespace lain. Anda dapat menghapus namespace dengan menggunakan Amazon S3 REST API AWS SDKs,,AWS CLI() AWS Command Line Interface , atau mesin kueri terintegrasi.

Untuk informasi tentang izin yang diperlukan untuk menghapus namespace, lihat DeleteNamespacedi Referensi API Amazon Simple Storage Service.

Menggunakan AWS CLI

Contoh ini menunjukkan cara menghapus namespace tabel dengan menggunakan. AWS CLI Untuk menggunakan contoh ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3tables delete-namespace \
    --table-bucket-arn arn:aws:s3tables:us-east-1:111122223333:bucket/amzn-s3-demo-
bucket1 \
    --namespace example_namespace
```

Tabel dalam ember meja S3

Tabel S3 mewakili kumpulan data terstruktur yang terdiri dari data tabel yang mendasari dan metadata terkait. Data ini disimpan di dalam bucket tabel sebagai subresource. Semua tabel dalam

ember meja disimpan dalam format tabel Apache Iceberg. Amazon S3 mengelola pemeliharaan tabel Anda melalui pemadatan file otomatis dan manajemen snapshot. Untuk informasi selengkapnya, lihat Pemeliharaan untuk tabel.

Untuk membuat tabel di akun Anda dapat diakses oleh layanan AWS analitik, Anda mengintegrasikan bucket tabel Amazon S3 Anda dengan Amazon Lakehouse. SageMaker Integrasi ini memungkinkan layanan AWS analitik seperti Amazon Athena dan Amazon Redshift untuk secara otomatis menemukan dan mengakses data tabel Anda.

Saat Anda membuat tabel, Amazon S3 secara otomatis menghasilkan lokasi gudang untuk tabel. Ini adalah lokasi S3 unik yang menyimpan objek yang terkait dengan tabel. Contoh berikut menunjukkan format lokasi gudang:

```
s3://63a8e430-6e0b-46f5-k833abtwr6s8tmtsycedn8s4yc3xhuse1b--table-s3
```

Dalam bucket tabel Anda, Anda dapat mengatur tabel ke dalam pengelompokan logis yang disebut namespaces. Untuk informasi selengkapnya, lihat Ruang nama tabel.

Anda dapat mengganti nama tabel, tetapi setiap tabel memiliki Nama Sumber Daya Amazon (ARN) yang unik dan ID tabel unik. Setiap tabel juga memiliki kebijakan sumber daya yang melekat padanya. Anda dapat menggunakan kebijakan ini untuk mengelola akses ke tabel.

Tabel ARNs menggunakan format berikut:

```
arn:aws:s3tables:region:owner-account-id:bucket/bucket-name/table/table-id
```

Secara default, Anda dapat membuat hingga 10.000 tabel dalam ember meja. Untuk meminta peningkatan kuota untuk bucket atau tabel tabel, hubungi. Dukungan

Amazon S3 mendukung jenis tabel berikut di bucket tabel:

Tabel pelanggan

Tabel pelanggan adalah tabel yang dapat Anda baca dan tulis. Anda dapat mengambil data dari tabel ini menggunakan mesin kueri terintegrasi. Anda dapat menyisipkan, memperbarui, atau menghapus data di dalamnya dengan menggunakan operasi API S3 atau mesin kueri terintegrasi.

AWS tabel

AWS tabel adalah tabel hanya-baca yang dihasilkan oleh Layanan AWS atas nama Anda. Tabel ini dikelola oleh Amazon S3 dan tidak dapat dimodifikasi oleh prinsipal IAM di luar Amazon S3 itu

sendiri. Anda dapat mengambil informasi dari tabel ini, tetapi Anda tidak dapat mengubah data di dalamnya. AWS tabel mencakup tabel Metadata S3, yang berisi metadata yang diambil dari objek dalam bucket tujuan umum S3. Lihat informasi yang lebih lengkap di Mempercepat penemuan data dengan Metadata S3.

Topik

- Membuat tabel Amazon S3
- Menghapus tabel Amazon S3
- Mengelola kebijakan tabel

Membuat tabel Amazon S3

Tabel Amazon S3 adalah subsumber daya dari ember meja. Tabel disimpan di Apache Iceberg format sehingga Anda dapat bekerja dengan mereka dengan menggunakan mesin kueri dan aplikasi lain yang mendukung Apache Iceberg. Amazon S3 terus mengoptimalkan tabel Anda untuk membantu mengurangi biaya penyimpanan dan meningkatkan kinerja kueri analitik.

Saat Anda membuat tabel, Amazon S3 secara otomatis menghasilkan lokasi gudang untuk tabel. Lokasi gudang adalah lokasi S3 unik di mana Anda dapat membaca dan menulis objek yang terkait dengan tabel. Contoh berikut menunjukkan format lokasi gudang:

```
s3://63a8e430-6e0b-46f5-k833abtwr6s8tmtsycedn8s4yc3xhuse1b--table-s3
```

Tabel memiliki format Amazon Resource Name (ARN) berikut:

```
arn:aws:s3tables:region:owner-account-id:bucket/bucket-name/table/table-id
```

Secara default, Anda dapat membuat hingga 10.000 tabel dalam ember meja. Untuk meminta peningkatan kuota untuk bucket atau tabel tabel, hubungi. Dukungan

Anda dapat membuat tabel menggunakan konsol Amazon S3, Amazon S3 REST API,,AWS CLI() AWS SDKs AWS Command Line Interface , atau mesin kueri yang terhubung ke bucket tabel Anda.

Saat Anda membuat tabel, Anda dapat menentukan pengaturan enkripsi untuk tabel tersebut, kecuali jika Anda membuat tabel dengan Athena. Jika Anda tidak menentukan pengaturan enkripsi, tabel dienkripsi dengan pengaturan default untuk keranjang tabel. Untuk informasi selengkapnya, lihat Menentukan enkripsi untuk tabel.

Prasyarat untuk membuat tabel

Untuk membuat tabel, Anda harus terlebih dahulu melakukan hal berikut:

- Buat ember meja.
- Buat namespacedi ember meja Anda.
- Pastikan Anda memiliki izin AWS Identity and Access Management (IAM) untuk `s3tables:CreateTable` dan. `s3tables:PutTableData`
-

> **Note**
>
> Jika Anda menggunakan enkripsi SSE-KMS untuk tabel Anda, Anda memerlukan izin untuks`3tables:PutTableEncryption`, dan `DescribeKey` izin pada kunci yang dipilih. AWS KMS Selain itu AWS KMS kunci yang Anda gunakan perlu memberikan izin Tabel S3 untuk melakukan pemeliharaan tabel otomatis. Untuk informasi selengkapnya, lihat Persyaratan izin untuk enkripsi SSE-KMS Tabel S3

Untuk informasi tentang nama tabel yang valid, lihatAturan penamaan untuk tabel dan ruang nama.

> **Important**
>
> Saat membuat tabel, pastikan Anda menggunakan semua huruf kecil dalam nama tabel dan definisi tabel Anda. Misalnya, pastikan bahwa nama kolom Anda semua huruf kecil. Jika nama tabel atau definisi tabel berisi huruf kapital, tabel tidak didukung oleh AWS Lake Formation atau AWS Glue Data Catalog. Dalam hal ini, tabel Anda tidak akan terlihat oleh layanan AWS analitik seperti Amazon Athena, meskipun bucket tabel Anda terintegrasi dengan AWS layanan analitik.
> Jika definisi tabel berisi huruf kapital, Anda menerima pesan galat berikut saat menjalankan `SELECT` kueri di Athena: “GENERIC_INTERNAL_ERROR: Get table request failed: com.amazonaws.services.glue.model. ValidationException: Sumber Daya Federasi Tidak Didukung - Nama tabel atau kolom tidak valid.

Menggunakan konsol S3 dan Amazon Athena

Prosedur berikut menggunakan konsol Amazon S3 untuk membuat tabel dengan Amazon Athena. Jika Anda belum membuat namespace di bucket tabel Anda, Anda dapat melakukannya

sebagai bagian dari proses ini. Sebelum melakukan langkah-langkah berikut, pastikan Anda telah mengintegrasikan bucket tabel Anda dengan layanan AWS analitik di Wilayah ini. Untuk informasi selengkapnya, lihat the section called “Menggunakan Tabel S3 dengan layanan AWS analitik”.

> **Note**
>
> Saat Anda membuat tabel menggunakan Athena, tabel tersebut mewarisi pengaturan enkripsi default dari keranjang tabel. Jika Anda ingin menggunakan jenis enkripsi yang berbeda, Anda perlu membuat tabel menggunakan metode lain.

Untuk membuat tabel

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tabel.
3. Pada halaman bucket Table, pilih bucket tempat Anda ingin membuat tabel.
4. Pada halaman detail bucket, pilih Buat tabel dengan Athena.
5. Dalam Buat tabel dengan Athena kotak dialog, lakukan salah satu hal berikut:
   - Buat namespace baru. Pilih Buat namespace, lalu masukkan nama di bidang Nama Namespace. Nama namespace harus 1 hingga 255 karakter dan unik di dalam keranjang tabel. Karakter yang valid adalah a—z, 0-9, dan garis bawah (). _ Garis bawah tidak diizinkan pada awal nama namespace.
   - Pilih Buat namespace.
   - Tentukan namespace yang ada. Pilih Tentukan namespace yang ada dalam keranjang tabel ini. Kemudian pilih Pilih dari ruang nama yang ada atau Masukkan nama namespace yang ada. Jika Anda memiliki lebih dari 1.000 ruang nama di bucket, Anda harus memasukkan nama namespace jika tidak muncul dalam daftar.
6. Pilih Buat tabel dengan Athena.
7. Konsol Amazon Athena terbuka dan editor kueri Athena muncul. Bidang Catalog harus diisi dengan s3tablescatalog/ diikuti dengan nama bucket tabel Anda, misalnya, s3tablescatalog/***amzn-s3-demo-bucket*** Bidang Database harus diisi dengan namespace yang Anda buat atau pilih sebelumnya.

> **Note**
>
> Jika Anda tidak melihat nilai ini di bidang Katalog dan Database, pastikan Anda telah mengintegrasikan bucket tabel Anda dengan layanan AWS analitik di Wilayah ini. Untuk informasi selengkapnya, lihat the section called “Menggunakan Tabel S3 dengan layanan AWS analitik”.

8. Editor kueri diisi dengan kueri sampel yang dapat Anda gunakan untuk membuat tabel. Ubah kueri untuk menentukan nama tabel dan kolom yang Anda ingin tabel Anda miliki.
9. Setelah selesai memodifikasi kueri, pilih Jalankan untuk membuat tabel Anda.

> **Note**
>
> - Jika Anda menerima kesalahan “Izin tidak cukup untuk menjalankan kueri. Principal tidak memiliki hak istimewa pada sumber daya tertentu “ketika Anda mencoba menjalankan kueri di Athena, Anda harus diberikan izin Formasi Danau yang diperlukan di atas meja. Untuk informasi selengkapnya, lihat the section called “Memberikan izin pada tabel atau database”.
> - Jika Anda menerima kesalahan “Gunung es tidak dapat mengakses sumber daya yang diminta” ketika Anda mencoba menjalankan kueri di Athena, buka AWS Lake Formation konsol dan pastikan bahwa Anda telah memberikan izin pada katalog keranjang tabel dan database (namespace) yang Anda buat. Jangan tentukan tabel saat memberikan izin ini. Untuk informasi selengkapnya, lihat the section called “Memberikan izin pada tabel atau database”.
> - Jika Anda menerima pesan galat berikut saat menjalankan `SELECT` kueri di Athena, pesan ini disebabkan oleh adanya huruf kapital di nama tabel atau nama kolom Anda dalam definisi tabel Anda: “GENERIC_INTERNAL_ERROR: Get table request failed: com.amazonaws.services.glue.model. ValidationException: Sumber Daya Federasi Tidak Didukung - Nama tabel atau kolom tidak valid. Pastikan bahwa nama tabel dan kolom Anda semuanya huruf kecil.

Jika pembuatan tabel Anda berhasil, nama tabel baru Anda muncul di daftar tabel di Athena. Saat Anda menavigasi kembali ke konsol Amazon S3, tabel baru akan muncul di daftar Tabel di halaman detail bucket untuk bucket tabel setelah Anda menyegarkan daftar.

Menggunakan AWS CLI

Contoh ini menunjukkan cara membuat tabel dengan skema dengan menggunakan AWS CLI dan menentukan metadata tabel dengan JSON. Untuk menggunakan contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3tables create-table --cli-input-json file://mytabledefinition.json
```

Untuk `mytabledefinition.json` file, gunakan definisi tabel contoh berikut. Untuk menggunakan contoh ini, ganti *user input placeholders*dengan informasi Anda sendiri.

```
{
    "tableBucketARN": "arn:aws:s3tables:us-east-1:111122223333:bucket/amzn-s3-demo-table-bucket",
    "namespace": "your_namespace",
    "name": "example_table",
    "format": "ICEBERG",
    "metadata": {
        "iceberg": {
            "schema": {
                "fields": [
                     {"name": "id", "type": "int","required": true},
                     {"name": "name", "type": "string"},
                     {"name": "value", "type": "int"}
                ]
            }
        }
    }
}
```

Menggunakan mesin kueri

Anda dapat membuat tabel di mesin kueri yang didukung yang terhubung ke bucket tabel Anda, seperti di Apache Spark sesi di Amazon EMR.

Contoh berikut menunjukkan cara membuat tabel dengan Spark dengan menggunakan `CREATE` pernyataan, dan menambahkan data tabel dengan menggunakan `INSERT` pernyataan atau dengan membaca data dari file yang ada. Untuk menggunakan contoh ini, ganti *user input placeholders*dengan informasi Anda sendiri.

```
spark.sql(
```

```
" CREATE TABLE IF NOT EXISTS s3tablesbucket.example_namespace.`example_table` (
    id INT,
    name STRING,
    value INT
)
USING iceberg "
)
```

Setelah Anda membuat tabel, Anda dapat memuat data ke dalam tabel. Pilih dari metode berikut:

- Tambahkan data ke dalam tabel dengan menggunakan `INSERT` pernyataan.

```
spark.sql(
"""
    INSERT INTO s3tablesbucket.my_namespace.my_table
    VALUES
        (1, 'ABC', 100),
        (2, 'XYZ', 200)
""")
```

- Memuat file data yang ada.

  1. Baca data ke Spark:

```
val data_file_location = "Path such as S3 URI to data file"
val data_file = spark.read.parquet(data_file_location)
```

  2. Tulis data ke dalam tabel Gunung Es:

```
data_file.writeTo("s3tablesbucket.my_namespace.my_table").using("Iceberg").tableProperty
 ("format-version", "2").createOrReplace()
```

Menghapus tabel Amazon S3

Anda dapat menghapus tabel dengan menggunakan Amazon S3 REST API, AWS SDK, AWS CLI atau menggunakan mesin kueri terintegrasi.

> **Note**
>
> Tabel S3 tidak mendukung `DROP TABLE` operasi dengan`purge=false`. Beberapa versi Spark selalu menyetel flag ini ke false bahkan saat menjalankan `DROP TABLE PURGE`

perintah. Anda dapat mencoba lagi `DROP TABLE` dengan `purge=true` atau menggunakan S3 Tables DeleteTableREST API untuk menghapus tabel.
Saat Anda menghapus tabel, objek yang terkait dengan tabel itu menjadi tidak terkini dan dapat memakan waktu hingga satu hari untuk dihapus.

Menggunakan AWS CLI

Contoh ini menunjukkan cara menghapus tabel dengan menggunakan AWS CLI. Untuk menggunakan perintah ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3tables delete-table \
    --table-bucket-arn arn:aws:s3tables:us-east-1:111122223333:bucket/amzn-s3-demo-
table-bucket \
    --namespace example_namespace --name example_table
```

Menggunakan mesin kueri

Anda dapat menghapus tabel dalam Apache Spark sesi yang terhubung ke bucket tabel Amazon S3 Anda.

Contoh ini menunjukkan cara menghapus tabel dengan menggunakan `DROP TABLE PURGE` perintah. Untuk menggunakan perintah ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
spark.sql(
" DROP TABLE [IF EXISTS] s3tablesbucket.example_namespace.example_table PURGE")
```

Mengelola kebijakan tabel

Anda dapat menambahkan, menghapus, memperbarui, dan melihat kebijakan tabel untuk tabel menggunakan Amazon S3 REST API, AWS SDK, dan file. AWS CLI Untuk informasi selengkapnya, lihat topik berikut. Untuk informasi selengkapnya tentang tindakan dan kunci kondisi yang didukung AWS Identity and Access Management (IAM) untuk Tabel Amazon S3, lihat. Manajemen akses untuk Tabel S3 Misalnya kebijakan tabel, lihatKebijakan berbasis sumber daya untuk Tabel S3.

Menambahkan kebijakan tabel

Untuk menambahkan kebijakan tabel ke tabel, Anda dapat menggunakan Amazon S3 REST API, AWS SDK, dan file. AWS CLI

Menggunakan AWS CLI

Contoh ini menunjukkan cara membuat kebijakan tabel dengan menggunakan AWS CLI. Untuk menggunakan perintah ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3tables put-table-policy \
    --table-bucket-arn arn:aws:s3tables:us-east-1:111122223333:bucket/amzn-s3-demo-bucket1 \
    --namespace my-namespace \
    --name my-table \
    --resource-policy your-policy-JSON
```

Melihat kebijakan tabel

Untuk melihat kebijakan bucket yang dilampirkan ke tabel, Anda dapat menggunakan Amazon S3 REST API, AWS SDK, dan file. AWS CLI

Menggunakan AWS CLI

Contoh ini menunjukkan cara melihat kebijakan yang dilampirkan ke tabel dengan menggunakan AWS CLI. Untuk menggunakan perintah ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3tables get-table-policy \
    --table-bucket-arn arn:aws:s3tables:us-east-1:111122223333:bucket/amzn-s3-demo-bucket1/table/tableID \
    --namespace my-namespace \
    --name my-table
```

Menghapus kebijakan tabel

Untuk menghapus kebijakan yang dilampirkan ke tabel, Anda dapat menggunakan Amazon S3 REST API, AWS SDK, dan file. AWS CLI

Menggunakan AWS CLI

Contoh ini menunjukkan cara menghapus kebijakan tabel dengan menggunakan AWS CLI. Untuk menggunakan perintah ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3tables delete-table-policy \
```

```
    --table-ARN arn:aws:s3tables:us-east-1:111122223333:bucket/amzn-s3-demo-bucket1/
table/tableID \
    --namespace your-namespace \
    --name your-table
```

Mengakses data tabel

Ada beberapa cara untuk mengakses tabel di bucket tabel Amazon S3, Anda dapat mengintegrasikan tabel dengan layanan AWS analitik menggunakan Amazon SageMaker Lakehouse, atau mengakses tabel secara langsung menggunakan titik akhir Tabel Amazon S3 atau Katalog Tabel Amazon S3 Iceberg REST untuk. Apache Iceberg Metode akses yang Anda gunakan akan bergantung pada pengaturan katalog, model tata kelola, dan kebutuhan kontrol akses Anda. Berikut ini adalah ikhtisar metode akses ini.

Integrasi Amazon SageMaker Lakehouse

Ini adalah metode akses yang disarankan untuk bekerja dengan tabel di ember tabel S3. Integrasi ini memberi Anda manajemen tabel terpadu, tata kelola terpusat, dan kontrol akses berbutir halus di berbagai layanan analitik. AWS

Akses langsung

Gunakan metode ini jika Anda perlu bekerja dengan implementasi katalog AWS Partner Network (APN), implementasi katalog kustom, atau jika Anda hanya perlu melakukan operasi baca/tulis dasar pada tabel dalam satu keranjang tabel.

> **Note**
>
> Untuk mengakses tabel, identitas IAM yang Anda gunakan memerlukan akses ke sumber daya tabel dan tindakan Tabel S3. Untuk informasi selengkapnya, lihat Manajemen akses untuk Tabel S3.

Mengakses tabel melalui integrasi Amazon SageMaker Lakehouse

Anda dapat mengintegrasikan bucket tabel S3 dengan Amazon SageMaker Lakehouse untuk mengakses tabel dari layanan AWS analitik, seperti Amazon Athena, Amazon Redshift, dan. QuickSight Amazon SageMaker Lakehouse menyatukan data Anda di seluruh danau data Amazon S3 dan gudang data Amazon Redshift, sehingga Anda dapat membangun analitik, pembelajaran

mesin, dan aplikasi AI generatif pada satu salinan data. Integrasi mengisi AWS Glue Data Catalog dengan sumber daya tabel Anda, dan menggabungkan akses ke sumber daya ini dengan. AWS Lake Formation Untuk informasi lebih lanjut tentang integrasi, lihatMenggunakan Tabel Amazon S3 dengan layanan analitik AWS.

Integrasi ini memungkinkan kontrol akses berbutir halus AWS Lake Formation untuk memberikan keamanan tambahan. Lake Formation menggunakan kombinasi model izinnya sendiri dan model izin IAM untuk mengontrol akses ke sumber daya tabel dan data yang mendasarinya. Ini berarti bahwa permintaan untuk mengakses tabel Anda harus melewati pemeriksaan izin oleh IAM dan Lake Formation. Untuk informasi selengkapnya, lihat ikhtisar izin Lake Formation di Panduan AWS Lake Formation Pengembang.

Layanan AWS analitik berikut dapat mengakses tabel melalui integrasi ini:

- Amazon Athena
- Amazon Redshift
- Amazon EMR
- QuickSight
- Amazon Data Firehose

Mengakses tabel menggunakan titik akhir AWS GlueIceberg REST

Setelah bucket tabel S3 Anda terintegrasi dengan Amazon SageMaker Lakehouse, Anda juga dapat menggunakan AWS GlueIceberg REST titik akhir untuk terhubung ke tabel S3 dari mesin kueri pihak ketiga yang mendukung. Iceberg Untuk informasi selengkapnya, lihat Mengakses tabel Amazon S3 menggunakan AWS Glue Iceberg REST titik akhir.

Sebaiknya gunakan AWS GlueIceberg REST titik akhir saat Anda ingin mengakses tabel dariSpark,PyIceberg, atau klien lain Iceberg yang kompatibel.

Klien berikut dapat mengakses tabel secara langsung melalui AWS GlueIceberg REST titik akhir:

- Setiap Iceberg klien, termasukSpark,PyIceberg, dan banyak lagi.

Mengakses tabel secara langsung

Anda dapat mengakses tabel langsung dari mesin kueri sumber terbuka melalui metode yang menjembatani operasi manajemen Tabel S3 ke aplikasi Apache Iceberg analitik Anda. Ada dua

metode akses langsung: Iceberg REST titik akhir Tabel Amazon S3 atau Katalog Tabel Amazon S3 untuk. Apache Iceberg RESTTitik akhir direkomendasikan.

Kami merekomendasikan akses langsung jika Anda mengakses tabel dalam implementasi katalog yang dikelola sendiri, atau hanya perlu melakukan operasi baca/tulis dasar pada tabel dalam satu keranjang tabel. Untuk skenario akses lainnya, kami merekomendasikan integrasi Amazon SageMaker Lakehouse.

Akses langsung ke tabel dikelola melalui kebijakan berbasis identitas IAM atau kebijakan berbasis sumber daya yang dilampirkan pada tabel dan bucket tabel. Anda tidak perlu mengelola izin Lake Formation untuk tabel saat Anda mengaksesnya secara langsung.

Mengakses tabel melalui titik akhir Tabel Amazon S3 Iceberg REST

Anda dapat menggunakan Iceberg REST endpoint Amazon S3 Tables untuk mengakses tabel Anda langsung dari klien yang Iceberg REST kompatibel melalui HTTP titik akhir, untuk informasi selengkapnya, lihat. Mengakses tabel menggunakan Tabel Amazon S3 Iceberg REST titik akhir

Layanan AWS analitik dan mesin kueri berikut dapat mengakses tabel secara langsung menggunakan titik akhir Tabel Iceberg REST Amazon S3:

Mesin kueri yang didukung

- Setiap Iceberg klien, termasukSpark,PyIceberg, dan banyak lagi.
- Amazon EMR
- AWS Glue ETL

Mengakses tabel langsung melalui Katalog Tabel Amazon S3 untuk Apache Iceberg

Anda juga dapat mengakses tabel langsung dari mesin kueri seperti Apache Spark dengan menggunakan katalog klien Tabel S3, untuk informasi selengkapnya, lihatMengakses tabel Amazon S3 dengan Katalog Tabel Amazon S3 untuk Apache Iceberg. Namun, S3 merekomendasikan penggunaan endpoint Amazon S3 Iceberg REST Tables untuk akses langsung karena mendukung lebih banyak aplikasi, tanpa memerlukan bahasa atau kode khusus mesin.

Mesin query berikut dapat mengakses tabel secara langsung menggunakan katalog klien:

- Apache Spark

Menggunakan Tabel Amazon S3 dengan layanan analitik AWS

Untuk membuat tabel di akun Anda dapat diakses oleh layanan AWS analitik, Anda mengintegrasikan bucket tabel Amazon S3 Anda dengan Amazon Lakehouse. SageMaker Integrasi ini memungkinkan layanan AWS analitik untuk secara otomatis menemukan dan mengakses data tabel Anda. Anda dapat menggunakan integrasi ini untuk bekerja dengan tabel Anda di layanan ini:

- Amazon Athena
- Amazon Redshift
- Amazon EMR
- QuickSight
- Amazon Data Firehose

> **Note**
>
> Integrasi ini menggunakan AWS Glue dan AWS Lake Formation layanan dan mungkin menimbulkan AWS Glue permintaan dan biaya penyimpanan. Untuk informasi selengkapnya, silakan lihat Harga AWS Glue.
> Harga tambahan berlaku untuk menjalankan kueri pada tabel S3 Anda. Untuk informasi selengkapnya, lihat informasi harga untuk mesin kueri yang Anda gunakan.

Bagaimana integrasi bekerja

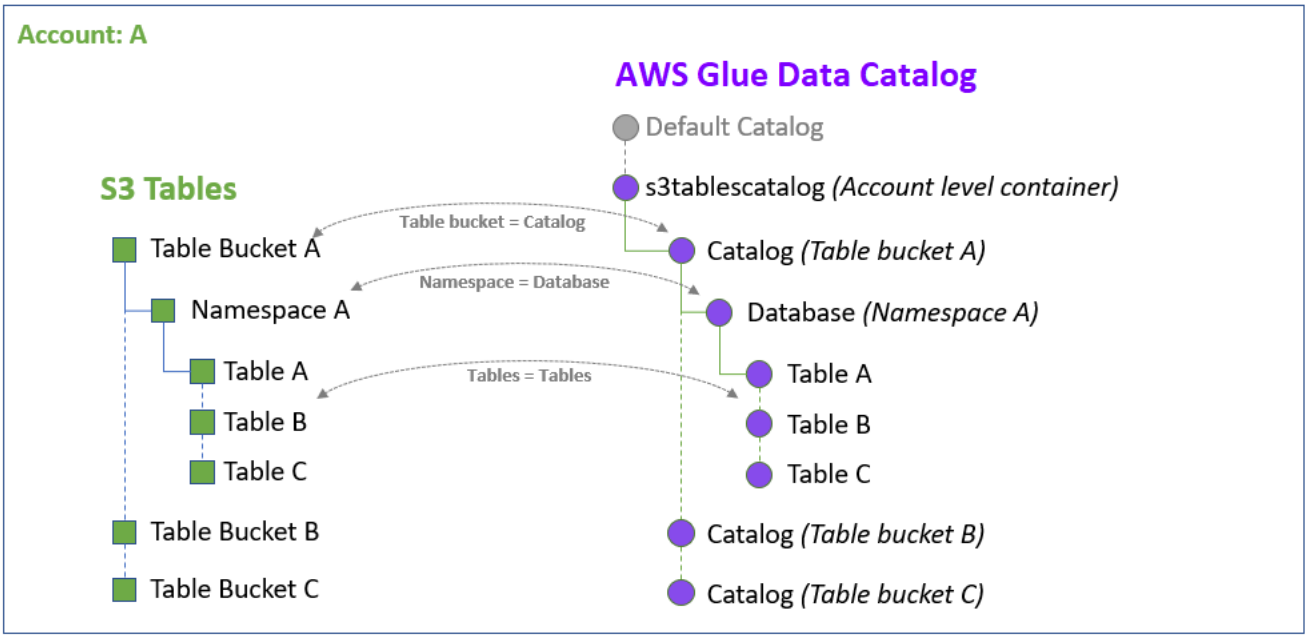
Saat membuat bucket tabel di konsol, Amazon S3 memulai tindakan berikut untuk mengintegrasikan bucket tabel di Wilayah yang telah dipilih dengan layanan analitik: AWS

1. Membuat peran layanan baru AWS Identity and Access Management (IAM) yang memberikan akses Lake Formation ke semua bucket tabel Anda.
2. Menggunakan peran layanan, Lake Formation mendaftarkan bucket tabel di Wilayah saat ini. Hal ini memungkinkan Lake Formation untuk mengelola akses, izin, dan tata kelola untuk semua bucket tabel saat ini dan masa depan di Wilayah tersebut.
3. Menambahkan `s3tablescatalog` katalog ke AWS Glue Data Catalog dalam Wilayah saat ini. Menambahkan `s3tablescatalog` katalog memungkinkan semua bucket tabel, ruang nama, dan tabel diisi dalam Katalog Data.

> **Note**
>
> Tindakan ini otomatis melalui konsol Amazon S3. Jika Anda melakukan integrasi ini secara terprogram, Anda harus secara manual mengambil semua tindakan ini.

Anda mengintegrasikan ember meja Anda sekali per AWS Wilayah. Setelah integrasi selesai, semua bucket tabel saat ini dan future, namespace, dan tabel ditambahkan ke AWS Glue Data Catalog dalam Region tersebut.

Ilustrasi berikut menunjukkan bagaimana `s3tablescatalog` katalog secara otomatis mengisi bucket tabel, ruang nama, dan tabel di Wilayah saat ini sebagai objek yang sesuai dalam Katalog Data. Bucket meja diisi sebagai subkatalog. Ruang nama dalam keranjang tabel diisi sebagai database dalam subkatalog masing-masing. Tabel diisi sebagai tabel di database masing-masing.



Cara kerja izin

Sebaiknya integrasikan bucket tabel Anda dengan layanan AWS analitik sehingga Anda dapat bekerja dengan data tabel di seluruh layanan yang menggunakan AWS Glue Data Catalog sebagai penyimpanan metadata. Integrasi memungkinkan kontrol akses berbutir halus melalui. AWS Lake Formation Pendekatan keamanan ini berarti bahwa, selain izin AWS Identity and Access Management (IAM), Anda harus memberikan izin utama Lake Formation IAM Anda di tabel Anda sebelum Anda dapat bekerja dengannya.

Ada dua jenis izin utama di AWS Lake Formation:

- Izin akses metadata mengontrol kemampuan untuk membuat, membaca, memperbarui, dan menghapus database dan tabel metadata di Katalog Data.
- Izin akses data yang mendasari mengontrol kemampuan untuk membaca dan menulis data ke lokasi Amazon S3 yang mendasari yang ditunjuk oleh sumber daya Katalog Data.

Lake Formation menggunakan kombinasi model izinnya sendiri dan model izin IAM untuk mengontrol akses ke sumber daya Katalog Data dan data yang mendasarinya:

- Agar permintaan mengakses sumber daya Katalog Data atau data dasar agar berhasil, permintaan harus lulus pemeriksaan izin oleh IAM dan Lake Formation.
- Izin IAM mengontrol akses ke Lake Formation dan AWS Glue APIs dan resource, sedangkan izin Lake Formation mengontrol akses ke sumber daya Katalog Data, lokasi Amazon S3, dan data yang mendasarinya.

Izin Lake Formation hanya berlaku di Wilayah di mana mereka diberikan, dan kepala sekolah harus diberi wewenang oleh administrator danau data atau kepala sekolah lain dengan izin yang diperlukan untuk diberikan izin Lake Formation.

Untuk informasi selengkapnya, lihat Ikhtisar izin Lake Formation di Panduan AWS Lake Formation Pengembang.

Pastikan Anda mengikuti langkah-langkah the section called “Prasyarat untuk integrasi” dan the section called “Mengintegrasikan bucket tabel dengan AWS layanan analitik” agar Anda memiliki izin yang sesuai untuk mengakses dan sumber daya tabel Anda, AWS Glue Data Catalog dan untuk bekerja dengan layanan AWS analitik.

> **Important**
>
> Jika Anda bukan pengguna yang melakukan integrasi bucket tabel dengan layanan AWS analitik untuk akun Anda, Anda harus diberikan izin Lake Formation yang diperlukan di atas meja. Untuk informasi selengkapnya, lihat the section called “Memberikan izin pada tabel atau database”.

Prasyarat untuk integrasi

Prasyarat berikut diperlukan untuk mengintegrasikan bucket tabel dengan layanan analitik: AWS

- Buat ember meja.
- Lampirkan kebijakan AWSLakeFormationDataAdmin AWS terkelola ke prinsipal AWS Identity and Access Management (IAM) Anda untuk menjadikan pengguna tersebut administrator data lake. Untuk informasi selengkapnya tentang cara membuat administrator data lake, lihat Membuat administrator data lake di Panduan AWS Lake Formation Pengembang.
- Tambahkan izin untuk `glue:PassConnection` operasi ke kepala IAM Anda.
- Tambahkan izin untuk `lakeformation:RegisterResource` dan `lakeformation:RegisterResourceWithPrivilegedAccess` operasi ke kepala IAM Anda.
- Perbarui ke versi terbaru dari AWS Command Line Interface (AWS CLI).

> **Important**
>
> Saat membuat tabel, pastikan Anda menggunakan semua huruf kecil dalam nama tabel dan definisi tabel Anda. Misalnya, pastikan bahwa nama kolom Anda semua huruf kecil. Jika nama tabel atau definisi tabel berisi huruf kapital, tabel tidak didukung oleh AWS Lake Formation atau AWS Glue Data Catalog. Dalam hal ini, tabel Anda tidak akan terlihat oleh layanan AWS analitik seperti Amazon Athena, meskipun bucket tabel Anda terintegrasi dengan AWS layanan analitik.
> Jika definisi tabel berisi huruf kapital, Anda menerima pesan galat berikut saat menjalankan `SELECT` kueri di Athena: “GENERIC_INTERNAL_ERROR: Get table request failed: com.amazonaws.services.glue.model. ValidationException: Sumber Daya Federasi Tidak Didukung - Nama tabel atau kolom tidak valid.

Mengintegrasikan bucket tabel dengan AWS layanan analitik

Integrasi ini harus dilakukan satu kali per AWS Wilayah.

> **Important**
>
> Integrasi layanan AWS analitik sekarang menggunakan `WithPrivilegedAccess` opsi dalam operasi `registerResource` Lake Formation API untuk mendaftarkan bucket tabel

S3. Integrasi juga sekarang membuat `s3tablescatalog` katalog di AWS Glue Data Catalog dengan menggunakan `AllowFullTableExternalDataAccess` opsi dalam operasi `CreateCatalog` AWS Glue API.
Jika Anda mengatur integrasi dengan rilis pratinjau, Anda dapat terus menggunakan integrasi Anda saat ini. Namun, proses integrasi yang diperbarui memberikan peningkatan kinerja, jadi kami sarankan untuk bermigrasi. Untuk bermigrasi ke integrasi yang diperbarui, lihatthe section called “Migrasi ke proses integrasi yang diperbarui”.

Menggunakan konsol S3

1. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tabel.
3. Pilih Buat ember meja.

   Halaman bucket Create table terbuka.
4. Masukkan nama bucket Table dan pastikan kotak centang Aktifkan integrasi dipilih.
5. Pilih Buat ember meja. Amazon S3 akan mencoba mengintegrasikan bucket tabel Anda secara otomatis di Wilayah tersebut.

Pertama kali Anda mengintegrasikan bucket tabel di Wilayah mana pun, Amazon S3 membuat peran layanan IAM baru atas nama Anda. Peran ini memungkinkan Lake Formation mengakses semua bucket tabel di akun Anda dan akses federasi ke tabel Anda. AWS Glue Data Catalog

Menggunakan AWS CLI

Untuk mengintegrasikan ember meja menggunakan AWS CLI

Langkah-langkah berikut menunjukkan cara menggunakan bucket tabel AWS CLI untuk mengintegrasikan. Untuk menggunakan langkah-langkah ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

1. Buat ember meja.

```
aws s3tables create-table-bucket \
--region us-east-1 \
--name amzn-s3-demo-table-bucket
```

2. Buat peran layanan IAM yang memungkinkan Lake Formation mengakses sumber daya tabel Anda.

   a. Buat file bernama `Role-Trust-Policy.json` yang berisi kebijakan kepercayaan berikut:

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Sid": "LakeFormationDataAccessPolicy",
        "Effect": "Allow",
        "Principal": {
          "Service": "lakeformation.amazonaws.com"
        },
        "Action": [
            "sts:AssumeRole",
            "sts:SetContext",
            "sts:SetSourceIdentity"
        ],
        "Condition": {
          "StringEquals": {
            "aws:SourceAccount": "111122223333"
          }
        }
      }
    ]
}
```

   Buat peran layanan IAM dengan menggunakan perintah berikut:

```
aws iam create-role \
--role-name S3TablesRoleForLakeFormation \
--assume-role-policy-document file://Role-Trust-Policy.json
```

   b. Buat file bernama `LF-GluePolicy.json` yang berisi kebijakan berikut:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "LakeFormationPermissionsForS3ListTableBucket",
            "Effect": "Allow",
```

```
            "Action": [
                "s3tables:ListTableBuckets"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "LakeFormationDataAccessPermissionsForS3TableBucket",
            "Effect": "Allow",
            "Action": [
                "s3tables:CreateTableBucket",
                "s3tables:GetTableBucket",
                "s3tables:CreateNamespace",
                "s3tables:GetNamespace",
                "s3tables:ListNamespaces",
                "s3tables:DeleteNamespace",
                "s3tables:DeleteTableBucket",
                "s3tables:CreateTable",
                "s3tables:DeleteTable",
                "s3tables:GetTable",
                "s3tables:ListTables",
                "s3tables:RenameTable",
                "s3tables:UpdateTableMetadataLocation",
                "s3tables:GetTableMetadataLocation",
                "s3tables:GetTableData",
                "s3tables:PutTableData"
            ],
            "Resource": [
                "arn:aws:s3tables:us-east-1:111122223333:bucket/*"
            ]
        }
    ]
}
```

Lampirkan kebijakan ke peran dengan menggunakan perintah berikut:

```
aws iam put-role-policy \
--role-name S3TablesRoleForLakeFormation  \
--policy-name LakeFormationDataAccessPermissionsForS3TableBucket \
--policy-document file://LF-GluePolicy.json
```

3. Buat file bernama `input.json` yang berisi berikut ini:

```
{
    "ResourceArn": "arn:aws:s3tables:us-east-1:111122223333:bucket/*",

    "WithFederation": true,
    "RoleArn": "arn:aws:iam::111122223333:role/S3TablesRoleForLakeFormation"
}
```

Daftarkan bucket tabel dengan Lake Formation dengan menggunakan perintah berikut:

```
aws lakeformation register-resource \
--region us-east-1 \
--with-privileged-access \
--cli-input-json file://input.json
```

4. Buat file bernama `catalog.json` yang berisi katalog berikut:

```
{
   "Name": "s3tablescatalog",
   "CatalogInput": {
      "FederatedCatalog": {
          "Identifier": "arn:aws:s3tables:us-east-1:111122223333:bucket/*",
          "ConnectionName": "aws:s3tables"
       },
       "CreateDatabaseDefaultPermissions":[],
       "CreateTableDefaultPermissions":[],
       "AllowFullTableExternalDataAccess": "True"
   }
}
```

Buat `s3tablescatalog` katalog dengan menggunakan perintah berikut. Membuat katalog ini mengisi dengan objek yang sesuai AWS Glue Data Catalog dengan bucket tabel, ruang nama, dan tabel.

```
aws glue create-catalog \
--region us-east-1 \
--cli-input-json file://catalog.json
```

5. Verifikasi bahwa `s3tablescatalog` katalog telah ditambahkan AWS Glue dengan menggunakan perintah berikut:

```
aws glue get-catalog --catalog-id s3tablescatalog
```

Migrasi ke proses integrasi yang diperbarui

Proses integrasi layanan AWS analitik telah diperbarui. Jika Anda telah menyiapkan integrasi dengan rilis pratinjau, Anda dapat terus menggunakan integrasi Anda saat ini. Namun, proses integrasi yang diperbarui memberikan peningkatan kinerja, jadi kami sarankan untuk bermigrasi dengan menggunakan langkah-langkah berikut. Untuk informasi selengkapnya tentang proses migrasi atau integrasi, lihat Membuat katalog Tabel Amazon S3 AWS Glue Data Catalog di Panduan AWS Lake Formation Pengembang.

1. Buka AWS Lake Formation konsol dihttps://console.aws.amazon.com/lakeformation/, dan masuk sebagai administrator danau data. Untuk informasi selengkapnya tentang cara membuat administrator data lake, lihat Membuat administrator data lake di Panduan AWS Lake Formation Pengembang.
2. Hapus `s3tablescatalog` katalog Anda dengan melakukan hal berikut:
   - Di panel navigasi kiri, pilih Katalog.
   - Pilih tombol opsi di sebelah `s3tablescatalog` katalog dalam daftar Katalog. Dari menu Tindakan, pilih Hapus.
3. Batalkan pendaftaran lokasi data untuk `s3tablescatalog` katalog dengan melakukan hal berikut:
   - Di panel navigasi kiri, buka bagian Administrasi, dan pilih Lokasi danau data.
   - Pilih tombol opsi di sebelah lokasi danau `s3tablescatalog` data, misalnya,`s3://tables:`*`region`*`:`*`account-id`*`:bucket/*`.
   - Pada menu Tindakan, pilih Hapus.
   - Di kotak dialog konfirmasi yang muncul, pilih Hapus.
4. Setelah menghapus `s3tablescatalog` katalog dan lokasi data lake, Anda dapat mengikuti langkah-langkah untuk mengintegrasikan bucket tabel Anda dengan layanan AWS analitik dengan menggunakan proses integrasi yang diperbarui.

> **Note**
>
> Jika Anda ingin bekerja dengan tabel terenkripsi SSE-KMS dalam layanan AWS analitik terintegrasi, peran yang Anda gunakan harus memiliki izin untuk menggunakan kunci Anda AWS KMS untuk operasi enkripsi. Untuk informasi selengkapnya, lihat Memberikan izin kepada prinsipal IAM untuk bekerja dengan tabel terenkripsi dalam layanan analitik terintegrasi AWS.

Langkah selanjutnya

- Buat namespace.
- Membuat tabel.

Memberikan izin Lake Formation pada sumber daya tabel Anda

Setelah bucket tabel Anda terintegrasi dengan layanan AWS analitik, Lake Formation mengelola akses ke sumber daya tabel Anda. Lake Formation menggunakan model izinnya sendiri (izin Lake Formation) yang memungkinkan kontrol akses berbutir halus untuk sumber daya Katalog Data. Lake Formation mengharuskan setiap kepala sekolah IAM (pengguna atau peran) diberi wewenang untuk melakukan tindakan pada sumber daya yang dikelola Lake Formation. Untuk informasi selengkapnya, lihat Ikhtisar izin Lake Formation di Panduan AWS Lake Formation Pengembang. Untuk informasi tentang berbagi data lintas akun, lihat Berbagi data lintas akun di Lake Formation di Panduan AWS Lake Formation Pengembang.

Sebelum kepala sekolah IAM dapat mengakses tabel di layanan AWS analitik, Anda harus memberi mereka izin Lake Formation pada sumber daya tersebut.

> **Note**
>
> Jika Anda adalah pengguna yang melakukan integrasi bucket tabel, Anda sudah memiliki izin Lake Formation ke tabel Anda. Jika Anda satu-satunya kepala sekolah yang akan mengakses tabel Anda, Anda dapat melewati langkah ini. Anda hanya perlu memberikan izin Lake Formation di tabel Anda ke prinsipal IAM lainnya. Hal ini memungkinkan prinsipal lain untuk mengakses tabel saat menjalankan query. Untuk informasi selengkapnya, lihat Memberikan izin pada tabel atau database.

Anda harus memberikan izin Prinsipal Lake Formation IAM lainnya pada sumber daya tabel Anda untuk bekerja dengannya dalam layanan berikut:

- Amazon Redshift
- Amazon Data Firehose
- Amazon QuickSight
- Amazon Athena

Memberikan izin pada tabel atau database

Anda dapat memberikan izin utama Lake Formation pada tabel atau database dalam ember meja, baik melalui konsol Lake Formation atau. AWS CLI

> **Note**
>
> Saat Anda memberikan izin Lake Formation pada sumber daya Katalog Data ke akun eksternal atau langsung ke kepala IAM di akun lain, Lake Formation menggunakan layanan AWS Resource Access Manager (AWS RAM) untuk membagikan sumber daya. Jika akun penerima hibah berada di organisasi yang sama dengan akun pemberi hibah, sumber daya bersama segera tersedia untuk penerima hibah. Jika akun penerima hibah tidak berada di organisasi yang sama, AWS RAM kirimkan undangan ke akun penerima hibah untuk menerima atau menolak hibah sumber daya. Kemudian, untuk membuat sumber daya bersama tersedia, administrator data lake di akun penerima hibah harus menggunakan AWS RAM konsol atau AWS CLI untuk menerima undangan. Untuk informasi selengkapnya tentang berbagi data lintas akun, lihat Berbagi data lintas akun di Lake Formation di Panduan AWS Lake Formation Pengembang.

Console

1. Buka AWS Lake Formation konsol dihttps://console.aws.amazon.com/lakeformation/, dan masuk sebagai administrator danau data. Untuk informasi selengkapnya tentang cara membuat administrator data lake, lihat Membuat administrator data lake di Panduan AWS Lake Formation Pengembang.
2. Di panel navigasi, pilih Izin data, lalu pilih Hibah.
3. Pada halaman Izin Hibah, di bawah Prinsipal, lakukan salah satu hal berikut:

- Untuk Amazon Athena atau Amazon Redshift, pilih pengguna dan peran IAM, dan pilih prinsipal IAM yang Anda gunakan untuk kueri.
- Untuk Amazon Data Firehose, pilih pengguna dan peran IAM, lalu pilih peran layanan yang Anda buat untuk streaming ke tabel.
- Untuk QuickSight, pilih pengguna dan grup SAMP, lalu masukkan Nama Sumber Daya Amazon (ARN) pengguna admin QuickSight Anda.

4. Di bawah LF-tag atau sumber katalog, pilih Sumber daya Katalog Data Bernama.
5. Untuk Katalog, pilih subkatalog yang Anda buat saat mengintegrasikan bucket tabel, misalnya. *account-id*:s3tablescatalog/*amzn-s3-demo-bucket*
6. Untuk Database, pilih namespace bucket tabel S3 yang Anda buat.
7. (Opsional) Untuk Tabel, pilih tabel S3 yang Anda buat di bucket tabel Anda.

> **Note**
>
> Jika Anda membuat tabel baru di editor kueri Athena, jangan pilih tabel.

8. Lakukan salah satu tindakan berikut:
   - Jika Anda menentukan tabel di langkah sebelumnya, untuk izin Tabel, pilih Super.
   - Jika Anda tidak menentukan tabel pada langkah sebelumnya, buka izin Database. Untuk berbagi data lintas akun, Anda tidak dapat memilih Super untuk memberikan semua izin pada prinsipal lainnya di database Anda. Sebagai gantinya, pilih lebih banyak izin berbutir halus, seperti Deskripsikan.
9. PilihIzin.

CLI

1. Pastikan Anda menjalankan AWS CLI perintah berikut sebagai administrator data lake. Untuk informasi selengkapnya, lihat Membuat administrator data lake di Panduan AWS Lake Formation Pengembang.
2. Jalankan perintah berikut untuk memberikan izin Lake Formation pada tabel di bucket tabel S3 ke prinsipal IAM untuk mengakses tabel. Untuk menggunakan contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws lakeformation grant-permissions \
--region us-east-1 \
```

```
--cli-input-json \
'{
    "Principal": {
        "DataLakePrincipalIdentifier": "user or role ARN, for example,
 arn:aws:iam::account-id:role/example-role"
    },
    "Resource": {
        "Table": {
            "CatalogId": "account-id:s3tablescatalog/amzn-s3-demo-bucket",
            "DatabaseName": "S3 table bucket namespace, for example,
 test_namespace",
            "Name": "S3 table bucket table name, for example test_table"
        }
    },
    "Permissions": [
        "ALL"
    ]
}'
```

Mengakses tabel Amazon S3 menggunakan AWS Glue Iceberg REST titik akhir

Setelah bucket tabel S3 Anda terintegrasi dengan AWS Glue Data Catalog Anda dapat menggunakan AWS Glue Iceberg REST titik akhir untuk terhubung ke tabel S3 Anda dari Apache Iceberg-klien yang kompatibel, seperti PyIceberg atau Spark. The AWS Glue Iceberg REST endpoint mengimplementasikan Iceberg REST Spesifikasi API Terbuka Katalog yang menyediakan antarmuka standar untuk berinteraksi Iceberg tabel. Untuk mengakses tabel S3 menggunakan titik akhir, Anda perlu mengonfigurasi izin melalui kombinasi kebijakan dan hibah IAM. AWS Lake Formation Bagian berikut menjelaskan cara mengatur akses, termasuk membuat peran IAM yang diperlukan, mendefinisikan kebijakan yang diperlukan, dan menetapkan izin Lake Formation untuk akses database dan tingkat tabel.

Untuk panduan ujung ke ujung menggunakan PyIceberg, lihat Mengakses data di Tabel Amazon S3 menggunakan PyIceberg melalui AWS Glue Iceberg REST titik akhir.

Prasyarat

- Integrasikan bucket tabel Anda dengan layanan AWS analitik
- Buat namespace tabel

- Memiliki akses ke akun administrator data lake

Buat peran IAM untuk klien Anda

Untuk mengakses tabel melalui AWS Glue titik akhir, Anda perlu membuat peran IAM dengan izin ke dan tindakan Lake AWS Glue Formation. Prosedur ini menjelaskan cara membuat peran ini dan mengonfigurasi izinnya.

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi di sebelah kiri, pilih Kebijakan.
3. Pilih Buat kebijakan, lalu pilih JSON di editor kebijakan.
4. Tambahkan kebijakan sebaris berikut yang memberikan izin untuk mengakses dan tindakan Lake AWS Glue Formation:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "glue:GetCatalog",
                "glue:GetDatabase",
                "glue:GetDatabases",
                "glue:GetTable",
                "glue:GetTables",
                "glue:CreateTable",
                "glue:UpdateTable"
            ],
            "Resource": [
                "arn:aws:glue:<region>:<account-id>:catalog",
                "arn:aws:glue:<region>:<account-id>:catalog/s3tablescatalog",
                "arn:aws:glue:<region>:<account-id>:catalog/
s3tablescatalog/<s3_table_bucket_name>",
                "arn:aws:glue:<region>:<account-id>:table/
s3tablescatalog/<s3_table_bucket_name>/<namespace>/*",
                "arn:aws:glue:<region>:<account-id>:database/
s3tablescatalog/<s3_table_bucket_name>/<namespace>"
            ]
        },
        {
```

```
            "Effect": "Allow",
            "Action": [
                "lakeformation:GetDataAccess"
            ],
            "Resource": "*"
        }
    ]
}
```

5. Setelah membuat kebijakan, buat peran IAM dan pilih Kebijakan kepercayaan khusus sebagai jenis entitas Tepercaya.
6. Masukkan yang berikut ini untuk kebijakan kepercayaan kustom.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::<accountid>:role/<Admin_role>"
            },
            "Action": "sts:AssumeRole",
            "Condition": {}
        }
    ]
}
```

Tentukan akses di Lake Formation

Lake Formation menyediakan kontrol akses berbutir halus untuk tabel data lake Anda. Saat Anda mengintegrasikan bucket S3 Anda dengan AWS Glue Data Catalog, tabel Anda secara otomatis terdaftar sebagai sumber daya di Lake Formation. Untuk mengakses tabel ini, Anda harus memberikan izin Lake Formation tertentu ke identitas IAM Anda, selain izin kebijakan IAM-nya.

Langkah-langkah berikut menjelaskan cara menerapkan kontrol akses Lake Formation untuk mengizinkan Anda Iceberg klien untuk terhubung ke tabel Anda. Anda harus masuk sebagai administrator data lake untuk menerapkan izin ini.

Izinkan mesin eksternal mengakses data tabel

Di Lake Formation, Anda harus mengaktifkan akses tabel penuh untuk mesin eksternal untuk mengakses data. Hal ini memungkinkan aplikasi pihak ketiga untuk mendapatkan kredensi sementara dari Lake Formation saat menggunakan peran IAM yang memiliki izin penuh pada tabel yang diminta.

Buka konsol Lake Formation di https://console.aws.amazon.com/lakeformation/.

1. Buka konsol Lake Formation di https://console.aws.amazon.com/lakeformation/, dan masuk sebagai administrator danau data.
2. Di panel navigasi di bawah Administrasi, pilih Pengaturan integrasi aplikasi.
3. Pilih Izinkan mesin eksternal mengakses data di lokasi Amazon S3 dengan akses tabel penuh. Lalu, pilih Simpan.

Berikan izin Lake Formation pada sumber daya tabel Anda

Selanjutnya, berikan izin Lake Formation ke peran IAM yang Anda buat untuk klien yang kompatibel dengan Iceberg. Izin ini akan memungkinkan peran untuk membuat dan mengelola tabel di namespace Anda. Anda perlu memberikan izin database dan tingkat tabel:

Untuk memberikan izin database

1. Buka AWS Lake Formation konsol di https://console.aws.amazon.com/lakeformation/, dan masuk sebagai administrator danau data.
2. Di panel navigasi, pilih Izin data lalu pilih Hibah.
3. Pada halaman Izin Hibah, di bawah Prinsipal, pilih pengguna dan peran IAM dan pilih peran IAM yang Anda buat untuk Iceberg AWS Glue REST endpoint akses.
4. Di bawah LF-tag atau sumber katalog, pilih Sumber daya Katalog Data Bernama.
5. Untuk Katalog, pilih katalog AWS Glue data yang dibuat untuk bucket tabel Anda. Misalnya, `<accoundID>:s3tablescatalog/<table-bucket-name>`.
6. Untuk Database, pilih`mynamespace`.
7. Untuk izin Tabel, pilih Buat tabel dan Jelaskan.
8. PilihIzin.

Untuk memberikan izin tabel

1. Buka AWS Lake Formation konsol di https://console.aws.amazon.com/lakeformation/, dan masuk sebagai administrator danau data.
2. Di panel navigasi, pilih Izin data lalu pilih Hibah.
3. Pada halaman Izin Hibah, di bawah Prinsipal, pilih pengguna dan peran IAM dan pilih peran IAM yang Anda buat untuk Iceberg AWS Glue REST endpoint akses.
4. Di bawah LF-tag atau sumber katalog, pilih Sumber daya Katalog Data Bernama.
5. Untuk Katalog, pilih katalog AWS Glue data yang dibuat untuk bucket tabel Anda. Misalnya, *<accoundID>*:s3tablescatalog/*<table-bucket-name>*.
6. Untuk Database, pilih namespace bucket tabel S3 yang Anda buat.
7. Untuk Tabel, pilih ALL_TABLES.
8. Untuk izin Tabel, pilih Super.
9. PilihIzin.

Siapkan lingkungan Anda untuk menggunakan titik akhir

Setelah Anda mengatur peran IAM dengan izin yang diperlukan untuk akses tabel, Anda dapat menggunakannya untuk menjalankannya Iceberg klien dari mesin lokal Anda dengan mengonfigurasi AWS CLI dengan peran Anda, menggunakan perintah berikut:

```
aws sts assume-role --role-arn "arn:aws:iam::<accountid>:role/<glue-irc-role>" --role-session-name <glue-irc-role>
```

Untuk mengakses tabel melalui AWS Glue REST endpoint, Anda perlu menginisialisasi katalog di Iceberg-klien yang kompatibel. Inisialisasi ini memerlukan penentuan properti kustom, termasuk properti sigv4, URI titik akhir, dan lokasi gudang. Tentukan properti ini sebagai berikut:

- Properti Sigv4 - Sigv4 harus diaktifkan, nama penandatanganan adalah `glue`
- Lokasi gudang - Ini adalah ember meja Anda, yang ditentukan dalam format ini: *<accountid>*:s3tablescatalog/*<table-bucket-name>*
- URI titik akhir - Lihat panduan referensi titik akhir AWS Glue layanan untuk titik akhir khusus wilayah

Contoh berikut menunjukkan bagaimana menginisialisasi pyIceberg katalog.

```
rest_catalog = load_catalog(
        s3tablescatalog,
**{
"type": "rest",
"warehouse": "<accountid>:s3tablescatalog/<table-bucket-name>",
"uri": "https://glue.<region>.amazonaws.com/iceberg",
"rest.sigv4-enabled": "true",
"rest.signing-name": "glue",
"rest.signing-region": region
        }
)
```

Untuk informasi tambahan tentang AWS Glue Iceberg REST implementasi endpoint, lihat Menghubungkan ke Katalog Data menggunakan AWS Glue Iceberg REST titik akhir dalam Panduan AWS Glue Pengguna.

Mengakses tabel menggunakan Tabel Amazon S3 Iceberg REST titik akhir

Anda dapat menghubungkan Iceberg REST klien ke Tabel Amazon S3 Iceberg REST Endpoint dan make REST API panggilan untuk membuat, memperbarui, atau menanyakan tabel di bucket tabel S3. Titik akhir mengimplementasikan satu set standar Iceberg REST APIs ditentukan dalam Apache Iceberg REST Katalog Spesifikasi API Terbuka. Endpoint bekerja dengan menerjemahkan Iceberg REST API operasi ke dalam operasi Tabel S3 yang sesuai.

> **Note**
>
> Tabel Amazon S3 Iceberg REST endpoint dapat digunakan untuk mengakses tabel dalam implementasi katalog AWS Partner Network (APN) atau implementasi katalog kustom. Ini juga dapat digunakan jika Anda hanya memerlukan akses baca/tulis dasar ke ember meja tunggal. Untuk skenario akses lainnya, kami sarankan menggunakan AWS Glue Iceberg REST endpoint untuk terhubung ke tabel, yang menyediakan manajemen tabel terpadu, tata kelola terpusat, dan kontrol akses berbutir halus. Untuk informasi selengkapnya, silakan lihat Mengakses tabel Amazon S3 menggunakan AWS Glue Iceberg REST titik akhir

Mengkonfigurasi titik akhir

Anda terhubung ke Tabel Amazon S3 Iceberg REST titik akhir menggunakan titik akhir layanan. Tabel S3 Iceberg REST endpoint memiliki format berikut:

```
https://s3tables.<REGION>.amazonaws.com/iceberg
```

Lihat Tabel S3 Wilayah AWS dan titik akhir untuk titik akhir khusus Wilayah.

Properti konfigurasi katalog

Saat menggunakan klien Iceberg untuk menghubungkan mesin analitik ke titik akhir layanan, Anda harus menentukan properti konfigurasi berikut saat menginisialisasi katalog. Ganti *placeholder values* dengan informasi untuk Wilayah dan keranjang meja Anda.

- Titik akhir khusus wilayah sebagai URI titik akhir: `https://s3tables.<REGION>.amazonaws.com/iceberg`
- Bucket meja ARN anda sebagai lokasi gudang: `arn:aws:s3tables:<region>:<accountID>:bucket/<bucketname>`
- Properti Sigv4 untuk otentikasi. Nama penandatanganan SigV4 untuk permintaan titik akhir layanan adalah: `s3tables`

Contoh berikut menunjukkan cara mengonfigurasi klien yang berbeda untuk menggunakan Tabel Amazon S3 Iceberg REST titik akhir.

PyIceberg

Untuk menggunakan Tabel Amazon S3 Iceberg REST titik akhir dengan PyIceberg, tentukan properti konfigurasi aplikasi berikut:

```
rest_catalog = load_catalog(
  catalog_name,
  **{
    "type": "rest",
    "warehouse":"arn:aws:s3tables:<Region>:<accountID>:bucket/<bucketname>",
    "uri": "https://s3tables.<Region>.amazonaws.com/iceberg",
    "rest.sigv4-enabled": "true",
    "rest.signing-name": "s3tables",
    "rest.signing-region": "<Region>"
  }
)
```

Apache Spark

Untuk menggunakan Tabel Amazon S3 Iceberg REST titik akhir dengan Spark, tentukan properti konfigurasi aplikasi berikut, ganti *placeholder values* dengan informasi untuk Region dan bucket tabel Anda.

```
spark-shell \
  --packages "org.apache.iceberg:iceberg-spark-
runtime-3.5_2.12:1.4.1,software.amazon.awssdk:bundle:2.20.160,software.amazon.awssdk:url-
connection-client:2.20.160" \
  --master "local[*]" \
  --conf
 "spark.sql.extensions=org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions"
 \
  --conf "spark.sql.defaultCatalog=spark_catalog" \
   --conf "spark.sql.catalog.spark_catalog=org.apache.iceberg.spark.SparkCatalog" \
  --conf "spark.sql.catalog.spark_catalog.type=rest" \
  --conf "spark.sql.catalog.spark_catalog.uri=https://
s3tables.<Region>.amazonaws.com/iceberg" \
  --conf
 "spark.sql.catalog.spark_catalog.warehouse=arn:aws:s3tables:<Region>:<accountID>:bucket/
<bucketname>" \
  --conf "spark.sql.catalog.spark_catalog.rest.sigv4-enabled=true" \
  --conf "spark.sql.catalog.spark_catalog.rest.signing-name=s3tables" \
  --conf "spark.sql.catalog.spark_catalog.rest.signing-region=<Region>" \
  --conf "spark.sql.catalog.spark_catalog.io-
impl=org.apache.iceberg.aws.s3.S3FileIO" \
  --conf
 "spark.hadoop.fs.s3a.aws.credentials.provider=org.apache.hadoop.fs.s3a.SimpleAWSCredentialP
 \
  --conf "spark.sql.catalog.spark_catalog.rest-metrics-reporting-enabled=false"
```

Mengautentikasi dan mengotorisasi akses ke titik akhir

Permintaan API ke titik akhir layanan Tabel S3 diautentikasi menggunakan AWS Signature Version 4 (SigV4). Lihat AWS Sigv4 Versi Tanda Tangan untuk permintaan API guna mempelajari lebih lanjut tentang AWS SigV4.

Nama penandatanganan SiGv4 untuk Tabel Amazon S3 Iceberg REST permintaan titik akhir adalah: `s3tables`

Permintaan ke Tabel Amazon S3 Iceberg REST titik akhir diotorisasi menggunakan tindakan `s3tables` IAM yang sesuai dengan REST Operasi API. Izin ini dapat didefinisikan dalam kebijakan berbasis identitas IAM atau kebijakan berbasis sumber daya yang dilampirkan pada tabel dan bucket tabel. Untuk informasi selengkapnya, lihat Manajemen akses untuk Tabel S3.

Anda dapat melacak permintaan yang dibuat ke tabel Anda melalui REST titik akhir dengan AWS CloudTrail. Permintaan akan dicatat sebagai tindakan IAM S3 yang sesuai. Misalnya, `LoadTable` API akan menghasilkan peristiwa manajemen untuk `GetTableMetadataLocation` operasi dan peristiwa data untuk `GetTableData` operasi. Untuk informasi selengkapnya, lihat Logging dengan AWS CloudTrail untuk Tabel S3.

Parameter awalan dan jalur

Iceberg REST katalog APIs memiliki awalan bentuk bebas dalam permintaan mereka. URLs Misalnya, panggilan `ListNamespaces` API menggunakan format `GET/v1/{prefix}/namespaces` URL. Untuk Tabel S3, jalur REST `{prefix}` selalu ARN bucket tabel yang disandikan url Anda.

Misalnya, untuk keranjang tabel ARN berikut: `arn:aws:s3tables:`*`us-east-1`*`:111122223333:bucket/`*`bucketname`* awalannya adalah: `arn%3Aaws%3As3tables%3A`*`us-east-1`*`%3A`*`111122223333`*`%3Abucket%2F`*`bucketname`*

Parameter jalur namespace

Ruang nama dalam sebuah Iceberg REST jalur API katalog dapat memiliki beberapa level. Namun, Tabel S3 hanya mendukung ruang nama tingkat tunggal. Untuk mengakses namespace dalam hierarki katalog multi-level, Anda dapat terhubung ke katalog multi-level di atas namespace saat mereferensikan namespace. Ini memungkinkan mesin kueri apa pun yang mendukung notasi 3 bagian `catalog.namespace.table` untuk mengakses objek dalam hierarki katalog Tabel S3 tanpa masalah kompatibilitas dibandingkan dengan menggunakan namespace multi-level.

Didukung Iceberg REST Operasi API

Tabel berikut berisi Iceberg REST yang didukung APIs dan bagaimana mereka sesuai dengan tindakan Tabel S3.

Operasi Iceberg REST	Jalur REST	S3 Tabel aksi IAM	CloudTrail EventName
`getConfig`	`GET /v1/config`	`s3tables: GetTableBucket`	`s3tables: GetTableBucket`

Operasi Iceberg REST	Jalur REST	S3 Tabel aksi IAM	CloudTrail EventName
listNamespaces	GET /v1/{prefix}/namespaces	s3tables:ListNamespaces	s3tables:ListNamespaces
createNamespace	POST /v1/{prefix}/namespaces	s3tables:CreateNamespace	s3tables:CreateNamespace
loadNamespaceMetadata	GET /v1/{prefix}/namespaces/{namespace}	s3tables:GetNamespace	s3tables:GetNamespace
dropNamespace	DELETE /v1/{prefix}/namespaces/{namespace}	s3tables:DeleteNamespace	s3tables:DeleteNamespace
listTables	GET /v1/{prefix}/namespaces/{namespace}/tables	s3tables:ListTables	s3tables:ListTables
createTable	POST /v1/{prefix}/namespaces/{namespace}/tables	s3tables:CreateTable , s3tables:PutTableData	s3tables:CreateTable , s3tables:PutObject
loadTable	GET /v1/{prefix}/namespaces/{namespace}/tables/{table}	s3tables:GetTableMetadataLocation , s3tables:GetTableData	s3tables:GetTableMetadataLocation , s3tables:GetObject

Operasi Iceberg REST	Jalur REST	S3 Tabel aksi IAM	CloudTrail EventName
updateTable	POST /v1/{prefix}/namespaces/{namespace}/tables/{table}	s3tables:UpdateTableMetadataLocation, s3tables:PutTableData, s3tables:GetTableData	s3tables:UpdateTableMetadataLocation, s3tables:PutObject, s3tables:GetObject
dropTable	DELETE /v1/{prefix}/namespaces/{namespace}/tables/{table}	s3tables:DeleteTable	s3tables:DeleteTable
renameTable	POST /v1/{prefix}/tables/rename	s3tables:RenameTable	s3tables:RenameTable
tableExists	HEAD /v1/{prefix}/namespaces/{namespace}/tables/{table}	s3tables:GetTable	s3tables:GetTable
namespaceExists	HEAD /v1/{prefix}/namespaces/{namespace}	s3tables:GetNamespace	s3tables:GetNamespace

Pertimbangan dan batasan

Berikut ini adalah pertimbangan dan batasan saat menggunakan Tabel Amazon S3 Iceberg REST titik akhir.

Pertimbangan-pertimbangan

- CreateTable Perilaku API — `stage-create` Opsi ini tidak didukung untuk operasi ini, dan menghasilkan `400 Bad Request` kesalahan. Ini berarti Anda tidak dapat membuat tabel dari hasil kueri menggunakan `CREATE TABLE AS SELECT` (CTAS).
- DeleteTable Perilaku API - Anda hanya dapat menjatuhkan tabel dengan pembersihan diaktifkan. Menjatuhkan tabel dengan `purge=false` tidak didukung dan menghasilkan `400 Bad Request` kesalahan. Beberapa versi Spark selalu setel flag ini ke false bahkan saat menjalankan `DROP TABLE PURGE` perintah. Anda dapat mencoba dengan `DROP TABLE PURGE` atau menggunakan DeleteTableoperasi Tabel S3 untuk menghapus tabel.
- Titik akhir hanya mendukung operasi metadata tabel standar. Untuk pemeliharaan tabel, seperti manajemen snapshot dan pemadatan, gunakan operasi API pemeliharaan Tabel S3. Untuk informasi selengkapnya, lihat Pemeliharaan Tabel S3.

Batasan

- Ruang nama bertingkat tidak didukung.
- OAuthotentikasi berbasis tidak didukung.
- Hanya `owner` properti yang didukung untuk ruang nama.
- Terkait tampilan APIs didefinisikan dalam Apache Iceberg REST Spesifikasi API terbuka tidak didukung.
- Menjalankan operasi di atas meja dengan `metadata.json` file lebih dari 5MB tidak didukung, dan akan mengembalikan `400 Bad Request` kesalahan. Untuk mengontrol ukuran `metadata.json` file Anda, gunakan operasi pemeliharaan tabel. Untuk informasi selengkapnya, lihat Pemeliharaan Tabel S3.

Mengakses tabel Amazon S3 dengan Katalog Tabel Amazon S3 untuk Apache Iceberg

Anda dapat mengakses tabel S3 dari mesin kueri sumber terbuka seperti Apache Spark dengan menggunakan Katalog Tabel Amazon S3 untuk Apache Iceberg katalog klien. Katalog Tabel Amazon S3 untuk Apache Iceberg adalah perpustakaan sumber terbuka yang diselenggarakan oleh AWS Labs. Ia bekerja dengan menerjemahkan Apache Iceberg operasi di mesin kueri Anda (seperti penemuan tabel, pembaruan metadata, dan penambahan atau penghapusan tabel) ke dalam operasi API Tabel S3.

Katalog Tabel Amazon S3 untuk Apache Iceberg didistribusikan sebagai Maven JAR disebuts3-tables-catalog-for-iceberg.jar. Anda dapat membangun katalog klien JAR dari AWS Labs GitHubrepositori_atau unduh dari Maven. Saat menghubungkan ke tabel, katalog klien JAR digunakan sebagai ketergantungan saat Anda menginisialisasi a Spark sesi untuk Apache Iceberg.

Menggunakan Katalog Tabel Amazon S3 untuk Apache Iceberg dengan Apache Spark

Anda dapat menggunakan Katalog Tabel Amazon S3 untuk Apache Iceberg katalog klien untuk terhubung ke tabel dari aplikasi sumber terbuka saat Anda menginisialisasi Spark sesi. Dalam konfigurasi sesi Anda, Anda menentukan Iceberg dan dependensi Amazon S3, dan buat katalog khusus yang menggunakan keranjang tabel Anda sebagai gudang metadata.

Prasyarat

- Identitas IAM dengan akses ke keranjang tabel dan tindakan Tabel S3 Anda. Untuk informasi selengkapnya, lihat Manajemen akses untuk Tabel S3.

Untuk menginisialisasi a Spark sesi menggunakan Katalog Tabel Amazon S3 untuk Apache Iceberg

- Inisialisasi Spark menggunakan perintah berikut. Untuk menggunakan perintah, ganti ganti Katalog Tabel Amazon S3 untuk Apache Iceberg *version number*dengan versi terbaru dari AWS Labs GitHub repositori, dan *table bucket ARN* dengan ARN ember meja Anda sendiri.

```
spark-shell \
--packages org.apache.iceberg:iceberg-spark-
runtime-3.5_2.12:1.6.1,software.amazon.s3tables:s3-tables-catalog-for-iceberg-
runtime:0.1.4 \
--conf spark.sql.catalog.s3tablesbucket=org.apache.iceberg.spark.SparkCatalog \
--conf spark.sql.catalog.s3tablesbucket.catalog-
impl=software.amazon.s3tables.iceberg.S3TablesCatalog \
--conf spark.sql.catalog.s3tablesbucket.warehouse=arn:aws:s3tables:us-
east-1:111122223333:bucket/amzn-s3-demo-table-bucket \
--conf
 spark.sql.extensions=org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions
```

Menanyakan tabel S3 dengan Spark SQL

Penggunaan Spark, Anda dapat menjalankan operasi DQL, DML, dan DDL pada tabel S3. Saat Anda menanyakan tabel, Anda menggunakan nama tabel yang sepenuhnya memenuhi syarat, termasuk nama katalog sesi mengikuti pola ini:

CatalogName.*NamespaceName*.*TableName*

Contoh kueri berikut menunjukkan beberapa cara Anda dapat berinteraksi dengan tabel S3. Untuk menggunakan contoh kueri ini di mesin kueri Anda, ganti *user input placeholder* nilainya dengan milik Anda sendiri.

Untuk menanyakan tabel dengan Spark

- Buat namespace

```
spark.sql(" CREATE NAMESPACE IF NOT EXISTS s3tablesbucket.my_namespace")
```

- Membuat tabel

```
spark.sql(" CREATE TABLE IF NOT EXISTS s3tablesbucket.my_namespace.`my_table`
( id INT, name STRING, value INT ) USING iceberg ")
```

- Mengkueri Tabel

```
spark.sql(" SELECT * FROM s3tablesbucket.my_namespace.`my_table` ").show()
```

- Masukkan data ke dalam tabel

```
spark.sql(
"""
    INSERT INTO s3tablesbucket.my_namespace.my_table
    VALUES
        (1, 'ABC', 100),
        (2, 'XYZ', 200)
""")
```

- Memuat file data yang ada ke dalam tabel

1. Baca data ke Spark.

```
val data_file_location = "Path such as S3 URI to data file"
```

```
val data_file = spark.read.parquet(data_file_location)
```

2. Tulis data ke dalam tabel Gunung Es.

```
data_file.writeTo("s3tablesbucket.my_namespace.my_table").using("Iceberg").tableProperty
 ("format-version", "2").createOrReplace()
```

Menanyakan tabel Amazon S3 dengan Athena

Amazon Athena adalah layanan kueri interaktif yang dapat Anda gunakan untuk menganalisis data secara langsung di Amazon S3 dengan menggunakan SQL standar. Untuk informasi selengkapnya, silakan lihat Apa yang dimaksud dengan Amazon Athena? di Panduan Pengguna Amazon Athena.

Setelah mengintegrasikan bucket tabel dengan layanan AWS analitik, Anda dapat menjalankan kueri Data Definition Language (DDL), Data Manipulation Language (DHTML), dan Data Query Language (DQL) pada tabel S3 menggunakan Athena. Untuk informasi selengkapnya tentang cara menanyakan tabel di keranjang tabel, lihat Mendaftarkan katalog bucket Tabel S3 di Panduan Pengguna Amazon Athena.

Anda juga dapat menjalankan kueri di Athena dari konsol Amazon S3.

Menggunakan konsol S3 dan Amazon Athena

Prosedur berikut menggunakan konsol Amazon S3 untuk mengakses editor kueri Athena sehingga Anda dapat menanyakan tabel dengan Amazon Athena.

> **Note**
>
> Sebelum melakukan langkah-langkah berikut, pastikan Anda telah mengintegrasikan bucket tabel Anda dengan layanan AWS analitik di Wilayah ini. Untuk informasi selengkapnya, lihat the section called “Menggunakan Tabel S3 dengan layanan AWS analitik”.

Untuk menanyakan tabel

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tabel.

3. Pada halaman bucket Table, pilih bucket yang berisi tabel yang ingin Anda kueri.
4. Pada halaman detail bucket, pilih tombol opsi di sebelah nama tabel yang ingin Anda kueri.
5. Pilih tabel Query dengan Athena.
6. Konsol Amazon Athena terbuka dan editor kueri Athena muncul dengan contoh `SELECT` kueri yang dimuat untuk Anda. Ubah kueri ini sesuai kebutuhan untuk kasus penggunaan Anda.

   Di editor kueri, bidang Katalog harus diisi dengan s3tablescatalog/diikuti dengan nama ember meja Anda, misalnya, s3tablescatalog/***`amzn-s3-demo-bucket`***. Bidang Database harus diisi dengan namespace tempat tabel Anda disimpan.

   > **Note**
   >
   > Jika Anda tidak melihat nilai ini di bidang Katalog dan Database, pastikan Anda telah mengintegrasikan bucket tabel Anda dengan layanan AWS analitik di Wilayah ini. Untuk informasi selengkapnya, lihat the section called “Menggunakan Tabel S3 dengan layanan AWS analitik”.

7. Untuk menjalankan kueri, pilih Jalankan.

   > **Note**
   >
   > - Jika Anda menerima kesalahan “Izin tidak cukup untuk menjalankan kueri. Principal tidak memiliki hak istimewa pada sumber daya tertentu “ketika Anda mencoba menjalankan kueri di Athena, Anda harus diberikan izin Formasi Danau yang diperlukan di atas meja. Untuk informasi selengkapnya, lihat the section called “Memberikan izin pada tabel atau database”.
   > - Jika Anda menerima kesalahan “Gunung es tidak dapat mengakses sumber daya yang diminta” ketika Anda mencoba menjalankan kueri, buka AWS Lake Formation konsol dan pastikan bahwa Anda telah memberikan diri Anda izin pada katalog keranjang tabel dan database (namespace) yang Anda buat. Jangan tentukan tabel saat memberikan izin ini. Untuk informasi selengkapnya, lihat the section called “Memberikan izin pada tabel atau database”.
   > - Jika Anda menerima pesan galat berikut saat menjalankan `SELECT` kueri di Athena, pesan ini disebabkan oleh adanya huruf kapital di nama tabel atau nama kolom Anda dalam definisi tabel Anda: “GENERIC_INTERNAL_ERROR: Get table request failed: com.amazonaws.services.glue.model. ValidationException: Sumber Daya Federasi

Tidak Didukung - Nama tabel atau kolom tidak valid.” Pastikan bahwa nama tabel dan kolom Anda semuanya huruf kecil.

Mengakses tabel Amazon S3 dengan Amazon Redshift

Amazon Redshift adalah layanan gudang data berskala petabyte yang cepat, dikelola sepenuhnya, yang membuatnya sederhana dan hemat biaya untuk menganalisis semua data Anda secara efisien menggunakan alat intelijen bisnis yang ada. Redshift Tanpa Server memungkinkan Anda mengakses dan menganalisis data tanpa semua konfigurasi gudang data yang disediakan. Untuk informasi selengkapnya, lihat Memulai gudang data tanpa server di Panduan Memulai Amazon Redshift.

Kueri tabel Amazon S3 dengan Amazon Redshift

Prasyarat

- Integrasikan bucket tabel Anda dengan layanan AWS analitik.
  - Buat namespace.
  - Membuat tabel.
- Memberikan izin Lake Formation pada sumber daya tabel Anda.

Setelah menyelesaikan prasyarat, Anda dapat mulai menggunakan Amazon Redshift untuk menanyakan tabel dengan salah satu cara berikut:

- Menggunakan editor kueri Amazon Redshift v2
- Menyambung ke gudang data Amazon Redshift menggunakan alat klien SQL
- Menggunakan Amazon Redshift Data API

Mengakses tabel Amazon S3 dengan Amazon EMR

Amazon EMR (sebelumnya disebut Amazon Elastic MapReduce) adalah platform cluster terkelola yang menyederhanakan menjalankan kerangka kerja data besar, seperti Apache Hadoop danApache Spark, AWS untuk memproses dan menganalisis sejumlah besar data. Dengan menggunakan kerangka kerja ini dan proyek sumber terbuka terkait, Anda dapat memproses data untuk tujuan analitik dan beban kerja intelijen bisnis. Amazon EMR juga memungkinkan Anda mengubah dan

memindahkan sejumlah besar data ke dalam dan keluar dari penyimpanan AWS data dan database lainnya.

Anda dapat menggunakan Apache Iceberg cluster di Amazon EMR untuk bekerja dengan tabel S3 dengan menghubungkan ke bucket tabel dalam satu sesi. Spark Untuk menyambung ke bucket tabel di Amazon EMR, Anda dapat menggunakan AWS integrasi layanan analitik AWS Glue Data Catalog melalui, atau Anda dapat menggunakan Katalog Tabel Amazon S3 open source untuk katalog klien. Apache Iceberg

> **Note**
>
> Tabel S3 didukung di Amazon EMR versi 7.5 atau lebih tinggi.

Menghubungkan ke bucket tabel S3 dengan Spark pada cluster EMR Amazon Iceberg

Dalam prosedur ini, Anda menyiapkan kluster EMR Amazon yang dikonfigurasi untuk Apache Iceberg dan kemudian meluncurkan Spark sesi yang terhubung ke bucket tabel Anda. Anda dapat mengaturnya menggunakan integrasi layanan AWS analitik AWS Glue, atau Anda dapat menggunakan Katalog Tabel Amazon S3 open source untuk katalog Apache Iceberg klien. Untuk informasi tentang katalog klien, lihatMengakses tabel menggunakan Tabel Amazon S3 Iceberg REST titik akhir.

Pilih metode Anda menggunakan tabel dengan Amazon EMR dari opsi berikut.

Amazon S3 Tables Catalog for Gunung Es Apache

Prasyarat berikut diperlukan untuk menanyakan tabel dengan EMR Amazon menggunakan Spark Katalog Tabel Amazon S3 untuk. Apache Iceberg

Prasyarat

- Lampirkan `AmazonS3TablesFullAccess` kebijakan ke peran IAM yang Anda gunakan untuk Amazon EMR.

Untuk menyiapkan kluster EMR Amazon ke tabel kueri dengan Spark

1. Buat cluster dengan konfigurasi berikut. Untuk menggunakan contoh ini, ganti *`user input placeholders`*dengan informasi Anda sendiri.

```
aws emr create-cluster --release-label emr-7.5.0 \
--applications Name=Spark \
--configurations file://configurations.json \
--region us-east-1 \
--name My_Spark_Iceberg_Cluster \
--log-uri s3://amzn-s3-demo-bucket/ \
--instance-type m5.xlarge \
--instance-count 2 \
--service-role EMR_DefaultRole \
--ec2-attributes \

InstanceProfile=EMR_EC2_DefaultRole,SubnetId=subnet-1234567890abcdef0,KeyName=my-key-pair
```

`configurations.json`:

```
[{
"Classification":"iceberg-defaults",
"Properties":{"iceberg.enabled":"true"}
}]
```

2. Connect ke node Spark utama menggunakan SSH.
3. Untuk menginisialisasi Spark sesi Iceberg yang terhubung ke keranjang tabel Anda, masukkan perintah berikut. Ganti *user input placeholders* dengan ember meja ARN Anda.

```
spark-shell \
--packages software.amazon.s3tables:s3-tables-catalog-for-iceberg-runtime:0.1.3
 \
--conf spark.sql.catalog.s3tablesbucket=org.apache.iceberg.spark.SparkCatalog \
--conf spark.sql.catalog.s3tablesbucket.catalog-
impl=software.amazon.s3tables.iceberg.S3TablesCatalog \
--conf spark.sql.catalog.s3tablesbucket.warehouse=arn:aws:s3tables:us-
east-1:111122223333:bucket/amzn-s3-demo-bucket1 \
--conf spark.sql.defaultCatalog=s3tablesbucket \
--conf
 spark.sql.extensions=org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions
```

4. Kueri tabel Anda dengan Spark SQL. Misalnya kueri, lihatthe section called “Menanyakan tabel S3 dengan Spark SQL”.

AWS analytics services integration

Prasyarat berikut diperlukan untuk menanyakan tabel dengan EMR Spark Amazon menggunakan integrasi layanan analitik. AWS

Prasyarat

- Integrasikan bucket tabel Anda dengan layanan AWS analitik.
- Buat peran layanan default untuk Amazon EMR ()`EMR_DefaultRole_V2`. Untuk detailnya, lihat Peran layanan untuk Amazon EMR (peran EMR).
- Buat profil EC2 instans Amazon untuk Amazon EMR ()`EMR_EC2_DefaultRole`. Untuk detailnya, lihat Peran layanan untuk EC2 instance klaster (profil EC2 instance).
  - Lampirkan `AmazonS3TablesFullAccess` kebijakan ke`EMR_EC2_DefaultRole`.

Untuk menyiapkan kluster EMR Amazon ke tabel kueri dengan Spark

1. Buat cluster dengan konfigurasi berikut. Untuk menggunakan contoh ini, ganti *`user input placeholder`* nilai dengan informasi Anda sendiri.

```
aws emr create-cluster --release-label emr-7.5.0 \
--applications Name=Spark \
--configurations file://configurations.json \
--region us-east-1 \
--name My_Spark_Iceberg_Cluster \
--log-uri s3://amzn-s3-demo-bucket/ \
--instance-type m5.xlarge \
--instance-count 2 \
--service-role EMR_DefaultRole \
--ec2-attributes \

InstanceProfile=EMR_EC2_DefaultRole,SubnetId=subnet-1234567890abcdef0,KeyName=my-key-pair
```

`configurations.json`:

```
[{
"Classification":"iceberg-defaults",
"Properties":{"iceberg.enabled":"true"}
}]
```

2. Connect ke node Spark utama menggunakan SSH.
3. Masukkan perintah berikut untuk menginisialisasi Spark sesi Iceberg yang terhubung ke tabel Anda. Ganti *user input placeholders* dengan informasi Anda sendiri.

```
spark-shell \
--conf
 spark.sql.extensions=org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions
 \
--conf spark.sql.defaultCatalog=s3tables \
--conf spark.sql.catalog.s3tables=org.apache.iceberg.spark.SparkCatalog \
--conf spark.sql.catalog.s3tables.catalog-
impl=org.apache.iceberg.aws.glue.GlueCatalog \
--conf spark.sql.catalog.s3tables.client.region=us-east-1 \
--conf spark.sql.catalog.s3tables.glue.id=111122223333
```

4. Kueri tabel Anda dengan Spark SQL. Misalnya kueri, lihat the section called “Menanyakan tabel S3 dengan Spark SQL”

> **Note**
>
> Jika Anda menggunakan `DROP TABLE PURGE` perintah dengan Amazon EMR:
>
> - Amazon EMR versi 7.5
>
> Setel konfigurasi `spark.sql.catalog.`*`your-catalog-name`*`.cache-enabled` Spark ke. `false` Jika konfigurasi ini diatur `ketrue`, jalankan perintah dalam sesi atau aplikasi baru sehingga cache tabel tidak diaktifkan.
> - Amazon EMR versi lebih tinggi dari 7,5
>
> `DROP TABLE` tidak didukung. Anda dapat menggunakan `DeleteTable` REST API Tabel S3 untuk menghapus tabel.

Memvisualisasikan data tabel dengan QuickSight

QuickSight adalah layanan analisis bisnis cepat untuk membangun visualisasi, melakukan analisis ad hoc, dan dengan cepat mendapatkan wawasan bisnis dari data Anda. QuickSight menemukan sumber AWS data dengan mulus, memungkinkan organisasi untuk menskalakan hingga ratusan ribu pengguna, dan memberikan kinerja kueri yang cepat dan responsif dengan menggunakan QuickSight

Super-cepat, Paralel, Dalam Memori, Mesin Perhitungan (SPICE). Untuk informasi lebih lanjut, lihat Apa itu QuickSight? dalam panduan QuickSight pengguna.

Setelah Anda Mengintegrasikan bucket tabel Anda dengan layanan AWS analitik, Anda dapat membuat kumpulan data dari tabel Anda dan bekerja dengannya dalam QuickSight menggunakan SPICE atau kueri SQL langsung dari mesin kueri Anda. QuickSight mendukung Athena sebagai sumber data untuk tabel S3.

Konfigurasikan izin QuickSight untuk mengakses tabel

Sebelum bekerja dengan data tabel S3 di QuickSight Anda harus memberikan izin ke peran QuickSight layanan, pengguna QuickSight admin, dan memberikan izin Lake Formation pada tabel yang ingin Anda akses.

Berikan izin untuk peran QuickSight layanan

Saat disiapkan QuickSight untuk pertama kalinya di akun Anda, AWS buat peran layanan yang memungkinkan QuickSight untuk mengakses sumber data di AWS layanan lain, seperti Athena atau Amazon Redshift. Nama peran default adalah `aws-quicksight-service-role-v0`.

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Pilih Peran dan pilih peran QuickSight layanan. Nama defaultnya adalah `aws-quicksight-service-role-v0`
3. Pilih Tambahkan izin lalu Buat kebijakan sebaris.
4. Pilih JSON untuk membuka editor kebijakan JSON, lalu tambahkan kebijakan berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "glue:GetCatalog",
      "Resource": "*"
    }
  ]
}
```

5. Pilih Berikutnya, masukkan nama Kebijakan, lalu Buat kebijakan.

Untuk mengkonfigurasi izin untuk pengguna QuickSight admin

1. Jalankan AWS CLI perintah berikut untuk menemukan ARN pengguna QuickSight admin Anda.

```
aws quicksight list-users --aws-account-id 111122223333 --namespace default --
region region
```

2. Berikan izin Lake Formation ke ARN ini. Lihat perinciannya di Memberikan izin Lake Formation pada sumber daya tabel Anda.

Menggunakan data tabel di QuickSight

Anda dapat terhubung ke data tabel menggunakan Athena sebagai sumber data.

Prasyarat

- Integrasikan bucket tabel Anda dengan layanan AWS analitik.
  - Buat namespace.
  - Membuat tabel.
  - Konfigurasikan izin QuickSight untuk mengakses tabel.
- Mendaftar untuk QuickSight.

1. Masuk ke QuickSight akun Anda di https://quicksight.aws.amazon.com/
2. Di dasbor, pilih Analisis baru.
3. Pilih Dataset baru.
4. Pilih Athena.
5. Masukkan nama sumber data, lalu pilih Buat sumber data.
6. Pilih Gunakan SQL khusus. Anda tidak akan dapat memilih tabel Anda dari panel Pilih tabel Anda.
7. Masukkan kueri Athena SQL yang menangkap kolom yang ingin Anda visualisasikan, lalu pilih Konfirmasi kueri. Misalnya, gunakan kueri berikut untuk memilih semua kolom:

```
SELECT * FROM "s3tablescatalog/table-bucket-name".namespace.table-name
```

8. Pilih Visualisasikan untuk menganalisis data dan mulai membangun dasbor. Untuk informasi selengkapnya, lihat Memvisualisasikan data QuickSight dan Menjelajahi dasbor interaktif di QuickSight

Streaming data ke tabel dengan Amazon Data Firehose

Amazon Data Firehose adalah layanan yang dikelola sepenuhnya untuk mengirimkan data streaming real-time ke tujuan seperti Amazon S3, Amazon Redshift, OpenSearch Amazon Service, tabel SplunkApache Iceberg, dan titik akhir HTTP kustom atau titik akhir HTTP yang dimiliki oleh penyedia layanan pihak ketiga yang didukung. Dengan Amazon Data Firehose, Anda tidak perlu menulis aplikasi atau mengelola sumber daya. Anda mengonfigurasi produsen data Anda untuk mengirim data ke Firehose, dan secara otomatis mengirimkan data ke tujuan yang Anda tentukan. Anda juga dapat mengonfigurasi Firehose untuk mengubah data Anda sebelum mengirimkannya. Untuk mempelajari lebih lanjut tentang Amazon Data Firehose, lihat Apa itu Amazon Data Firehose?

Selesaikan langkah-langkah ini untuk mengatur streaming Firehose ke tabel di bucket tabel S3:

1. Integrasikan bucket tabel Anda dengan layanan AWS analitik.
2. Konfigurasikan Firehose untuk mengirimkan data ke tabel S3 Anda. Untuk melakukannya, Anda membuat peran layanan AWS Identity and Access Management (IAM) yang memungkinkan Firehose mengakses tabel Anda.
3. Berikan izin eksplisit peran layanan Firehose ke namespace tabel atau tabel Anda. Untuk informasi selengkapnya, lihat Berikan izin Lake Formation pada sumber daya tabel Anda.
4. Buat aliran Firehose yang merutekan data ke tabel Anda.

Membuat peran Firehose untuk menggunakan tabel S3 sebagai tujuan

Firehose memerlukan peran layanan IAM dengan izin khusus untuk mengakses AWS Glue tabel dan menulis data ke tabel S3. Anda membutuhkan ini untuk menyediakan peran IAM ini saat membuat aliran Firehose.

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi kiri, pilih Kebijakan
3. Pilih Buat kebijakan, lalu pilih JSON di editor kebijakan.
4. Tambahkan kebijakan inline berikut yang memberikan izin ke semua database dan tabel di katalog data Anda. Jika mau, Anda dapat memberikan izin hanya untuk tabel dan database

tertentu. Untuk menggunakan kebijakan ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "S3TableAccessViaGlueFederation",
      "Effect": "Allow",
      "Action": [
        "glue:GetTable",
        "glue:GetDatabase",
        "glue:UpdateTable"
      ],
      "Resource": [
        "arn:aws:glue:region:account-id:catalog/s3tablescatalog/*",
        "arn:aws:glue:region:account-id:catalog/s3tablescatalog",
        "arn:aws:glue:region:account-id:catalog",
        "arn:aws:glue:region:account-id:database/*",
        "arn:aws:glue:region:account-id:table/*/*"
      ]
    },
    {
      "Sid": "S3DeliveryErrorBucketPermission",
      "Effect": "Allow",
      "Action": [
        "s3:AbortMultipartUpload",
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::error delivery bucket",
        "arn:aws:s3:::error delivery bucket/*"
      ]
    },
    {
      "Sid": "RequiredWhenUsingKinesisDataStreamsAsSource",
      "Effect": "Allow",
      "Action": [
        "kinesis:DescribeStream",
```

```
        "kinesis:GetShardIterator",
        "kinesis:GetRecords",
        "kinesis:ListShards"
      ],
      "Resource": "arn:aws:kinesis:region:account-id:stream/stream-name"
    },
    {
      "Sid":
 "RequiredWhenDoingMetadataReadsANDDataAndMetadataWriteViaLakeformation",
      "Effect": "Allow",
      "Action": [
        "lakeformation:GetDataAccess"
      ],
      "Resource": "*"
    },
    {
      "Sid": "RequiredWhenUsingKMSEncryptionForS3ErrorBucketDelivery",
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
      ],
      "Resource": [
        "arn:aws:kms:region:account-id:key/KMS-key-id"
      ],
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "s3.region.amazonaws.com"
        },
        "StringLike": {
          "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::error delivery bucket/
prefix*"
        }
      }
    },
    {
      "Sid": "LoggingInCloudWatch",
      "Effect": "Allow",
      "Action": [
        "logs:PutLogEvents"
      ],
      "Resource": [
        "arn:aws:logs:region:account-id:log-group:log-group-name:log-stream:log-
stream-name"
```

```
      ]
    },
    {
      "Sid": "RequiredWhenAttachingLambdaToFirehose",
      "Effect": "Allow",
      "Action": [
        "lambda:InvokeFunction",
        "lambda:GetFunctionConfiguration"
      ],
      "Resource": [
        "arn:aws:lambda:region:account-id:function:function-name:function-version"
      ]
    }
  ]
}
```

Kebijakan ini memiliki pernyataan yang memungkinkan akses ke Kinesis Data Streams, menjalankan fungsi Lambda dan akses ke kunci. AWS KMS Jika Anda tidak menggunakan salah satu sumber daya ini, Anda dapat menghapus pernyataan masing-masing.

Jika pencatatan kesalahan diaktifkan, Firehose juga mengirimkan kesalahan pengiriman data ke grup CloudWatch log dan aliran Anda. Untuk ini, Anda harus mengonfigurasi grup log dan nama aliran log. Untuk nama grup log dan aliran log, lihat Memantau Amazon Data Firehose Menggunakan CloudWatch Log.

5. Setelah Anda membuat kebijakan, buat peran IAM dengan AWS layanan sebagai jenis entitas Tepercaya.
6. Untuk Service atau use case, pilih Kinesis. Untuk kasus Penggunaan pilih Kinesis Firehose.
7. Pilih Berikutnya, lalu pilih kebijakan yang Anda buat sebelumnya.
8. Beri nama peran Anda. Tinjau detail peran Anda, dan pilih Buat peran. Peran tersebut akan memiliki kebijakan kepercayaan berikut.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole"
            ],
            "Principal": {
```

```
                "Service": [
                    "firehose.amazonaws.com"
                ]
            }
        }
    ]
}
```

Membuat aliran Firehose ke tabel S3

Prosedur berikut menunjukkan cara membuat aliran Firehose untuk mengirimkan data ke tabel S3 menggunakan konsol. Prasyarat berikut diperlukan untuk menyiapkan aliran Firehose ke tabel S3.

Prasyarat

- Integrasikan bucket tabel Anda dengan layanan AWS analitik.
  - Buat namespace.
  - Membuat tabel.
- Buat Role for Firehose untuk mengakses Tabel S3.
- Berikan izin Lake Formation ke peran layanan Firehose yang Anda buat untuk mengakses tabel.

Untuk memberikan informasi perutean ke Firehose saat mengonfigurasi aliran, Anda menggunakan namespace sebagai nama database dan nama tabel di namespace tersebut. Anda dapat menggunakan nilai-nilai ini di bagian kunci Unik dari konfigurasi aliran Firehose untuk merutekan data ke satu tabel. Anda juga dapat menggunakan nilai-nilai ini untuk rute ke tabel menggunakan ekspresi JSON Query. Untuk informasi selengkapnya, lihat Merutekan catatan masuk ke satu tabel Gunung Es.

Untuk mengatur aliran Firehose ke tabel S3 (Konsol)

1. Buka konsol Firehose di. https://console.aws.amazon.com/firehose/
2. Pilih Buat aliran Firehose.
3. Untuk Sumber, pilih salah satu sumber berikut:
   - Amazon Kinesis Data Streams
   - Amazon MSK
   - Langsung PUT

4. Untuk Tujuan, pilih Apache IcebergTabel.
5. Masukkan nama aliran Firehose.
6. Konfigurasikan pengaturan Sumber Anda.
7. Untuk pengaturan Tujuan, pilih Akun saat ini untuk melakukan streaming ke tabel di akun Anda atau Akun silang untuk tabel di akun lain.
   - Untuk tabel di akun saat ini, pilih katalog Tabel S3 Anda dari tarik-turun Katalog.
   - Untuk tabel di akun Lintas, masukkan ARN Katalog dari katalog yang ingin Anda streaming di akun lain.
8. Konfigurasikan nama database dan tabel menggunakan konfigurasi Kunci Unik, JSONQuery ekspresi, atau dalam fungsi Lambda. Untuk informasi selengkapnya, lihat Rutekan catatan masuk ke satu tabel Gunung Es dan Rute catatan masuk ke tabel Gunung Es yang berbeda di Panduan Pengembang Firehose Data Amazon.
9. Di bawah Pengaturan Backup, tentukan bucket cadangan S3.
10. Untuk peran IAM yang ada di bawah Pengaturan lanjutan, pilih peran IAM yang Anda buat untuk Firehose.
11. Pilih Buat aliran Firehose.

Untuk informasi selengkapnya tentang setelan lain yang dapat Anda konfigurasikan untuk streaming, lihat Menyiapkan aliran Firehose di Panduan Pengembang Firehose Data Amazon.

Menjalankan pekerjaan ETL di tabel Amazon S3 dengan AWS Glue

AWS Glue adalah layanan integrasi data tanpa server yang memudahkan pengguna analitik untuk menemukan, menyiapkan, memindahkan, dan mengintegrasikan data dari berbagai sumber. Anda dapat menggunakan AWS Glue pekerjaan untuk menjalankan pipeline ekstrak, transformasi, dan pemuatan (ETL) untuk memuat data ke dalam data lake Anda. Untuk informasi lebih lanjut tentang AWS Glue, lihat Apa itu AWS Glue? di Panduan AWS Glue Pengembang.

AWS Glue Pekerjaan merangkum skrip yang terhubung ke data sumber Anda, memprosesnya, dan kemudian menuliskannya ke target data Anda. Biasanya, sebuah tugas menjalankan skrip extract, transform, and load (ETL). Pekerjaan dapat menjalankan skrip yang dirancang untuk lingkungan Apache Spark runtime. Anda dapat memantau eksekusi tugas untuk memahami metrik waktu aktif seperti status penyelesaian, durasi, dan waktu mulai.

Anda dapat menggunakan AWS Glue pekerjaan untuk memproses data dalam tabel S3 Anda dengan menghubungkan ke tabel Anda melalui integrasi dengan layanan AWS analitik, atau, terhubung

langsung menggunakan titik akhir Iceberg REST Tabel Amazon S3 atau Katalog Tabel Amazon S3 untuk. Apache Iceberg Panduan ini mencakup langkah-langkah dasar untuk mulai menggunakan AWS Glue dengan Tabel S3, termasuk:

Topik

- Prasyarat
- Buat skrip untuk terhubung ke bucket tabel
- Buat AWS Glue pekerjaan yang menanyakan tabel

> **Note**
>
> Tabel S3 didukung pada AWS Glue versi 5.0 atau lebih tinggi.

Prasyarat

Sebelum Anda dapat melakukan kueri tabel dari AWS Glue pekerjaan, Anda harus mengonfigurasi peran IAM yang AWS Glue dapat digunakan untuk menjalankan pekerjaan, dan mengunggah Katalog Apache Iceberg JAR Tabel Amazon S3 ke bucket S3 AWS Glue yang dapat diakses saat menjalankan pekerjaan.

- Integrasikan bucket tabel Anda dengan layanan AWS analitik.
- Buat peran IAM untuk AWS Glue.
  - Lampirkan kebijakan yang `AmazonS3TablesFullAccess` dikelola ke peran.
  - Lampirkan kebijakan yang `AmazonS3FullAccess` dikelola ke peran.
- (Opsional) Jika Anda menggunakan Katalog Tabel Amazon S3 untuk Apache Iceberg Anda perlu mengunduh katalog klien JAR dan mengunggahnya ke ember S3.

  Mengunduh katalog JAR

  1. Periksa versi terbaru di Maven Central. Anda dapat mengunduh JAR dari pusat Maven menggunakan browser Anda, atau menggunakan perintah berikut. Pastikan untuk mengganti *`version number`* dengan versi terbaru.

     ```
     wget https://repo1.maven.org/maven2/software/amazon/s3tables/s3-tables-catalog-for-iceberg-runtime/0.1.5/s3-tables-catalog-for-iceberg-runtime-0.1.5.jar
     ```

2. Unggah unduhan JAR ke bucket S3 yang dapat diakses oleh peran AWS Glue IAM Anda. Anda dapat menggunakan AWS CLI perintah berikut untuk mengunggah fileJAR. Pastikan untuk mengganti *version number* dengan versi terbaru, dan *bucket name* dan *path* dengan versi Anda sendiri.

```
aws s3 cp s3-tables-catalog-for-iceberg-runtime-0.1.5.jar s3://amzn-s3-demo-bucket/jars/
```

Buat skrip untuk terhubung ke bucket tabel

Untuk mengakses data tabel saat menjalankan tugas AWS Glue ETL, Anda mengonfigurasi Spark sesi untuk Apache Iceberg yang terhubung ke bucket tabel S3 Anda. Anda dapat memodifikasi skrip yang ada untuk terhubung ke keranjang tabel Anda atau membuat skrip baru. Untuk informasi selengkapnya tentang membuat AWS Glue skrip, lihat Tutorial: Menulis skrip AWS Glue untuk Spark di Panduan AWS Glue Pengembang.

Anda dapat mengonfigurasi sesi untuk terhubung ke bucket tabel Anda melalui salah satu metode akses Tabel S3 berikut:

- Integrasi Tabel S3 dengan layanan AWS analitik
- Titik akhir Tabel Amazon S3 Iceberg REST
- Katalog Tabel Amazon S3 untuk Apache Iceberg

Pilih dari metode akses berikut untuk melihat instruksi penyiapan dan contoh konfigurasi.

AWS analytics services integration

Sebagai prasyarat untuk menanyakan tabel dengan Spark AWS Glue menggunakan integrasi layanan AWS analitik, Anda harus Mengintegrasikan bucket tabel Anda dengan layanan analitik AWS

Anda dapat mengonfigurasi koneksi ke keranjang meja Anda melalui Spark sesi dalam pekerjaan atau dengan AWS Glue Studio sihir dalam sesi interaktif. Untuk menggunakan contoh berikut, ganti *placeholder values* dengan informasi untuk ember meja Anda sendiri.

Menggunakan PySpark skrip

Gunakan cuplikan kode berikut dalam PySpark skrip untuk mengonfigurasi AWS Glue pekerjaan agar terhubung ke keranjang tabel menggunakan integrasi.

```
spark = SparkSession.builder.appName("SparkIcebergSQL") \
    .config("spark.sql.extensions",
 "org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions") \
    .config("spark.sql.defaultCatalog","s3tables") \
    .config("spark.sql.catalog.s3tables",
 "org.apache.iceberg.spark.SparkCatalog") \
    .config("spark.sql.catalog.s3tables.catalog-impl",
 "org.apache.iceberg.aws.glue.GlueCatalog") \
    .config("spark.sql.catalog.s3tables.glue.id",
 "111122223333:s3tablescatalog/amzn-s3-demo-table-bucket") \
    .config("spark.sql.catalog.s3tables.warehouse", "s3://amzn-s3-demo-table-
bucket/warehouse/") \
    .getOrCreate()
```

Menggunakan AWS Glue sesi interaktif

Jika Anda menggunakan sesi notebook interaktif dengan AWS Glue 5.0, tentukan konfigurasi yang sama menggunakan `%%configure` sihir dalam sel sebelum eksekusi kode.

```
%%configure
{"conf":
 "spark.sql.extensions=org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions
 --conf spark.sql.defaultCatalog=s3tables --conf
 spark.sql.catalog.s3tables=org.apache.iceberg.spark.SparkCatalog --conf
 spark.sql.catalog.s3tables.catalog-impl=org.apache.iceberg.aws.glue.GlueCatalog
 --conf spark.sql.catalog.s3tables.glue.id=111122223333:s3tablescatalog/amzn-s3-
demo-table-bucket --conf spark.sql.catalog.s3tables.warehouse=s3://amzn-s3-demo-
table-bucket/warehouse/"}
```

Amazon S3 Tables Iceberg REST endpoint

Anda dapat mengonfigurasi koneksi ke keranjang meja Anda melalui Spark sesi dalam pekerjaan atau dengan AWS Glue Studio sihir dalam sesi interaktif. Untuk menggunakan contoh berikut, ganti *`placeholder values`* dengan informasi untuk ember meja Anda sendiri.

Menggunakan PySpark skrip

Gunakan cuplikan kode berikut dalam PySpark skrip untuk mengonfigurasi AWS Glue pekerjaan agar terhubung ke keranjang tabel menggunakan titik akhir.

```
spark = SparkSession.builder.appName("glue-s3-tables-rest") \
    .config("spark.sql.extensions",
 "org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions") \
    .config("spark.sql.defaultCatalog", "s3_rest_catalog") \
    .config("spark.sql.catalog.s3_rest_catalog",
 "org.apache.iceberg.spark.SparkCatalog") \
    .config("spark.sql.catalog.s3_rest_catalog.type", "rest") \
    .config("spark.sql.catalog.s3_rest_catalog.uri", "https://
s3tables.Region.amazonaws.com/iceberg") \
    .config("spark.sql.catalog.s3_rest_catalog.warehouse",
 "arn:aws:s3tables:Region:111122223333:s3tablescatalog/amzn-s3-demo-table-
bucket") \
    .config("spark.sql.catalog.s3_rest_catalog.rest.sigv4-enabled", "true") \
    .config("spark.sql.catalog.s3_rest_catalog.rest.signing-name", "s3tables") \
    .config("spark.sql.catalog.s3_rest_catalog.rest.signing-region", "Region") \
    .config('spark.sql.catalog.s3_rest_catalog.io-
impl','org.apache.iceberg.aws.s3.S3FileIO') \
    .config('spark.sql.catalog.s3_rest_catalog.rest-metrics-reporting-
enabled','false') \
    .getOrCreate()
```

Menggunakan AWS Glue sesi interaktif

Jika Anda menggunakan sesi notebook interaktif dengan AWS Glue 5.0, tentukan konfigurasi yang sama menggunakan `%%configure` sihir dalam sel sebelum eksekusi kode. Ganti nilai placeholder dengan informasi untuk bucket tabel Anda sendiri.

```
%%configure
{"conf":
 "spark.sql.extensions=org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions
 --conf spark.sql.defaultCatalog=s3_rest_catalog --conf
 spark.sql.catalog.s3_rest_catalog=org.apache.iceberg.spark.SparkCatalog
 --conf spark.sql.catalog.s3_rest_catalog.type=rest --conf
 spark.sql.catalog.s3_rest_catalog.uri=https://s3tables.Region.amazonaws.com/
iceberg --conf
 spark.sql.catalog.s3_rest_catalog.warehouse=arn:aws:s3tables:Region:111122223333:s3table
amzn-s3-demo-table-bucket --conf spark.sql.catalog.s3_rest_catalog.rest.sigv4-
enabled=true --conf spark.sql.catalog.s3_rest_catalog.rest.signing-name=s3tables
```

```
 --conf spark.sql.catalog.s3_rest_catalog.rest.signing-region=Region --conf
 spark.sql.catalog.s3_rest_catalog.io-impl=org.apache.iceberg.aws.s3.S3FileIO --
conf spark.sql.catalog.s3_rest_catalog.rest-metrics-reporting-enabled=false"}
```

Amazon S3 Tables Catalog for Gunung Es Apache

Sebagai prasyarat untuk menghubungkan ke tabel menggunakan Katalog Tabel Amazon S3 untuk Apache Iceberg Anda harus terlebih dahulu mengunduh toples katalog terbaru dan mengunggahnya ke ember S3. Kemudian, ketika Anda membuat pekerjaan Anda, Anda menambahkan jalur ke katalog klien JAR sebagai parameter khusus. Untuk informasi selengkapnya tentang parameter pekerjaan di AWS Glue, lihat Parameter khusus yang digunakan dalam AWS Glue pekerjaan di Panduan AWS Glue Pengembang.

Anda dapat mengonfigurasi koneksi ke keranjang meja Anda melalui Spark sesi dalam pekerjaan atau dengan AWS Glue Studio sihir dalam sesi interaktif. Untuk menggunakan contoh berikut, ganti *`placeholder values`* dengan informasi untuk ember meja Anda sendiri.

Menggunakan PySpark skrip

Gunakan cuplikan kode berikut dalam PySpark skrip untuk mengonfigurasi AWS Glue pekerjaan agar terhubung ke bucket tabel Anda menggunakan file. JAR Ganti nilai placeholder dengan informasi untuk bucket tabel Anda sendiri.

```
spark = SparkSession.builder.appName("glue-s3-tables") \
    .config("spark.sql.extensions",
 "org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions") \
    .config("spark.sql.defaultCatalog", "s3tablesbucket") \
    .config("spark.sql.catalog.s3tablesbucket",
 "org.apache.iceberg.spark.SparkCatalog") \
    .config("spark.sql.catalog.s3tablesbucket.catalog-impl",
 "software.amazon.s3tables.iceberg.S3TablesCatalog") \
    .config("spark.sql.catalog.s3tablesbucket.warehouse",
 "arn:aws:s3tables:Region:111122223333:bucket/amzn-s3-demo-table-bucket") \
    .getOrCreate()
```

Menggunakan AWS Glue sesi interaktif

Jika Anda menggunakan sesi notebook interaktif dengan AWS Glue 5.0, tentukan konfigurasi yang sama menggunakan `%%configure` sihir dalam sel sebelum eksekusi kode. Ganti nilai placeholder dengan informasi untuk bucket tabel Anda sendiri.

```
%%configure
{"conf":
 "spark.sql.extensions=org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions
 --conf spark.sql.defaultCatalog=s3tablesbucket --conf
 spark.sql.catalog.s3tablesbucket=org.apache.iceberg.spark.SparkCatalog
 --conf spark.sql.catalog.s3tablesbucket.catalog-
impl=software.amazon.s3tables.iceberg.S3TablesCatalog --conf
 spark.sql.catalog.s3tablesbucket.warehouse=arn:aws:s3tables:Region:111122223333:bucket/
amzn-s3-demo-table-bucket", "extra-jars": "s3://amzn-s3-demo-bucket/jars/s3-
tables-catalog-for-iceberg-runtime-0.1.5.jar"}
```

Contoh skrip

Contoh PySpark skrip berikut dapat digunakan untuk menguji kueri tabel S3 dengan pekerjaan. AWS Glue Skrip ini terhubung ke bucket tabel Anda dan menjalankan kueri untuk: membuat namespace baru, membuat tabel sampel, menyisipkan data ke dalam tabel, dan mengembalikan data tabel. Untuk menggunakan skrip, ganti *placeholder values* dengan informasi untuk bucket tabel Anda sendiri.

Pilih dari skrip berikut berdasarkan metode akses Tabel S3 Anda.

S3 Tables integration with AWS analytics services

```
from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("SparkIcebergSQL") \
    .config("spark.sql.extensions",
 "org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions") \
    .config("spark.sql.defaultCatalog","s3tables")
    .config("spark.sql.catalog.s3tables", "org.apache.iceberg.spark.SparkCatalog") \
    .config("spark.sql.catalog.s3tables.catalog-impl",
 "org.apache.iceberg.aws.glue.GlueCatalog") \
    .config("spark.sql.catalog.s3tables.glue.id",
 "111122223333:s3tablescatalog/amzn-s3-demo-table-bucket") \
    .config("spark.sql.catalog.s3tables.warehouse", "s3://amzn-s3-demo-table-bucket/
bucket/amzn-s3-demo-table-bucket") \
    .getOrCreate()

namespace = "new_namespace"
table = "new_table"
```

```
spark.sql("SHOW DATABASES").show()

spark.sql(f"DESCRIBE NAMESPACE {namespace}").show()

spark.sql(f"""
    CREATE TABLE IF NOT EXISTS {namespace}.{table} (
       id INT,
       name STRING,
       value INT
    )
""")

spark.sql(f"""
    INSERT INTO {namespace}.{table}
    VALUES
       (1, 'ABC', 100),
       (2, 'XYZ', 200)
""")

spark.sql(f"SELECT * FROM {namespace}.{table} LIMIT 10").show()
```

Amazon S3 Tables Iceberg REST endpoint

```
from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("glue-s3-tables-rest") \
    .config("spark.sql.extensions",
 "org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions") \
    .config("spark.sql.defaultCatalog", "s3_rest_catalog") \
    .config("spark.sql.catalog.s3_rest_catalog",
 "org.apache.iceberg.spark.SparkCatalog") \
    .config("spark.sql.catalog.s3_rest_catalog.type", "rest") \
    .config("spark.sql.catalog.s3_rest_catalog.uri", "https://
s3tables.Region.amazonaws.com/iceberg") \
    .config("spark.sql.catalog.s3_rest_catalog.warehouse",
 "arn:aws:s3tables:Region:111122223333:bucket/amzn-s3-demo-table-bucket") \
    .config("spark.sql.catalog.s3_rest_catalog.rest.sigv4-enabled", "true") \
    .config("spark.sql.catalog.s3_rest_catalog.rest.signing-name", "s3tables") \
    .config("spark.sql.catalog.s3_rest_catalog.rest.signing-region", "Region") \
    .config('spark.sql.catalog.s3_rest_catalog.io-
impl','org.apache.iceberg.aws.s3.S3FileIO') \
    .config('spark.sql.catalog.s3_rest_catalog.rest-metrics-reporting-
enabled','false') \
```

```
    .getOrCreate()

namespace = "s3_tables_rest_namespace"
table = "new_table_s3_rest"

spark.sql("SHOW DATABASES").show()

spark.sql(f"DESCRIBE NAMESPACE {namespace}").show()

spark.sql(f"""
    CREATE TABLE IF NOT EXISTS {namespace}.{table} (
       id INT,
       name STRING,
       value INT
    )
""")

spark.sql(f"""
    INSERT INTO {namespace}.{table}
    VALUES
       (1, 'ABC', 100),
       (2, 'XYZ', 200)
""")

spark.sql(f"SELECT * FROM {namespace}.{table} LIMIT 10").show()
```

Amazon S3 Tables Catalog for Gunung Es Apache

```
from pyspark.sql import SparkSession

#Spark session configurations
spark = SparkSession.builder.appName("glue-s3-tables") \
    .config("spark.sql.extensions",
 "org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions") \
    .config("spark.sql.defaultCatalog", "s3tablesbucket") \
    .config("spark.sql.catalog.s3tablesbucket",
 "org.apache.iceberg.spark.SparkCatalog") \
    .config("spark.sql.catalog.s3tablesbucket.catalog-impl",
 "software.amazon.s3tables.iceberg.S3TablesCatalog") \
    .config("spark.sql.catalog.s3tablesbucket.warehouse",
 "arn:aws:s3tables:Region:111122223333:bucket/amzn-s3-demo-table-bucket") \
```

```
    .getOrCreate()

#Script
namespace = "new_namespace"
table = "new_table"

spark.sql(f"CREATE NAMESPACE IF NOT EXISTS s3tablesbucket.{namespace}")
spark.sql(f"DESCRIBE NAMESPACE {namespace}").show()

spark.sql(f"""
    CREATE TABLE IF NOT EXISTS {namespace}.{table} (
       id INT,
       name STRING,
       value INT
    )
""")

spark.sql(f"""
    INSERT INTO {namespace}.{table}
    VALUES
       (1, 'ABC', 100),
       (2, 'XYZ', 200)
""")

spark.sql(f"SELECT * FROM {namespace}.{table} LIMIT 10").show()
```

Buat AWS Glue pekerjaan yang menanyakan tabel

Prosedur berikut menunjukkan cara mengatur AWS Glue pekerjaan yang terhubung ke bucket tabel S3 Anda. Anda dapat melakukan ini menggunakan AWS CLI atau menggunakan konsol dengan editor AWS Glue Studio skrip. Untuk informasi selengkapnya, lihat Menulis lowongan AWS Glue di Panduan AWS Glue Pengguna.

Menggunakan editor AWS Glue Studio skrip

Prosedur berikut menunjukkan cara menggunakan editor AWS Glue Studio skrip untuk membuat pekerjaan ETL yang menanyakan tabel S3 Anda.

Prasyarat

- Prasyarat
- Buat skrip untuk terhubung ke bucket tabel

1. Buka AWS Glue konsol di https://console.aws.amazon.com/glue/.
2. Dari panel Navigasi, pilih pekerjaan ETL.
3. Pilih Editor skrip, lalu pilih Unggah skrip dan unggah PySpark skrip yang Anda buat untuk menanyakan tabel S3.
4. Pilih tab Job details dan masukkan yang berikut ini untuk properti Basic.
   - Untuk Nama, masukkan nama untuk pekerjaan itu.
   - Untuk Peran IAM, pilih peran yang Anda buat AWS Glue.
5. (Opsional) Jika Anda menggunakan Katalog Tabel Amazon S3 untuk metode Apache Iceberg akses, perluas properti Lanjutan, dan untuk JARs jalur Dependent, masukkan URI S3 dari jar katalog klien yang Anda unggah ke bucket S3 sebagai prasyarat. Misalnya, s3://*amzn-s3-demo-bucket1*/*jars*tables-catalog-for-iceberg/s3- -runtime- .jar *0.1.5*
6. Pilih Simpan untuk membuat pekerjaan.
7. Pilih Jalankan mulai pekerjaan, dan tinjau status pekerjaan di bawah tab Jalankan.

Menggunakan AWS CLI

Prosedur berikut menunjukkan cara menggunakan AWS CLI untuk membuat pekerjaan ETL yang menanyakan tabel S3 Anda. Untuk menggunakan perintah ganti *placeholder values* dengan perintah Anda sendiri.

Prasyarat

- Prasyarat
- Buat skrip untuk terhubung ke bucket tabeldan unggah ke ember S3.

1. Buat AWS Glue pekerjaan.

```
aws glue create-job \
--name etl-tables-job \
--role arn:aws:iam::111122223333:role/AWSGlueServiceRole \
--command '{
    "Name": "glueetl",
    "ScriptLocation": "s3://amzn-s3-demo-bucket1/scripts/glue-etl-query.py",
    "PythonVersion": "3"
}' \
--default-arguments '{
```

```
    "--job-language": "python",
    "--class": "GlueApp"
}' \
--glue-version "5.0"
```

> **Note**
>
> (Opsional) Jika Anda menggunakan Katalog Tabel Amazon S3 untuk metode Apache Iceberg akses, tambahkan katalog klien JAR ke `--default-arguments` menggunakan parameter. `--extra-jars` Ganti *`input placeholders`* dengan milik Anda sendiri saat Anda menambahkan parameter.
>
> ```
> "--extra-jars": "s3://amzn-s3-demo-bucket/jar-path/s3-tables-catalog-for-iceberg-runtime-0.1.5.jar"
> ```

2. Mulai pekerjaan Anda.

```
aws glue start-job-run \
--job-name etl-tables-job
```

3. Untuk meninjau status pekerjaan Anda, salin ID run dari perintah sebelumnya dan masukkan ke dalam perintah berikut.

```
aws glue get-job-run --job-name etl-tables-job \
--run-id jr_ec9a8a302e71f8483060f87b6c309601ea9ee9c1ffc2db56706dfcceb3d0e1ad
```

Tabel S3 Wilayah AWS, titik akhir, dan kuota layanan

Bagian berikut mencakup kuota yang didukung Wilayah AWS dan layanan untuk Tabel S3.

Topik

- Tabel S3 Wilayah AWS dan titik akhir
- Kuota Tabel S3

Tabel S3 Wilayah AWS dan titik akhir

Untuk daftar Tabel Wilayah S3 yang saat ini tersedia di, lihat titik akhir Amazon S3. Untuk terhubung secara terprogram ke AWS layanan, Anda menggunakan titik akhir. Untuk informasi selengkapnya, lihat AWS titik akhir layanan. Untuk informasi titik akhir Amazon S3 selengkapnya, lihat titik akhir Amazon S3.

Nama Wilayah	Wilayah	Titik Akhir	Protokol	Dukungan Versi Khusus
Afrika (Cape Town)	af-south-1	`s3tables.af-south-1.amazonaws.com`	HTTPS	4
Asia Pasifik (Hong Kong)	ap-east-1	`s3tables.ap-east-1.amazonaws.com`	HTTPS	4
Asia Pacific (Tokyo)	ap-northeast-1	`s3tables.ap-northeast-1.amazonaws.com`	HTTPS	4
Asia Pasifik (Seoul)	ap-northeast-2	`s3tables.ap-northeast-2.amazonaws.com`	HTTPS	4
Asia Pasifik (Osaka)	ap-northeast-3	`s3tables.ap-northeast-3.amazonaws.com`	HTTPS	4
Asia Pasifik (Mumbai)	ap-south-1	`s3tables.ap-south-`	HTTPS	4

Nama Wilayah	Wilayah	Titik Akhir	Protokol	Dukungan Versi Khusus
		1.amazonaws.com		
Asia Pasifik (Hyderabad)	ap-south-2	s3tables.ap-south-2.amazonaws.com	HTTPS	4
Asia Pasifik (Singapura)	ap-southeast-1	s3tables.ap-southeast-1.amazonaws.com	HTTPS	4
Asia Pacific (Sydney)	ap-southeast-2	s3tables.ap-southeast-2.amazonaws.com	HTTPS	4
Asia Pasifik (Jakarta)	ap-southeast-3	s3tables.ap-southeast-3.amazonaws.com	HTTPS	4
Asia Pasifik (Melbourne)	ap-southeast-4	s3tables.ap-southeast-4.amazonaws.com	HTTPS	4
Asia Pasifik (Malaysia)	ap-southeast-5	s3tables.ap-southeast-5.amazonaws.com	HTTPS	4

Nama Wilayah	Wilayah	Titik Akhir	Protokol	Dukungan Versi Khusus
Asia Pasifik (Thailand)	ap-tenggara 7	s3tables.ap-southeast-7.amazonaws.com	HTTPS	4
Canada (Central)	ca-central-1	s3tables.ca-central-1.amazonaws.com	HTTPS	4
Kanada Barat (Calgary)	ca-west-1	s3tables.ca-west-1.amazonaws.com	HTTPS	4
Eropa (Frankfurt)	eu-central-1	s3tables.eu-central-1.amazonaws.com	HTTPS	4
Eropa (Zürich)	eu-central-2	s3tables.eu-central-2.amazonaws.com	HTTPS	4
Eropa (Stockholm)	eu-north-1	s3tables.eu-north-1.amazonaws.com	HTTPS	4
Eropa (Milan)	eu-south-1	s3tables.eu-south-1.amazonaws.com	HTTPS	4

Nama Wilayah	Wilayah	Titik Akhir	Protokol	Dukungan Versi Khusus
Eropa (Spanyol)	eu-south-2	s3tables.eu-south-2.amazonaws.com	HTTPS	4
Eropa (Irlandia)	eu-west-1	s3tables.eu-west-1.amazonaws.com	HTTPS	4
Europe (London)	eu-west-2	s3tables.eu-west-2.amazonaws.com	HTTPS	4
Eropa (Paris)	eu-west-3	s3tables.eu-west-3.amazonaws.com	HTTPS	4
Israel (Tel Aviv)	il-central-1	s3tables.il-central-1.amazonaws.com	HTTPS	4
Timur Tengah (UAE)	me-central-1	s3tables.me-central-1.amazonaws.com	HTTPS	4
Timur Tengah (Bahrain)	me-south-1	s3tables.me-south-1.amazonaws.com	HTTPS	4

Nama Wilayah	Wilayah	Titik Akhir	Protokol	Dukungan Versi Khusus
Meksiko (Tengah)	mx-pusat-1	`s3tables.mx-central-1.amazonaws.com`	HTTPS	4
Amerika Selatan (Sao Paulo)	sa-east-1	`s3tables.sa-east-1.amazonaws.com`	HTTPS	4
AS Timur (Virginia Utara)	us-east-1	`s3tables.us-east-1.amazonaws.com`	HTTPS	4
AS Timur (Ohio)	us–east-2	`s3tables.us-east-2.amazonaws.com`	HTTPS	4
AS Barat (California Utara)	us-west-1	`s3tables.us-west-1.amazonaws.com`	HTTPS	4
US West (Oregon)	us-west-2	`s3tables.us-west-2.amazonaws.com`	HTTPS	4

Kuota Tabel S3

Kuota, juga disebut sebagai batas, adalah jumlah maksimum sumber daya layanan atau operasi untuk Anda Akun AWS. Berikut ini adalah kuota untuk sumber daya Tabel S3. Untuk informasi kuota Amazon S3 selengkapnya, lihat kuota Amazon S3.

Nama	Default	Dapat disesuaikan	Deskripsi
Ember Meja	10	Untuk meminta kenaikan kuota, hubungiDukungan.	Jumlah bucket tabel Amazon S3 yang dapat Anda buat per akun Wilayah AWS .
Namespace	10.000	Untuk meminta kenaikan kuota, hubungiDukungan.	Jumlah ruang nama tabel Amazon S3 yang dapat Anda buat per bucket tabel.
Tabel	10.000	Untuk meminta kenaikan kuota, hubungiDukungan.	Jumlah tabel Amazon S3 yang dapat Anda buat per ember meja.

Keamanan untuk Tabel S3

Amazon S3 menyediakan beragam fitur dan alat keamanan. Berikut ini adalah daftar fitur dan alat yang didukung oleh Tabel S3. Aplikasi yang tepat dari alat-alat ini dapat membantu memastikan bahwa sumber daya Anda dilindungi dan hanya dapat diakses oleh pengguna yang dituju.

Kebijakan berbasis identitas

Kebijakan berbasis identitas terlampir pada pengguna, grup, atau peran IAM. Anda dapat menggunakan kebijakan berbasis identitas untuk memberikan akses identitas IAM ke bucket atau tabel tabel Anda. Secara default, pengguna dan peran tidak memiliki izin untuk membuat dan memodifikasi tabel dan bucket tabel. Mereka juga tidak dapat melakukan tugas dengan menggunakan konsol S3, AWS CLI, atau Amazon APIs S3 REST. Anda dapat membuat pengguna, grup, dan peran IAM di akun Anda dan melampirkan kebijakan akses kepada mereka. Anda kemudian dapat memberikan akses ke sumber daya Anda. Untuk membuat dan mengakses bucket

dan tabel tabel, administrator IAM harus memberikan izin yang diperlukan untuk peran AWS Identity and Access Management (IAM) atau pengguna. Untuk informasi selengkapnya, lihat Manajemen akses untuk Tabel S3.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya dilampirkan pada sumber daya. Anda dapat membuat kebijakan berbasis sumber daya untuk bucket dan tabel tabel. Anda dapat menggunakan kebijakan bucket tabel untuk mengontrol bucket tabel dan izin akses API tingkat namespace. Anda juga dapat menggunakan kebijakan bucket tabel untuk mengontrol izin API tingkat tabel pada beberapa tabel dalam bucket. Bergantung pada definisi kebijakan, izin yang dilampirkan ke bucket dapat diterapkan ke semua atau tabel tertentu di bucket. Anda juga dapat menggunakan kebijakan tabel untuk memberikan izin akses API tingkat tabel ke tabel individual di bucket.

Ketika Tabel S3 menerima permintaan untuk melakukan operasi bucket tabel atau operasi tabel, pertama kali memverifikasi bahwa pemohon memiliki izin yang diperlukan. Ini mengevaluasi semua kebijakan akses yang relevan, kebijakan pengguna, dan kebijakan berbasis sumber daya dalam memutuskan apakah akan mengotorisasi permintaan (kebijakan pengguna IAM, kebijakan peran IAM, kebijakan keranjang tabel, dan kebijakan tabel). Dengan kebijakan keranjang tabel dan kebijakan tabel, Anda dapat mempersonalisasi akses ke sumber daya untuk memastikan bahwa hanya identitas yang telah Anda setujui yang dapat mengakses sumber daya dan melakukan tindakan terhadapnya. Untuk informasi selengkapnya, lihat Manajemen akses untuk Tabel S3.

AWS Organizations kebijakan kontrol layanan (SCPs) untuk Tabel S3.

Anda dapat menggunakan Tabel Amazon S3 di Kebijakan Kontrol Layanan (SCPs) untuk mengelola izin kepada pengguna di organisasi Anda. Mirip dengan IAM dan kebijakan sumber daya, semua tindakan tingkat tabel dan bucket direferensikan sebagai bagian dari `s3tables` namespace dalam kebijakan. Untuk informasi selengkapnya, lihat Kebijakan kontrol layanan (SCPs) di Panduan AWS Organizations Pengguna.

Topik

- Melindungi data tabel S3 dengan enkripsi
- Manajemen akses untuk Tabel S3
- Konektivitas VPC untuk Tabel S3
- Pertimbangan dan batasan keamanan untuk Tabel S3

Melindungi data tabel S3 dengan enkripsi

Menggunakan enkripsi sisi server dengan AWS KMS kunci (SSE-KMS) di bucket tabel

Topik

- Cara kerja SSE-KMS untuk tabel dan ember meja
- Menegakkan dan melingkupi penggunaan SSE-KMS untuk tabel dan bucket meja
- Pemantauan dan Audit enkripsi SSE-KMS untuk tabel dan bucket tabel
- Persyaratan izin untuk enkripsi SSE-KMS Tabel S3
- Menentukan enkripsi sisi server dengan AWS KMS kunci (SSE-KMS) di bucket tabel

Bucket tabel memiliki konfigurasi enkripsi default yang secara otomatis mengenkripsi tabel dengan menggunakan enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3). Enkripsi ini berlaku untuk semua tabel di bucket tabel S3 Anda, dan tidak dikenakan biaya kepada Anda.

Jika Anda memerlukan kontrol lebih besar atas kunci enkripsi, seperti mengelola rotasi kunci dan hibah kebijakan akses, Anda dapat mengonfigurasi bucket tabel untuk menggunakan enkripsi sisi server dengan kunci AWS Key Management Service (AWS KMS) (SSE-KMS). Kontrol keamanan AWS KMS dapat membantu Anda memenuhi persyaratan kepatuhan terkait enkripsi. Untuk informasi lebih lanjut tentang SSE-KMS, lihat Menggunakan enkripsi sisi server dengan AWS KMS kunci (SSE-KMS).

Cara kerja SSE-KMS untuk tabel dan ember meja

SSE-KMS dengan ember meja berbeda dari SSE-KMS dalam ember tujuan umum dengan cara berikut:

- Anda dapat menentukan pengaturan enkripsi untuk bucket tabel dan tabel individual.
- Anda hanya dapat menggunakan kunci yang dikelola pelanggan dengan SSE-KMS. AWS kunci terkelola tidak didukung.
- Anda harus memberikan izin untuk peran dan prinsip AWS layanan tertentu untuk mengakses kunci Anda. AWS KMS Untuk informasi selengkapnya, lihat Persyaratan izin untuk enkripsi SSE-KMS Tabel S3. Ini termasuk pemberian akses ke:
  - Prinsip pemeliharaan S3 — untuk melakukan pemeliharaan meja pada tabel terenkripsi
  - Peran integrasi Tabel S3 Anda — untuk bekerja dengan tabel terenkripsi dalam layanan analitik AWS

  - Peran akses klien Anda — untuk akses langsung ke tabel terenkripsi dari klien Apache Iceberg
  - Prinsipal Metadata S3 — untuk memperbarui tabel metadata S3 terenkripsi
- Tabel terenkripsi menggunakan kunci tingkat tabel yang meminimalkan jumlah permintaan yang dibuat AWS KMS untuk membuat bekerja dengan tabel terenkripsi SSE-KMS lebih hemat biaya.

Enkripsi SSE-KMS untuk bucket tabel

Saat membuat bucket tabel, Anda dapat memilih SSE-KMS sebagai jenis enkripsi default dan memilih kunci KMS tertentu yang akan digunakan untuk enkripsi. Setiap tabel yang dibuat dalam bucket itu akan secara otomatis mewarisi pengaturan enkripsi ini dari keranjang tabelnya. Anda dapat menggunakan AWS CLI, S3 API, atau AWS SDKs untuk memodifikasi atau menghapus pengaturan enkripsi default pada keranjang tabel kapan saja. Saat Anda mengubah setelan enkripsi pada keranjang tabel, pengaturan tersebut hanya berlaku untuk tabel baru yang dibuat di bucket tersebut. Pengaturan enkripsi untuk tabel yang sudah ada sebelumnya tidak diubah. Untuk informasi selengkapnya, lihat Menentukan enkripsi untuk bucket tabel.

Enkripsi SSE-KMS untuk tabel

Anda juga memiliki opsi untuk mengenkripsi tabel individual dengan kunci KMS yang berbeda terlepas dari konfigurasi enkripsi default bucket. Untuk mengatur enkripsi untuk tabel individual, Anda harus menentukan kunci enkripsi yang diinginkan pada saat pembuatan tabel. Jika Anda ingin mengubah enkripsi untuk tabel yang ada, maka Anda harus membuat tabel dengan kunci yang diinginkan dan menyalin data dari tabel lama ke yang baru. Untuk informasi selengkapnya, lihat Menentukan enkripsi untuk tabel.

Saat menggunakan AWS KMS enkripsi, Tabel S3 secara otomatis membuat kunci data tingkat tabel unik yang mengenkripsi objek baru yang terkait dengan setiap tabel. Kunci ini digunakan untuk jangka waktu terbatas, meminimalkan kebutuhan akan AWS KMS permintaan tambahan selama operasi enkripsi dan mengurangi biaya enkripsi. Ini mirip dengan Kunci Bucket S3 untuk SSE-KMS.

Menegakkan dan melingkupi penggunaan SSE-KMS untuk tabel dan bucket meja

Anda dapat menggunakan kebijakan berbasis sumber daya Tabel S3, kebijakan kunci KMS, kebijakan berbasis identitas IAM, atau kombinasi lainnya, untuk menerapkan penggunaan SSE-KMS untuk tabel S3 dan bucket tabel. Untuk informasi selengkapnya tentang identitas dan kebijakan sumber daya untuk tabel, lihat Manajemen akses untuk Tabel S3. Untuk informasi tentang menulis kebijakan utama, lihat Kebijakan utama di Panduan AWS Key Management Service Pengembang.

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan kebijakan untuk menegakkan SSE-KMS.

Menegakkan penggunaan SSE-KMS untuk semua tabel dengan kebijakan bucket tabel

Ini adalah contoh kebijakan bucket tabel yang mencegah pengguna membuat tabel di keranjang tabel tertentu kecuali mereka mengenkripsi tabel dengan AWS KMS kunci tertentu. Untuk menggunakan kebijakan ini, ganti *user input placeholders* dengan informasi Anda sendiri:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EnforceKMSEncryption",
      "Effect": "Deny",
      "Principal": "*",
      "Action": [
        "s3tables:CreateTable"
      ],
      "Resource": [
        "<table-bucket-arn>/*"
      ],
      "Condition": {
        "StringNotEquals": {
          "s3tables:sseAlgorithm": "aws:kms",
          "s3tables:kmsKeyArn": "<kms-key-arn>"
        }
      }
    }
  ]
}
```

Mewajibkan pengguna untuk menggunakan enkripsi SSE-KMS dengan kebijakan IAM

Kebijakan identitas IAM ini mengharuskan pengguna untuk menggunakan AWS KMS kunci khusus untuk enkripsi saat membuat atau mengonfigurasi sumber daya Tabel S3. Untuk menggunakan kebijakan ini, ganti *user input placeholders* dengan informasi Anda sendiri:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
      "Sid": "RequireKMSKeyOnTables",
      "Action": [
          "s3tables:CreateTableBucket",
          "s3tables:PutTableBucketEncryption",
          "s3tables:CreateTable"
      ]
      "Effect": "Deny",
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
            "s3tables:sseAlgorithm": "aws:kms",
            "s3tables:kmsKeyArn": "<key_arn>"
        }
      }
    }
  ]
}
```

Membatasi penggunaan kunci ke keranjang tabel tertentu dengan kebijakan kunci KMS

Contoh kebijakan kunci KMS ini memungkinkan kunci untuk digunakan oleh pengguna tertentu hanya untuk operasi enkripsi dalam keranjang tabel tertentu. Jenis kebijakan ini berguna untuk membatasi akses ke kunci dalam skenario lintas akun. Untuk menggunakan kebijakan ini, ganti *`user input placeholders`* dengan informasi Anda sendiri:

```
{
  "Version": "2012-10-17",
  "Id": "Id",
  "Statement": [
    {
      "Sid": "AllowPermissionsToKMS",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      },
      "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "kms:EncryptionContext:aws:s3:arn": "<table-bucket-arn>/*"
```

```
        }
      }
    }
  ]
}
```

Pemantauan dan Audit enkripsi SSE-KMS untuk tabel dan bucket tabel

Untuk mengaudit penggunaan AWS KMS kunci Anda untuk data terenkripsi SSE-KMS Anda, Anda dapat menggunakan log. AWS CloudTrail Anda bisa mendapatkan wawasan tentang operasi kriptografi Anda, seperti `GenerateDataKey` dan`Decrypt`. CloudTrail mendukung banyak nilai atribut untuk memfilter pencarian Anda, termasuk nama acara, nama pengguna, dan sumber acara.

Anda dapat melacak permintaan konfigurasi enkripsi untuk tabel Amazon S3 dan bucket tabel dengan menggunakan peristiwa. CloudTrail Nama peristiwa API berikut digunakan dalam CloudTrail log:

- `s3tables:PutTableBucketEncryption`
- `s3tables:GetTableBucketEncryption`
- `s3tables:DeleteTableBucketEncryption`
- `s3tables:GetTableEncryption`
- `s3tables:CreateTable`
- `s3tables:CreateTableBucket`

> **Note**
>
> EventBridge tidak didukung untuk ember meja.

Persyaratan izin untuk enkripsi SSE-KMS Tabel S3

Saat Anda menggunakan enkripsi sisi server dengan kunci AWS Key Management Service (AWS KMS) (SSE-KMS) untuk tabel di bucket tabel S3, Anda harus memberikan izin untuk identitas yang berbeda di akun Anda. Minimal identitas akses Anda dan prinsip pemeliharaan Tabel S3 memerlukan akses ke kunci Anda, izin lain yang diperlukan bergantung pada kasus penggunaan Anda.

Izin yang Diperlukan

Untuk mengakses tabel yang dienkripsi dengan kunci KMS, Anda memerlukan izin ini pada kunci itu:

- `kms:GenerateDataKey`
- `kms:Decrypt`

> **Important**
>
> Untuk menggunakan SSE-KMS pada tabel, prinsipal (`maintenance.s3tables.amazonaws.com`) layanan pemeliharaan Tabel Amazon S3 membutuhkan `kms:GenerateDataKey` dan `kms:Decrypt` izin pada kunci.

Izin tambahan

Izin tambahan ini diperlukan tergantung pada kasus penggunaan Anda:

- Izin untuk integrasi layanan AWS analitik — Jika Anda bekerja dengan tabel terenkripsi SSE-KMS dalam layanan AWS analitik, peran integrasi Anda memerlukan izin untuk menggunakan kunci KMS Anda.
- Izin untuk akses langsung - Jika Anda bekerja dengan tabel terenkripsi SSE-KMS secara langsung, melalui metode seperti titik akhir Tabel Amazon S3 Iceberg REST atau Katalog Tabel Amazon S3, Anda harus memberikan peran IAM yang digunakan Apache Iceberg klien Anda mengakses kunci Anda.
- Izin untuk tabel Metadata S3 — Jika Anda menggunakan enkripsi SSE-KMS untuk tabel Metadata S3, Anda harus memberikan akses utama layanan S3 Metadata () ke kunci KMS Anda. `metadata.s3.amazonaws.com` Hal ini memungkinkan Metadata S3 untuk memperbarui tabel terenkripsi sehingga mereka akan mencerminkan perubahan data terbaru Anda.

> **Note**
>
> Untuk kunci KMS lintas akun, peran IAM Anda memerlukan izin akses kunci dan otorisasi eksplisit dalam kebijakan kunci. Untuk informasi selengkapnya tentang izin lintas akun untuk kunci KMS, lihat Mengizinkan AWS akun eksternal menggunakan kunci KMS di Panduan Pengembang AWS Key Management Service Layanan.

Topik

- Memberikan izin utama layanan pemeliharaan Tabel S3 ke kunci KMS Anda
- Memberikan izin kepada prinsipal IAM untuk bekerja dengan tabel terenkripsi dalam layanan analitik terintegrasi AWS
- Memberikan izin kepada prinsipal IAM untuk bekerja dengan tabel terenkripsi secara langsung
- Memberikan izin utama layanan Metadata S3 untuk menggunakan kunci KMS Anda

Memberikan izin utama layanan pemeliharaan Tabel S3 ke kunci KMS Anda

Izin ini diperlukan untuk membuat tabel terenkripsi SSE-KMS dan untuk memungkinkan pemeliharaan tabel otomatis seperti pemadatan, manajemen snapshot, dan penghapusan file yang tidak direferensikan pada tabel terenkripsi.

> **Note**
>
> Setiap kali Anda membuat permintaan untuk membuat tabel terenkripsi SSE-KMS, Tabel S3 memeriksa untuk memastikan `maintenance.s3tables.amazonaws.com` kepala sekolah memiliki akses ke kunci KMS Anda. Untuk melakukan pemeriksaan ini, objek zero-byte dibuat sementara di bucket tabel Anda, objek ini akan secara otomatis dihapus oleh operasi pemeliharaan penghapusan file yang tidak direferensikan. Jika kunci KMS yang Anda tentukan untuk enkripsi tidak memiliki akses pemeliharaan, operasi CreateTable akan gagal.

Untuk memberikan akses pemeliharaan pada tabel terenkripsi SSE-KMS, Anda dapat menggunakan contoh kebijakan kunci berikut. Dalam kebijakan ini, kepala `maintenance.s3tables.amazonaws.com` layanan diberikan izin untuk menggunakan kunci KMS tertentu untuk mengenkripsi dan mendekripsi tabel dalam keranjang tabel tertentu. Untuk menggunakan kebijakan ini, ganti *`user input placeholders`* dengan informasi Anda sendiri:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EnableKeyUsage",
            "Effect": "Allow",
            "Principal": {
                "Service": "maintenance.s3tables.amazonaws.com"
```

```
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "<kms-key-arn>",
            "Condition": {
                "StringLike": {
                    "kms:EncryptionContext:aws:s3:arn":"<table-or-table-bucket-arn>/*"
                }
            }
        }
    ]
}
```

Memberikan izin kepada prinsipal IAM untuk bekerja dengan tabel terenkripsi dalam layanan analitik terintegrasi AWS

Untuk bekerja dengan tabel S3 dalam layanan AWS analitik, Anda mengintegrasikan bucket tabel Anda dengan Amazon SageMaker Lakehouse. Integrasi ini memungkinkan layanan AWS analitik untuk secara otomatis menemukan dan mengakses data tabel. Untuk informasi lebih lanjut tentang integrasi, lihatMenggunakan Tabel Amazon S3 dengan layanan analitik AWS.

Saat Anda bekerja dengan tabel terenkripsi SSE-KMS di layanan tersebut, peran yang Anda gunakan harus memiliki izin untuk menggunakan kunci Anda AWS KMS untuk operasi enkripsi. Anda dapat menerapkan izin ini ke `S3TablesRoleForLakeFormation` peran yang dibuat selama integrasi, atau ke peran IAM Anda sendiri.

Contoh kebijakan IAM inline berikut dapat digunakan untuk memberikan izin peran `S3TablesRoleForLakeFormation` layanan untuk menggunakan kunci KMS tertentu di akun Anda untuk operasi enkripsi. Untuk menggunakan kebijakan, ganti *`input placeholder values`* dengan kebijakan Anda sendiri.

```
{
  "Sid": "AllowTableRoleAccess",
  "Effect": "Allow",
  "Principal": {
 "AWS": "arn:aws:iam::111122223333:role/service-role/S3TablesRoleForLakeFormation"
  },
  "Action": [
      "kms:GenerateDataKey",
```

```
        "kms:Decrypt"
    ],
    "Resource": "<kms-key-arn>"
}
```

Memberikan izin kepada prinsipal IAM untuk bekerja dengan tabel terenkripsi secara langsung

Ketika Anda bekerja dengan tabel terenkripsi menggunakan pihak ketiga atau metode akses langsung, Anda harus memberikan peran yang Anda gunakan akses ke kunci KMS Anda. Contoh berikut menunjukkan cara memberikan akses melalui kebijakan IAM atau kebijakan kunci KMS.

IAM policy

Lampirkan kebijakan inline ini ke peran IAM Anda untuk mengizinkan akses kunci KMS. Untuk menggunakan kebijakan ini ganti *input placeholder values* dengan kunci KMS Anda sendiri.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
        "Effect": "Allow",
        "Action": [
            "kms:Decrypt",
            "kms:GenerateDataKey"
         ],
        "Resource": "<kms-key-arn>"
        }
    ]
}
```

KMS key policy

Lampirkan kebijakan inline ini ke kunci KMS untuk memungkinkan AWS KMS peran yang ditentukan menggunakan kunci tersebut. Untuk menggunakan kebijakan ini ganti *input placeholder values* dengan peran IAM Anda.

```
{
    "Sid": "Allow use of the key",
    "Effect": "Allow",
    "Principal": {
```

```
        "AWS": [
            "arn:aws:iam::<catalog-account-id>:role/<role-name>"
        ]
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey",
    ],
    "Resource": "*"
}
```

Memberikan izin utama layanan Metadata S3 untuk menggunakan kunci KMS Anda

Untuk mengizinkan Amazon S3 memperbarui tabel metadata terenkripsi SSE-KMS, dan melakukan pemeliharaan pada tabel metadata tersebut, Anda dapat menggunakan kebijakan kunci contoh berikut. Dalam kebijakan ini, Anda mengizinkan kepala sekolah `metadata.s3.amazonaws.com` dan `maintenance.s3tables.amazonaws.com` layanan mengenkripsi dan mendekripsi tabel dalam keranjang tabel tertentu menggunakan kunci tertentu. Untuk menggunakan kebijakan ini, ganti *`user input placeholders`* dengan informasi Anda sendiri:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EnableKeyUsage",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "maintenance.s3tables.amazonaws.com",
                    "metadata.s3.amazonaws.com"
                ]
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "<kms-key-arn>",
            "Condition": {
                "StringLike": {
                    "kms:EncryptionContext:aws:s3:arn":"<table-or-table-bucket-arn>/*"
                }
            }
```

```
        }
    ]
}
```

Menentukan enkripsi sisi server dengan AWS KMS kunci (SSE-KMS) di bucket tabel

Semua bucket tabel Amazon S3 memiliki enkripsi yang dikonfigurasi secara default, dan semua tabel baru yang dibuat dalam keranjang tabel secara otomatis dienkripsi saat istirahat. Enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3) adalah konfigurasi enkripsi default untuk setiap bucket tabel. Jika Anda ingin menentukan jenis enkripsi yang berbeda, Anda dapat menggunakan enkripsi sisi server dengan AWS Key Management Service (AWS KMS) kunci (SSE-KMS).

Anda dapat menentukan enkripsi SSE-KMS dalam `CreateTable` permintaan `CreateTableBucket` atau Anda, atau Anda dapat mengatur konfigurasi enkripsi default di keranjang tabel dalam permintaan. `PutTableBucketEncryption`

Important

Untuk mengizinkan pemeliharaan otomatis pada tabel terenkripsi SSE-KMS dan bucket tabel, Anda harus memberikan izin utama layanan maintenance.s3tables.amazonaws.com untuk menggunakan kunci KMS Anda. Untuk informasi selengkapnya, lihat Persyaratan izin untuk enkripsi SSE-KMS Tabel S3.

Menentukan enkripsi untuk bucket tabel

Anda dapat menentukan SSE-KMS sebagai jenis enkripsi default saat membuat bucket tabel baru, misalnya, lihat. Membuat ember meja Setelah membuat bucket tabel, Anda dapat menentukan penggunaan SSE-KMS sebagai pengaturan enkripsi default menggunakan operasi REST API AWS SDKs, dan (). AWS Command Line Interface AWS CLI

Note

Saat Anda menentukan SSE-KMS sebagai tipe enkripsi default, kunci yang Anda gunakan untuk enkripsi harus mengizinkan akses ke prinsipal layanan pemeliharaan Tabel S3. Jika kepala layanan pemeliharaan tidak memiliki akses, Anda tidak akan dapat membuat tabel di keranjang meja itu. Untuk informasi selengkapnya, lihat Memberikan izin utama layanan pemeliharaan Tabel S3 ke kunci KMS Anda .

Menggunakan AWS CLI

Untuk menggunakan AWS CLI perintah contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3tables put-table-bucket-encryption \
    --table-bucket-arn arn:aws:s3tables:us-east-1:111122223333:bucket/amzn-s3-demo-table-bucket; \
    --encryption-configuration '{
        "sseAlgorithm": "aws:kms",
        "kmsKeyArn": "arn:aws:kms:us-east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }' \
    --region us-east-1
```

Anda dapat menghapus setelan enkripsi default untuk bucket tabel menggunakan operasi DeleteTableBucketEncryptionAPI. Saat Anda menghapus pengaturan enkripsi, tabel baru yang dibuat di bucket tabel akan menggunakan enkripsi SSE-S3 default.

Menentukan enkripsi untuk tabel

Anda dapat menerapkan enkripsi SSE-KMS ke tabel baru saat Anda membuatnya menggunakan mesin kueri, operasi REST API AWS SDKs, dan (). AWS Command Line Interface AWS CLI Pengaturan enkripsi yang Anda tentukan saat membuat tabel lebih diutamakan daripada setelan enkripsi default bucket tabel.

> **Note**
>
> Saat Anda menggunakan enkripsi SSE-KMS untuk tabel, kunci yang Anda gunakan untuk enkripsi harus mengizinkan prinsipal layanan pemeliharaan Tabel S3 untuk mengaksesnya. Jika kepala layanan pemeliharaan tidak memiliki akses, Anda tidak akan dapat membuat tabel. Untuk informasi selengkapnya, lihat Memberikan izin utama layanan pemeliharaan Tabel S3 ke kunci KMS Anda .

Izin yang diperlukan

Izin berikut diperlukan untuk membuat tabel terenkripsi

- `s3tables:CreateTable`
- `s3tables:PutTableEncryption`

Menggunakan AWS CLI

AWS CLI Contoh berikut membuat tabel baru dengan skema dasar, dan mengenkripsi dengan kunci yang dikelola pelanggan. AWS KMS Untuk menggunakan perintah, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3tables create-table \
  --table-bucket-arn "arn:aws:s3tables:Region:ownerAccountId:bucket/amzn-s3-demo-table-bucket" \
  --namespace "mydataset" \
  --name "orders" \
  --format "ICEBERG" \
  --encryption-configuration '{
    "sseAlgorithm": "aws:kms",
    "kmsKeyArn":
 "arn:aws:kms:Region:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  }' \
  --metadata '{
    "iceberg": {
      "schema": {
        "fields": [
          {
            "name": "order_id",
            "type": "string",
            "required": true
          },
          {
            "name": "order_date",
            "type": "timestamp",
            "required": true
          },
          {
            "name": "total_amount",
            "type": "decimal(10,2)",
            "required": true
          }
        ]
      }
    }
```

```
}'
```

Perlindungan data mengacu pada melindungi data saat dalam perjalanan (saat bepergian ke dan dari Amazon S3) dan saat istirahat (saat disimpan di disk di pusat data Amazon S3). Tabel S3 selalu melindungi data dalam perjalanan menggunakan Transport Layer Security (1.2 ke atas) melalui HTTPS. Untuk melindungi data saat istirahat di bucket tabel S3, Anda memiliki opsi berikut:

Enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3)

Semua bucket tabel Amazon S3 memiliki enkripsi yang dikonfigurasi secara default. Opsi default untuk enkripsi di sisi server adalah dengan kunci terkelola Amazon S3 (SSE-S3). Enkripsi ini tidak dikenakan biaya kepada Anda dan berlaku untuk semua tabel di bucket tabel S3 Anda kecuali Anda menentukan bentuk enkripsi lain. Setiap objek dienkripsi dengan kunci unik. Sebagai pelindung tambahan, SSE-S3 mengenkripsi kunci itu sendiri dengan kunci root yang secara rutin diputar. SSE-S3 menggunakan salah satu penyandian blok terkuat yang ada, Advanced Encryption Standard 256-bit (AES-256), untuk mengenkripsi data Anda.

Enkripsi sisi server dengan AWS KMS kunci (SSE-KMS)

Anda dapat memilih untuk mengonfigurasi bucket tabel atau tabel untuk menggunakan enkripsi sisi server dengan AWS Key Management Service (AWS KMS) kunci (SSE-KMS). Kontrol keamanan AWS KMS dapat membantu Anda memenuhi persyaratan kepatuhan terkait enkripsi. SSE-KMS memberi Anda lebih banyak kontrol atas kunci enkripsi Anda dengan memungkinkan Anda melakukan hal berikut:

- Membuat, melihat, mengedit, memantau, mengaktifkan atau menonaktifkan, memutar, dan menjadwalkan penghapusan tombol KMS.
- Menentukan kebijakan yang mengendalikan bagaimana dan oleh siapa kunci KMS digunakan.
- Lacak penggunaan kunci AWS CloudTrail untuk memverifikasi kunci KMS Anda digunakan dengan benar.

Tabel S3 mendukung penggunaan kunci yang dikelola pelanggan di SSE-KMS untuk mengenkripsi tabel. AWS kunci terkelola tidak didukung. Untuk informasi lebih lanjut tentang penggunaan SSE-KMS untuk tabel S3 dan bucket tabel, lihat. Menggunakan enkripsi sisi server dengan AWS KMS kunci (SSE-KMS) di bucket tabel

Manajemen akses untuk Tabel S3

Di S3 Tabel sumber daya termasuk bucket tabel dan tabel yang dikandungnya. Pengguna root Akun AWS yang menciptakan sumber daya (pemilik sumber daya) dan AWS Identity and Access Management (IAM) pengguna dalam akun yang memiliki izin yang diperlukan dapat mengakses sumber daya yang mereka buat. Pemilik sumber daya menentukan siapa lagi yang dapat mengakses sumber daya dan tindakan yang diizinkan untuk mereka lakukan pada sumber daya. Amazon S3 memiliki berbagai alat manajemen akses yang dapat Anda gunakan untuk memberi orang lain akses ke sumber daya S3 Anda. Topik berikut memberi Anda ikhtisar sumber daya, tindakan IAM, dan kunci kondisi untuk Tabel S3. Mereka juga memberikan contoh untuk kebijakan berbasis sumber daya dan berbasis identitas untuk Tabel S3.

Topik

- Sumber daya
- Tindakan untuk Tabel S3
- Kunci kondisi untuk Tabel S3
- Kebijakan berbasis identitas IAM untuk Tabel S3
- Kebijakan berbasis sumber daya untuk Tabel S3
- AWS kebijakan terkelola untuk Tabel S3
- Memberikan akses dengan semantik SQL

Sumber daya

Sumber daya Tabel S3 termasuk bucket tabel dan tabel yang dikandungnya.

- Bucket tabel — Bucket tabel dirancang khusus untuk tabel dan penyedia transaksi per detik (TPS) yang lebih tinggi dan throughput kueri yang lebih baik dibandingkan dengan tabel yang dikelola sendiri dalam bucket S3 tujuan umum. Bucket meja memberikan daya tahan, ketersediaan, skalabilitas, dan karakteristik kinerja yang sama dengan bucket tujuan umum Amazon S3.
- Tabel — Tabel di bucket tabel Anda disimpan dalam Apache Iceberg format. Anda dapat menanyakan tabel ini menggunakan SQL standar di mesin kueri yang mendukungIceberg.

Amazon Resource Names (ARNs) untuk tabel dan bucket tabel berisi `s3tables` namespace, Akun AWS ID, dan nama bucket. Wilayah AWS Untuk mengakses dan melakukan tindakan pada tabel dan bucket tabel Anda, Anda harus menggunakan format ARN berikut:

- Tabel format ARN:

```
arn:aws:s3tables:us-west-2:111122223333:bucket/amzn-s3-demo-bucket/
table/demo-tableID
```

Tindakan untuk Tabel S3

Dalam kebijakan berbasis identitas atau kebijakan berbasis sumber daya, Anda menentukan tindakan Tabel S3 mana yang diizinkan atau ditolak untuk prinsip IAM tertentu. Tindakan tabel sesuai dengan operasi API tingkat bucket dan tabel. Semua tindakan adalah bagian dari namespace IAM yang unik:. `s3tables`

Saat Anda menggunakan tindakan dalam kebijakan, Anda biasanya mengizinkan atau menolak akses ke operasi API dengan nama yang sama. Namun, dalam beberapa kasus, satu tindakan mengendalikan akses ke lebih dari satu operasi API. Misalnya, `s3tables:GetTableData` tindakan mencakup izin untuk`GetObject`,`ListParts`, dan operasi `ListMultiparts` API.

Berikut ini adalah tindakan yang didukung untuk bucket tabel. Anda dapat menentukan tindakan berikut dalam `Action` elemen kebijakan IAM atau kebijakan sumber daya.

Tindakan	Deskripsi	Tingkat akses	Akses lintas akun
`s3tables:CreateTableBucket`	Memberikan izin untuk membuat keranjang tabel	`Write`	Tidak
`s3tables:GetTableBucket`	Memberikan izin untuk mengambil ARN bucket tabel, nama bucket tabel, dan tanggal pembuatan.	`Write`	Ya

Tindakan	Deskripsi	Tingkat akses	Akses lintas akun
`s3tables:ListTableBuckets`	Memberikan izin untuk mencantumkan semua bucket tabel di akun ini.	`Read`	Tidak
`s3tables:CreateNamespace`	Memberikan izin untuk membuat spasi nama di keranjang tabel	`Read`	Ya
`s3tables:GetNamespace`	Memberikan izin untuk mengambil detail namespace	`Read`	Ya
`s3tables:ListNamespaces`	Memberikan izin untuk mencantumkan semua ruang nama di keranjang meja.	`Read`	Ya
`s3tables:DeleteNamespace`	Memberikan izin untuk menghapus namespace di keranjang tabel	`Write`	Ya

Tindakan	Deskripsi	Tingkat akses	Akses lintas akun
`s3tables:DeleteTableBucket`	Memberikan izin untuk menghapus ember	`Write`	Ya
`s3tables:PutTableBucketPolicy`	Memberikan izin untuk menambah atau mengganti kebijakan bucket	`Permissions Management`	Tidak
`s3tables:GetTableBucketPolicy`	Memberikan izin untuk mengambil kebijakan bucket	`Read`	Tidak
`s3tables:DeleteTableBucketPolicy`	Memberikan izin untuk menghapus kebijakan bucket	`Permissions Management`	Tidak
`s3tables:GetTableBucketMaintenanceConfiguration`	Memberikan izin untuk mengambil konfigurasi pemeliharaan untuk bucket meja	`Read`	Ya

Tindakan	Deskripsi	Tingkat akses	Akses lintas akun
`s3tables:PutTableBucketMaintenanceConfiguration`	Memberikan izin untuk menambah atau mengganti konfigurasi pemeliharaan untuk bucket meja	`Write`	Ya
`s3tables:PutTableBucketEncryption`	Memberikan izin untuk menambah atau mengganti konfigurasi enkripsi untuk keranjang tabel	`Write`	Tidak
`s3tables:GetTableBucketEncryption`	Memberikan izin untuk mengambil konfigurasi enkripsi untuk keranjang tabel	`Read`	Tidak
`s3tables:DeleteTableBucketEncryption`	Memberikan izin untuk menghapus konfigurasi enkripsi untuk keranjang tabel	`Write`	Tidak

Tindakan berikut didukung untuk tabel:

Tindakan	Deskripsi	Tingkat akses	Akses lintas akun
`s3tables:GetTableMaintenanceConfiguration`	Memberikan izin untuk mengambil konfigurasi pemeliharaan untuk tabel	`Read`	Ya
`s3tables:PutTableMaintenanceConfiguration`	Memberikan izin untuk menambah atau mengganti konfigurasi pemeliharaan untuk tabel	`Write`	Ya
`s3tables:PutTablePolicy`	Memberikan izin untuk menambah atau mengganti kebijakan tabel	`Permissions Management`	Tidak
`s3tables:GetTablePolicy`	Memberikan izin untuk mengambil kebijakan tabel	`Read`	Tidak
`s3tables:DeleteTablePolicy`	Memberikan izin untuk menghapus	`Permissions management`	Tidak

Tindakan	Deskripsi	Tingkat akses	Akses lintas akun
	kebijakan tabel		
`s3tables:CreateTable`	Memberikan izin untuk membuat tabel di ember meja	`Write`	Ya
`s3tables:GetTable`	Memberikan izin untuk mengambil informasi tabel	`Read`	Ya
`s3tables:GetTableMetadataLocation`	Memberikan izin untuk mengambil pointer root tabel (file metadata)	`Read`	Ya
`s3tables:ListTables`	Memberikan izin untuk mencantumkan semua tabel dalam ember meja	`Read`	Ya
`s3tables:RenameTable`	Berikan izin untuk mengubah nama tabel.	`Write`	Ya

Tindakan	Deskripsi	Tingkat akses	Akses lintas akun
`s3tables:UpdateTableMetadataLocation`	Memberikan izin untuk memperbarui penunjuk root tabel (file metadata)	`Write`	Ya
`s3tables:GetTableData`	Memberikan izin untuk membaca metadata tabel dan objek data yang disimpan dalam keranjang tabel	`Read`	Ya
`s3tables:PutTableData`	Memberikan izin untuk menulis metadata tabel dan objek data yang disimpan dalam keranjang tabel	`Write`	Ya

Tindakan	Deskripsi	Tingkat akses	Akses lintas akun
`s3tables:GetTableEncryption`	Memberikan izin untuk mengambil pengaturan enkripsi untuk tabel	`Write`	Tidak
`s3tables:PutTableEncryption`	Memberikan izin untuk menambahkan enkripsi ke tabel	`Write`	Tidak

Untuk melakukan tindakan baca dan tulis tingkat tabel, Tabel S3 mendukung operasi API Amazon S3 seperti dan. `GetObject` `PutObject` Tabel berikut menyediakan daftar tindakan tingkat objek. Saat memberikan izin baca dan tulis ke tabel Anda, Anda menggunakan tindakan berikut.

Tindakan	Objek S3 APIs		
`s3tables:GetTableData`	`GetObject`, `ListParts`, `HeadObject`		
`s3tables:PutTableData`	`PutObject`, `CreateMultipartUpload`, `CompleteMultipartU`		

Tindakan	Objek S3 APIs		
	`pload` , `UploadPart` , `AbortMultipartUpload`		

Misalnya, jika pengguna memiliki `GetTableData` izin, maka mereka dapat membaca semua file yang terkait dengan tabel, seperti file metadata, manifes, file daftar manifes, dan file data parket.

Kunci kondisi untuk Tabel S3

Tabel S3 mendukung kunci konteks kondisi AWS global.

Selain itu, Tabel S3 mendefinisikan kunci kondisi berikut yang dapat Anda gunakan dalam kebijakan akses.

Kunci syarat	Deskripsi	Tipe	
`s3tables:tableName`	Memfilter akses dengan nama tabel di keranjang tabel. Anda dapat menggunakan kunci `s3tables:tableName` kondisi untuk menulis IAM, atau kebijakan bucket	`String`	

Kunci syarat	Deskripsi	Tipe	
	tabel yang membatasi akses pengguna atau aplikasi hanya ke tabel yang memenuhi kondisi nama ini. Penting untuk dicatat bahwa jika Anda menggunakan tombol `s3tables:tableName` kondisi untuk mengontrol akses maka perubahan nama tabel dapat memengaruhi kebijakan ini. Nilai contoh: `"s3tables:tableName":"department*"`		

Kunci syarat	Deskripsi	Tipe	
`s3tables:namespace`	Memfilter akses oleh ruang nama yang dibuat di keranjang tabel. Anda dapat menggunakan kunci `s3tables:namespace` kondisi untuk menulis kebijakan IAM, tabel, atau keranjang tabel yang membatasi akses pengguna atau aplikasi ke tabel yang merupakan bagian dari namespace tertentu. Nilai contoh: `"s3tables:namespace":"hr"` Penting untuk dicatat bahwa	`String`	

Kunci syarat	Deskripsi	Tipe	
	jika Anda menggunakan kunci `s3tables:namespace` kondisi untuk mengontrol akses, maka perubahan dalam ruang nama dapat memengaruhi kebijakan ini.		

Kunci syarat	Deskripsi	Tipe	
`s3tables:SSEAlgorithm`	Memfilter akses oleh algoritma enkripsi sisi server yang digunakan untuk mengenkripsi tabel. Anda dapat menggunakan kunci `s3tables:SSEAlgorithm` kondisi untuk menulis kebijakan IAM, tabel, atau keranjang tabel yang membatasi akses pengguna atau aplikasi ke tabel yang dienkripsi dengan jenis enkripsi tertentu. Nilai contoh: `"s3tables:SSEAlgor`	`String`	

Kunci syarat	Deskripsi	Tipe	
	`ithm":"aws:kms"` Penting untuk dicatat bahwa jika Anda menggunakan kunci `s3tables:SSEAlgorithm` kondisi untuk mengontrol akses, maka perubahan enkripsi dapat memengaruhi kebijakan ini.		

Kunci syarat	Deskripsi	Tipe	
`s3tables:KMSKeyArn`	Memfilter akses dengan AWS KMS kunci ARN untuk kunci yang digunakan untuk mengenkripsi tabel Anda dapat menggunakan kunci `s3tables:KMSKeyArn` kondisi untuk menulis kebijakan IAM, tabel, atau keranjang tabel yang membatasi akses pengguna atau aplikasi ke tabel yang dienkripsi dengan kunci KMS tertentu. Penting untuk dicatat bahwa jika Anda	`ARN`	

Kunci syarat	Deskripsi	Tipe	
	menggunakan tombol `s3tables:KMSKeyArn` kondisi untuk mengontrol akses, maka mengubah kunci KMS Anda dapat memengaruhi kebijakan ini.		

Kebijakan berbasis identitas IAM untuk Tabel S3

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi tabel dan bucket tabel. Mereka juga tidak dapat melakukan tugas dengan menggunakan konsol s3, AWS Command Line Interface (AWS CLI), atau Amazon APIs S3 REST. Untuk membuat dan mengakses bucket dan tabel tabel, administrator AWS Identity and Access Management (IAM) harus memberikan izin yang diperlukan untuk peran IAM atau pengguna. Untuk mempelajari cara membuat kebijakan berbasis identitas IAM menggunakan contoh dokumen kebijakan JSON ini, lihat Membuat kebijakan IAM dalam Panduan Pengguna IAM.

Topik berikut mencakup contoh kebijakan berbasis identitas IAM. Untuk menggunakan contoh kebijakan berikut, ganti *`user input placeholders`* dengan informasi Anda sendiri.

Topik

- Contoh 1: Izinkan akses untuk membuat dan menggunakan bucket tabel
- Contoh 2: Izinkan akses untuk membuat dan menggunakan tabel di keranjang tabel

Contoh 1: Izinkan akses untuk membuat dan menggunakan bucket tabel

.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "AllowBucketActions",
        "Effect": "Allow",
        "Action": [
            "s3tables:CreateTableBucket",
            "s3tables:PutTableBucketPolicy",
            "s3tables:GetTableBucketPolicy",
            "s3tables:ListTableBuckets",
            "s3tables:GetTableBucket"
        ],

        "Resource": "arn:aws:s3tables:region:account_id:bucket/*"
    }]
}
```

Contoh 2: Izinkan akses untuk membuat dan menggunakan tabel di keranjang tabel

```
{
     "Version": "2012-10-17",
     "Statement": [
         {
             "Sid": "AllowBucketActions",
             "Effect": "Allow",
             "Action": [
                 "s3tables:CreateTable",
                 "s3tables:PutTableData",
                 "s3tables:GetTableData",
                 "s3tables:GetTableMetadataLocation",
                 "s3tables:UpdateTableMetadataLocation",
                 "s3tables:GetNamespace",
                 "s3tables:CreateNamespace"
             ],

             "Resource": [
               "arn:aws:s3tables:region:account_id:bucket/amzn-s3-demo-bucket",
               "arn:aws:s3tables:region:account_id:bucket/amzn-s3-demo-bucket/table/*"
            ]
         }
     ]
 }
```

Kebijakan berbasis sumber daya untuk Tabel S3

Tabel S3 menyediakan kebijakan berbasis sumber daya untuk mengelola akses ke bucket dan tabel tabel: kebijakan keranjang tabel dan kebijakan tabel. Anda dapat menggunakan kebijakan bucket tabel untuk memberikan izin akses API di keranjang tabel, namespace, atau tingkat tabel. Izin yang dilampirkan pada bucket tabel dapat diterapkan ke semua tabel di bucket atau tabel tertentu di bucket, tergantung pada definisi kebijakan. Anda dapat menggunakan kebijakan tabel untuk memberikan izin di tingkat tabel.

Ketika Tabel S3 menerima permintaan, pertama-tama memverifikasi bahwa pemohon memiliki izin yang diperlukan. Ini mengevaluasi semua kebijakan akses yang relevan, kebijakan pengguna, dan kebijakan berbasis sumber daya dalam memutuskan apakah akan mengotorisasi permintaan (kebijakan pengguna IAM, kebijakan peran IAM, kebijakan keranjang tabel, dan kebijakan tabel). Misalnya, jika kebijakan bucket tabel memberikan izin kepada pengguna untuk melakukan semua tindakan pada tabel dalam bucket (termasuk`DeleteTable`), tetapi tabel individual memiliki kebijakan tabel yang menolak `DeleteTable` tindakan untuk semua pengguna, maka pengguna tidak dapat menghapus tabel tersebut.

Topik berikut mencakup contoh kebijakan keranjang tabel dan tabel. Untuk menggunakan kebijakan ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

> **Note**
>
> - Setiap kebijakan yang memberikan izin untuk mengubah tabel harus menyertakan izin `GetTableMetadataLocation` untuk mengakses file root tabel. Untuk informasi selengkapnya, lihat `GetTableMetadataLocation`.
> - Setiap kali Anda melakukan aktivitas tulis atau hapus di tabel, sertakan izin `UpdateTableMetadataLocation` dalam kebijakan akses Anda.
> - Sebaiknya gunakan kebijakan bucket tabel untuk mengatur akses ke tindakan tingkat ember dan kebijakan tabel untuk mengatur akses ke tindakan tingkat tabel. Jika Anda ingin menentukan kumpulan izin yang sama di beberapa tabel, sebaiknya gunakan kebijakan bucket tabel.

Topik

- Contoh 1: Kebijakan bucket tabel memungkinkan akses ke PutBucketMaintenanceConfiguration bucket di akun

- Contoh 2: Kebijakan keranjang tabel untuk memungkinkan akses baca (SELECT) ke tabel yang disimpan di hr namespace
- Contoh 3: Kebijakan tabel untuk memungkinkan pengguna menghapus tabel

Contoh 1: Kebijakan bucket tabel memungkinkan akses ke **`PutBucketMaintenanceConfiguration`** bucket di akun

Contoh kebijakan bucket tabel berikut memungkinkan IAM `data steward` menghapus objek yang tidak direferensikan untuk semua bucket di akun dengan mengizinkan akses ke. `PutBucketMaintenanceConfiguration`

```
{
     "Version": "2012-10-17",
     "Statement": [
         {
             "Effect": "Allow",
             "Principal": {
               "AWS": "arn:aws:iam::account_id:role/datasteward"
             },
             "Action": ["s3tables:PutTableBucketMaintenanceConfiguration"],
             "Resource":"arn:aws:s3tables:region:account_id:bucket/*"
     }]
 }
```

Contoh 2: Kebijakan keranjang tabel untuk memungkinkan akses baca (SELECT) ke tabel yang disimpan di **`hr`** namespace

Berikut contoh kebijakan bucket tabel memungkinkan Jane, pengguna dari Akun AWS ID `123456789012` untuk mengakses tabel yang disimpan di `hr` namespace dalam bucket tabel.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/Jane"
            },
            "Action": [
                "s3tables:GetTableData",
```

```
                "s3tables:GetTableMetadataLocation"
            ],
            "Resource": "arn:aws:s3tables:region:account_id:bucket/amzn-s3-demo-table-
bucket/table/*",
            "Condition": {
                "StringLike": {"s3tables:namespace": "hr"}
            }
        }
    ]
}
```

Contoh 3: Kebijakan tabel untuk memungkinkan pengguna menghapus tabel

Berikut contoh kebijakan tabel yang memungkinkan peran IAM `data steward` untuk menghapus tabel.

```
{
     "Version": "2012-10-17",
     "Id": "DeleteTable",
     "Statement": [
      {
       "Effect": "Allow",
       "Principal": {
         "AWS": "arn:aws:iam::account_id:role/datasteward"
        },
        "Action": [
           "s3tables:DeleteTable",
           "s3tables:UpdateTableMetadataLocation",
           "s3tables:PutTableData",
           "s3tables:GetTableMetadataLocation"

        ],
        "Resource":"arn:aws:s3tables:region:account_id:bucket/amzn-s3-demo-bucket1/
table/tableUUID"
     }]
}
```

AWS kebijakan terkelola untuk Tabel S3

Kebijakan AWS terkelola adalah kebijakan mandiri yang dibuat dan dikelola oleh AWS. AWS Kebijakan terkelola dirancang untuk memberikan izin bagi banyak kasus penggunaan umum sehingga Anda dapat mulai menetapkan izin kepada pengguna, grup, dan peran.

Perlu diingat bahwa kebijakan AWS terkelola mungkin tidak memberikan izin hak istimewa paling sedikit untuk kasus penggunaan spesifik Anda karena tersedia untuk digunakan semua pelanggan. AWS Kami menyarankan Anda untuk mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola pelanggan yang khusus untuk kasus penggunaan Anda.

Anda tidak dapat mengubah izin yang ditentukan dalam kebijakan AWS terkelola. Jika AWS memperbarui izin yang ditentukan dalam kebijakan AWS terkelola, pembaruan akan memengaruhi semua identitas utama (pengguna, grup, dan peran) yang dilampirkan kebijakan tersebut. AWS kemungkinan besar akan memperbarui kebijakan AWS terkelola saat baru Layanan AWS diluncurkan atau operasi API baru tersedia untuk layanan yang ada.

Untuk informasi selengkapnya, lihat Kebijakan terkelola AWS dalam Panduan Pengguna IAM.

AWS kebijakan terkelola: AmazonS3TablesFullAccess

Anda dapat melampirkan kebijakan `AmazonTablesS3FullAccess` ke identitas IAM Anda. Kebijakan ini memberikan izin yang memungkinkan akses penuh ke Tabel Amazon S3.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3tables:*"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS kebijakan terkelola: AmazonS3TablesReadOnlyAccess

Anda dapat melampirkan kebijakan `AmazonS3TablesReadOnlyAccess` ke identitas IAM Anda. Kebijakan ini memberikan izin yang memungkinkan akses hanya-baca ke Tabel Amazon S3.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
```

```
                "s3tables:Get*",
                "s3tables:List*"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS kebijakan terkelola: AmazonS3TablesLakeFormationServiceRole

Anda dapat melampirkan kebijakan `AmazonS3TablesLakeFormationServiceRole` ke identitas IAM Anda. Kebijakan ini memberikan izin yang memungkinkan akses peran AWS Lake Formation layanan ke Tabel S3. AWS KMS izin digunakan untuk memungkinkan Lake Formation mengakses tabel terenkripsi.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PermissionsForS3ListTableBuckets",
            "Effect": "Allow",
            "Action": [
                "s3tables:ListTableBuckets"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "111122223333"
                }
            }
        },
        {
            "Sid": "DataAccessPermissionsForS3TablesResources",
            "Effect": "Allow",
            "Action": [
                "s3tables:CreateTableBucket",
                "s3tables:GetTableBucket",
                "s3tables:CreateNamespace",
                "s3tables:GetNamespace",
                "s3tables:ListNamespaces",
                "s3tables:DeleteNamespace",
```

```
                "s3tables:DeleteTableBucket",
                "s3tables:CreateTable",
                "s3tables:DeleteTable",
                "s3tables:GetTable",
                "s3tables:ListTables",
                "s3tables:RenameTable",
                "s3tables:UpdateTableMetadataLocation",
                "s3tables:GetTableMetadataLocation",
                "s3tables:GetTableData",
                "s3tables:PutTableData"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "111122223333"
                }
            }
        },
        {
            "Sid": "KMSDataAccessPermissionsForS3TablesResources",
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "s3.*.amazonaws.com"
                    ],
                    "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3tables:*:*:bucket/*/
table/*"
                },
                "StringEquals": {
                    "aws:ResourceAccount": "111122223333"
                }
            }
        },
        {
            "Sid": "KMSDescribeKeyAccessPermissionsForS3TablesResources",
            "Effect": "Allow",
```

```
            "Action": [
                "kms:DescribeKey"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "s3tables.*.amazonaws.com"
                    ]
                },
                "StringEquals": {
                    "aws:ResourceAccount": "111122223333"
                }
            }
        }
    ]
}
```

Tabel Amazon S3 memperbarui kebijakan terkelola AWS

Lihat detail tentang pembaruan kebijakan AWS terkelola untuk Tabel Amazon S3 sejak Tabel S3 mulai melacak perubahan ini.

Perubahan	Deskripsi	Tanggal
Tabel Amazon S3 ditambahkan. `AmazonS3TablesLakeFormationServiceRole`	Tabel S3 menambahkan kebijakan AWS-managed baru yang disebut. `AmazonS3TablesLakeFormationServiceRole` Kebijakan ini memberikan izin yang memungkinkan akses peran layanan Lake Formation ke Tabel S3.	19 Mei 2025
Tabel Amazon S3 ditambahkan. `AmazonTablesS3FullAccess`	Tabel S3 menambahkan kebijakan AWS-managed baru yang disebut. `AmazonTablesS3FullAccess` Kebijakan ini memberikan izin	Desember 03, 2024

Perubahan	Deskripsi	Tanggal
	yang memungkinkan akses penuh ke Tabel Amazon S3.	
Tabel Amazon S3 ditambahkan. `AmazonS3TablesReadOnlyAccess`	Tabel S3 menambahkan kebijakan AWS-managed baru yang disebut. `AmazonS3TablesReadOnlyAccess` Kebijakan ini memberikan izin untuk mengizinkan akses hanya-baca ke Tabel Amazon S3.	Desember 03, 2024
Tabel Amazon S3 mulai melacak perubahan.	Tabel Amazon S3 mulai melacak perubahan untuk kebijakan yang AWS dikelola.	Desember 03, 2024

Memberikan akses dengan semantik SQL

Anda dapat memberikan izin ke tabel dengan menggunakan semantik SQL dalam kebijakan bucket tabel dan tabel. Contoh semantik SQL yang dapat Anda gunakan adalah`CREATE`,,, `INSERT`, `DELETE`, `UPDATE` dan. `ALTER` Tabel berikut menyediakan daftar tindakan API yang terkait dengan semantik SQL yang dapat Anda gunakan untuk memberikan izin kepada pengguna Anda.

Tabel S3 sebagian mendukung izin menggunakan semantik SQL. Misalnya, `CreateTable` API hanya membuat tabel kosong di keranjang tabel. Anda memerlukan izin tambahan seperti,`UpdateTableMetadata`,`PutTableData`, dan `GetTableMetadataLocation` untuk dapat mengatur skema tabel. Izin tambahan ini juga berarti bahwa Anda juga memberikan akses pengguna untuk menyisipkan baris dalam tabel. Jika Anda ingin mengatur akses murni berdasarkan semantik SQL, maka sebaiknya gunakan AWS Lake Formationatau solusi pihak ketiga yang terintegrasi dengan Tabel S3.

Aktivitas tingkat meja	Tindakan IAM		
`SELECT`	`s3tables:GetTableD`		

Aktivitas tingkat meja	Tindakan IAM		
	ata , s3tables: GetTableM etadataLo cation		
CREATE	s3tables: CreateTab le , s3tables: UpdateTab leMetadat aLocation , s3tables: PutTableD ata , s3tables: GetTableM etadataLo cation ,		
INSERT	s3tables: UpdateTab leMetadat aLocation , s3tables: PutTableD ata , s3tables: GetTableM etadataLo cation		

Aktivitas tingkat meja	Tindakan IAM		
UPDATE	s3tables:UpdateTableMetadataLocation , s3tables:PutTableData , s3tables:GetTableMetadataLocation		
ALTER,RENAME	s3tables:UpdateTableMetadataLocation , s3tables:PutTableData , s3tables:GetTableMetadataLocation , s3tables:RenameTable		

Aktivitas tingkat meja	Tindakan IAM		
DELETE,DROP	s3tables:DeleteTable , s3tables:UpdateTableMetadataLocation , s3tables:PutTableData , s3tables:GetTableMetadataLocation		

Konektivitas VPC untuk Tabel S3

Semua tabel di Tabel S3 ada di Apache Iceberg format dan terdiri dari dua jenis objek S3. Kedua jenis objek ini adalah file data yang menyimpan data dan file metadata yang melacak informasi tentang file data pada titik waktu yang berbeda. Semua keranjang tabel, namespace, dan operasi tabel (misalnya,, `CreateNamespaceCreateTable`, dan seterusnya) dirutekan melalui titik akhir Tabel S3 (`s3tables.region.amazonaws.com`) dan semua operasi tingkat objek yang membaca atau menulis file data dan metadata terus dirutekan melalui titik akhir layanan S3 (). `s3.region.amazonaws.com`

Untuk mengakses Tabel S3, Amazon S3 mendukung dua jenis titik akhir VPC dengan AWS PrivateLink menggunakan: titik akhir gateway dan titik akhir antarmuka. Titik akhir gateway adalah gateway yang Anda tentukan dalam tabel rute untuk mengakses S3 dari VPC Anda melalui jaringan. AWS Titik akhir antarmuka memperluas fungsionalitas titik akhir gateway dengan menggunakan alamat IP pribadi untuk merutekan permintaan ke Amazon S3 dari dalam VPC Anda, di tempat, atau dari VPC di tempat lain dengan menggunakan peering VPC atau. Wilayah AWS AWS Transit Gateway

Untuk mengakses Tabel S3 dari VPC, sebaiknya buat dua titik akhir VPC (satu untuk S3 dan yang lainnya untuk Tabel S3). Anda dapat membuat gateway atau titik akhir antarmuka untuk merutekan operasi level file (objek) ke S3 dan titik akhir antarmuka untuk merutekan operasi tingkat bucket dan tabel ke Tabel S3. Anda dapat membuat dan menggunakan titik akhir VPC untuk permintaan tingkat file menggunakan S3. Untuk informasi selengkapnya, lihat titik akhir Gateway di Panduan AWS PrivateLinkPengguna.

Untuk mempelajari lebih lanjut tentang penggunaan AWS PrivateLink untuk membuat dan bekerja dengan titik akhir untuk Tabel S3, lihat topik berikut. Untuk membuat titik akhir antarmuka VPC, lihat Membuat titik akhir VPC dalam AWS PrivateLink Panduan.

Topik

- Membuat titik akhir VPC untuk Tabel S3
- Mengakses bucket tabel dan tabel melalui titik akhir menggunakan AWS CLI
- Mengkonfigurasi jaringan VPC saat menggunakan mesin kueri
- Membatasi akses ke Tabel S3 dalam jaringan VPC

Membuat titik akhir VPC untuk Tabel S3

Saat Anda membuat titik akhir VPC, Tabel S3 menghasilkan dua jenis nama DNS spesifik titik akhir: Regional dan Zonal.

- Nama DNS Regional adalah dari format berikut:`VPCendpointID.s3tables.AWSregion.vpce.amazonaws.com`. Misalnya, untuk ID titik akhir VPC`vpce-1a2b3c4d`, nama DNS yang dihasilkan akan mirip dengan `vpce-1a2b3c4d-5e6f.s3tables.us-east-1.vpce.amazonaws.com`
- Nama DNS Zonal adalah dari format berikut:. `VPCendpointID-AvailabilityZone.s3tables.AWSregion.vpce.amazonaws.com` Misalnya, Untuk ID titik akhir VPC`vpce-1a2b3c4d-5e6f.`, nama DNS yang dihasilkan akan mirip dengan `vpce-1a2b3c4d-5e6f-us-east-1a.s3tables.us-east-1.vpce.amazonaws.com`

  Nama DNS Zonal mencakup Availability Zone Anda. Anda dapat menggunakan nama DNS Zonal jika arsitektur Anda mengisolasi Availability Zones. Nama DNS S3 spesifik titik akhir dapat diselesaikan dari domain DNS publik S3.

Anda juga dapat menggunakan opsi DNS Pribadi untuk menyederhanakan perutean lalu lintas S3 melalui titik akhir VPC dan membantu Anda memanfaatkan jalur jaringan dengan biaya

terendah yang tersedia untuk aplikasi Anda. DNS pribadi memetakan titik akhir publik Tabel S3, misalnya`s3tables.region.amazonaws.com`, ke IP pribadi di VPC Anda. Anda dapat menggunakan opsi DNS pribadi untuk merutekan lalu lintas Regional S3 tanpa memperbarui klien S3 Anda untuk menggunakan nama DNS spesifik titik akhir dari titik akhir antarmuka Anda.

> **Note**
>
> AWS PrivateLink untuk Amazon S3 tidak mendukung penggunaan titik akhir tumpukan ganda Amazon S3. Untuk informasi selengkapnya, lihat Menggunakan titik akhir tumpukan ganda Amazon S3 di Referensi API Amazon S3.

Mengakses bucket tabel dan tabel melalui titik akhir menggunakan AWS CLI

Anda dapat menggunakan AWS Command Line Interface (AWS CLI) untuk mengakses bucket tabel dan tabel melalui titik akhir antarmuka. Dengan AWS CLI, `aws s3` perintah merutekan lalu lintas melalui titik akhir Amazon S3. `aws s3tables` AWS CLI Perintah menggunakan titik akhir Tabel Amazon S3.

Contoh titik akhir `s3tables` VPC adalah `vpce-0123456afghjipljw-nmopsqea.s3tables.region.vpce.amazonaws.com`

Endpoint `s3tables` VPC tidak menyertakan nama bucket. Anda dapat mengakses titik akhir `s3tables` VPC menggunakan perintah. `aws s3tables` AWS CLI

Contoh titik akhir `s3` VPC adalah `amzn-s3-demo-bucket.vpce-0123456afghjipljw-nmopsqea.s3.region.vpce.amazonaws.com`

Anda dapat mengakses titik akhir `s3` VPC menggunakan perintah. `aws s3` AWS CLI

Menggunakan AWS CLI

Untuk mengakses bucket tabel dan tabel melalui titik akhir antarmuka menggunakan AWS CLI, gunakan `-region` - dan `--endpoint-url` parameter. Untuk melakukan tindakan bucket tabel dan tingkat tabel, gunakan URL titik akhir Tabel S3. Untuk melakukan tindakan tingkat objek, gunakan URL titik akhir Amazon S3.

Dalam contoh berikut, ganti *`user input placeholders`* dengan informasi Anda sendiri.

Contoh 1: Gunakan URL endpoint untuk mencantumkan bucket tabel di akun Anda

```
aws s3tables list-table-buckets --endpoint https://vpce-0123456afghjipljb-
aac.s3tables.us-east-1.vpce.amazonaws.com –region us-east-1
```

Contoh 2: Gunakan URL endpoint untuk mencantumkan tabel di bucket

```
aws s3tables list-tables --table-bucket-arn arn:aws:s3tables:us-
east-1:123456789301:bucket/amzn-s3-demo-bucket --endpoint
 https://vpce-0123456afghjipljb-aac.s3tables.us-east-1.vpce.amazonaws.com --region us-
east-1
```

Mengkonfigurasi jaringan VPC saat menggunakan mesin kueri

Gunakan langkah-langkah berikut untuk mengonfigurasi jaringan VPC saat menggunakan mesin kueri.

1. Untuk memulai, Anda dapat membuat atau memperbarui VPC. Untuk informasi selengkapnya, lihat Membuat VPC.
2. Untuk operasi tingkat bucket tabel dan tabel yang merutekan ke Tabel S3, buat titik akhir antarmuka baru. Untuk informasi selengkapnya, lihat Mengakses AWS layanan menggunakan titik akhir VPC antarmuka.
3. Untuk semua operasi tingkat objek yang merutekan ke Amazon S3, buat titik akhir gateway atau titik akhir antarmuka. Untuk informasi selengkapnya tentang titik akhir gateway, lihat Membuat titik akhir gateway.
4. Selanjutnya, konfigurasikan sumber daya data Anda dan luncurkan kluster EMR Amazon. Untuk informasi selengkapnya, lihat Memulai Amazon EMR.
5. Anda kemudian dapat mengirimkan aplikasi Spark dengan konfigurasi tambahan dengan memilih nama DNS Anda dari titik akhir VPC. Misalnya, `spark.sql.catalog.ice_catalog.s3tables.endpoint` dan `https://interface-endpoint.s3tables.us-east-1.vpce.amazonaws.com` Untuk informasi selengkapnya, lihat Mengirimkan karya ke klaster EMR Amazon Anda.

Membatasi akses ke Tabel S3 dalam jaringan VPC

Mirip dengan kebijakan berbasis sumber daya, Anda dapat melampirkan kebijakan titik akhir ke titik akhir VPC yang mengontrol akses ke tabel dan bucket tabel. Dalam contoh berikut, kebijakan titik akhir antarmuka membatasi akses hanya ke bucket tabel tertentu.

```
{
    "Version": "2012-10-17",
    "Id": "Policy141511512309",
    "Statement": [{
        "Sid": "Access-to-specific-bucket-only",
        "Principal": "*",
        "Action": "s3tables:*",
        "Effect": "Allow",
        "Resource": [
            "arn:aws:s3tables:region:account_id:bucket/amzn-s3-demo-bucket",
            "arn:aws:s3tables:region:account_id:bucket/amzn-s3-demo-bucket/*"
        ]
    }]
}
```

Pertimbangan dan batasan keamanan untuk Tabel S3

Daftar berikut menjelaskan fitur dan fungsionalitas keamanan dan kontrol akses mana yang tidak didukung atau terbatas untuk Tabel S3.

- Kebijakan akses publik tidak didukung. Pengguna tidak dapat mengubah kebijakan bucket atau tabel untuk mengizinkan akses publik.
- Presigned URLs untuk mengakses objek yang terkait dengan tabel tidak didukung.
- Tag tidak didukung untuk ember meja dan tabel. Oleh karena itu, dukungan untuk kontrol akses berbasis atribut dan alokasi berbasis tag tidak tersedia.
- Permintaan yang dibuat melalui HTTP tidak didukung. Amazon S3 secara otomatis merespons dengan pengalihan HTTP untuk setiap permintaan yang dibuat melalui HTTP untuk meningkatkan permintaan ke HTTPS.
- Anda harus menggunakan AWS Signature Version 4 saat membuat permintaan ke titik akses dengan menggunakan REST APIs.
- Permintaan yang dibuat melalui Protokol Internet versi 6 (IPv6) hanya didukung untuk tindakan tingkat objek di atas titik akhir penyimpanan tabel, dan bukan untuk tindakan tingkat tabel dan ember.
- Kebijakan akses bucket dan tabel dibatasi hingga 20 KB.

Logging dengan AWS CloudTrail untuk Tabel S3

Amazon S3 terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau layanan. AWS CloudTrail menangkap semua panggilan API untuk Amazon S3 sebagai peristiwa. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat ke Amazon S3, alamat IP dari mana permintaan dibuat, kapan dibuat, dan detail tambahan. Saat aktivitas peristiwa yang didukung terjadi di Amazon S3, aktivitas tersebut direkam dalam suatu CloudTrail peristiwa. Anda dapat menggunakan AWS CloudTrail jejak untuk mencatat peristiwa manajemen dan peristiwa data untuk Tabel S3. Untuk informasi selengkapnya, lihat CloudTrail acara Amazon S3 dan Apa itu? AWS CloudTrail dalam AWS CloudTrail User Guide.

CloudTrail acara manajemen untuk Tabel S3

Acara manajemen memberikan informasi tentang operasi manajemen yang dilakukan pada sumber daya di AWS akun Anda.

Secara default, CloudTrail mencatat peristiwa manajemen untuk Tabel S3. Acara `eventsource` untuk CloudTrail manajemen untuk Tabel S3 adalah `s3tables.amazonaws.com`. Saat menyiapkan AWS akun, acara CloudTrail manajemen diaktifkan secara default. Peristiwa manajemen berikut dicatat CloudTrail.

- `CreateNamespace`
- `CreateTable`
- `CreateTableBucket`
- `DeleteNamespace`
- `DeleteTable`
- `DeleteTableBucket`
- `DeleteTableBucketPolicy`
- `DeleteTablePolicy`
- `GetNamespace`
- `GetTable`
- `GetTableBucket`
- `GetTableBucketMaintenanceConfiguration`
- `GetTableBucketPolicy`

- GetTableMaintenanceConfiguration
- GetTableMaintenanceJobStatus
- GetTableMetadataLocation
- GetTablePolicy
- ListNamespaces
- ListTableBuckets
- ListTables
- PutTableBucketMaintenanceConfiguration
- PutTableMaintenanceConfiguration
- PutBucketPolicy
- PutTablePolicy
- RenameTable
- UpdateTableMetadataLocation

Untuk informasi selengkapnya tentang peristiwa CloudTrail manajemen, lihat peristiwa manajemen log di Panduan AWS CloudTrail Pengguna.

CloudTrail peristiwa data untuk Tabel S3

Peristiwa data memberikan informasi tentang operasi sumber daya yang dilakukan pada atau di sumber daya.Secara default, CloudTrail jejak tidak mencatat peristiwa data, tetapi Anda dapat mengonfigurasi jejak untuk mencatat peristiwa data.

Saat Anda mencatat peristiwa data untuk jejak CloudTrail, Anda akan memilih atau menentukan jenis sumber daya. Tabel S3 memiliki dua jenis sumber daya, `AWS::S3Tables::Table` dan`AWS::S3Tables::TableBucket`.

Peristiwa data berikut dicatat CloudTrail.

- AbortMultipartUpload
- CompleteMultipartUpload
- CreateMultipartUpload
- GetObject
- HeadObject

- `ListParts`
- `PutObject`
- `UploadPart`

Untuk informasi selengkapnya tentang peristiwa CloudTrail data, lihat Mencatat peristiwa data di Panduan AWS CloudTrail Pengguna.

Untuk informasi tambahan tentang CloudTrail acara untuk Tabel S3, lihat topik berikut:

Topik

- AWS CloudTrail contoh file log peristiwa data untuk Tabel S3

AWS CloudTrail contoh file log peristiwa data untuk Tabel S3

File AWS CloudTrail log mencakup informasi tentang operasi API yang diminta, tanggal dan waktu operasi, parameter permintaan, dan sebagainya. Topik ini menyediakan contoh file log untuk peristiwa CloudTrail data untuk Tabel S3.

Topik

- Contoh - file CloudTrail log untuk acara GetObject data
- Contoh - file CloudTrail log untuk acara PutObject data

Contoh - file CloudTrail log untuk acara **`GetObject`** data

Contoh berikut menunjukkan contoh file CloudTrail log yang menunjukkan operasi `GetObject`API.

```
{
    "eventVersion": "1.11",
    "userIdentity": {
      "type": "IAMUser",
      "principalId": "123456789012",
      "arn": "arn:aws:iam::111122223333:user/"myUserName",
      "accountId": "111122223333",
      "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
      "userName":"myUserName"
    },
    "eventTime": "2024-11-22T17:12:25Z",
    "eventSource": "s3tables.amazonaws.com",
```

```
        "eventName": "GetObject",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "192.0.2.0",
        "userAgent": "[aws-cli/2.18.5]",
        "requestParameters": {
            "Host": "tableWarehouseLocation.s3.us-east-1.amazonaws.com",
            "key": "product-info.json"
        },
        "responseElements":  null,
        "additionalEventData": {
            "SignatureVersion": "SigV4",
            "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
            "bytesTransferredIn": 0,
            "AuthenticationMethod": "AuthHeader",
            "xAmzId2": "q6xhNJYmhg",
            "bytesTransferredOut": 28441

          },
          "requestID": "07D681123BD12AED",
          "eventID": "f2b287f3-0df1-1234-a2f4-c4bdfed47657",
          "readOnly": true,
          "resources": [{
              "accountId": "111122223333",
              "type": "AWS::S3Tables::TableBucket",
               "ARN": "arn:aws:s3tables:us-east-1:111122223333:bucket/amzn-s3-demo-
bucket1"
           }, {
              "accountId": "111122223333",
              "type": "AWS::S3Tables::Table",
              "ARN": "arn:aws:s3tables:us-east-1:111122223333:bucket/amzn-s3-demo-
bucket/table/111aa1111-22bb-33cc-44dd-5555eee66ffff"

           }],
           "eventType": "AwsApiCall",
           "managementEvent": false,
           "recipientAccountId": "444455556666",
           "eventCategory": "Data",
           "tlsDetails": {
             "tlsVersion": "TLSv1.2",
             "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
             "clientProvidedHostHeader": "tableWarehouseLocation.s3.us-
east-1.amazonaws.com"
          }
```

```
    }
```

Contoh - file CloudTrail log untuk acara **`PutObject`** data

Contoh berikut menunjukkan contoh file CloudTrail log yang menunjukkan operasi PutObjectAPI.

```
{
        "eventVersion": "1.11",
        "userIdentity": {
          "type": "IAMUser",
          "principalId": "123456789012",
          "arn":  "arn:aws:iam::444455556666:user/"myUserName",
          "accountId": "444455556666",
          "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
          "userName":"myUserName"
        },
        "eventTime": "2024-11-22T17:12:25Z",
        "eventSource": "s3tables.amazonaws.com",
        "eventName": "PutObject",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "192.0.2.0",
        "userAgent": "[aws-cli/2.18.5]",
        "requestParameters": {
            "Host": "tableWarehouseLocation.s3.us-east-1.amazonaws.com",
            "key": "product-info.json"
        },
        "responseElements":  {
            "x-amz-server-side-encryption": "AES256",
            "x-amz-version-id": "13zAFMdccAjt3MWd6ehxgCCCDRdkAKDw"
        },
        "additionalEventData": {
            "SignatureVersion": "SigV4",
            "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
            "bytesTransferredIn": 28441,
            "AuthenticationMethod": "AuthHeader",
            "xAmzId2": "q6xhCJYmhg",
            "bytesTransferredOut": 0

          },
          "requestID": "28d2faaf-1234-4649-997d-EXAMPLE72818",
          "eventID": "694d604a-d190-1234-0dd1-EXAMPLEe20c1",
          "readOnly": false,
          "resources": [{
```

```
              "accountId": "444455556666",
              "type": "AWS::S3Tables::TableBucket",
               "ARN": "arn:aws:s3tables:us-east-1:444455556666:bucket/amzn-s3-demo-
bucket1"
           }, {
              "accountId": "444455556666",
              "type": "AWS::S3Tables::Table",
              "ARN": "arn:aws:s3tables:us-east-1:444455556666:bucket/amzn-s3-demo-
bucket1/table/b89ec883-b1d9-4b37-9cd7-b86f590123f4"
           }],
           "eventType": "AwsApiCall",
           "managementEvent": false,
           "recipientAccountId": "111122223333",
           "eventCategory": "Data",
           "tlsDetails": {
             "tlsVersion": "TLSv1.2",
             "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
             "clientProvidedHostHeader": "tableWarehouseLocation.s3.us-
east-1.amazonaws.com"
            }
         }
```

Kontrol akses di Amazon S3

Di AWS, sumber daya adalah entitas yang dapat Anda gunakan. Di Amazon Simple Storage Service (S3), bucket dan objek adalah sumber daya Amazon S3 asli. Setiap pelanggan S3 kemungkinan memiliki ember dengan benda-benda di dalamnya. Karena fitur baru ditambahkan ke S3, sumber daya tambahan juga ditambahkan, tetapi tidak setiap pelanggan menggunakan sumber daya khusus fitur ini. Untuk informasi selengkapnya tentang sumber daya Amazon S3, lihat. Sumber daya S3

Secara default, semua sumber daya Amazon S3 bersifat pribadi. Juga secara default, pengguna root dari Akun AWS yang menciptakan sumber daya (pemilik sumber daya) dan pengguna IAM dalam akun itu dengan izin yang diperlukan dapat mengakses sumber daya yang mereka buat. Pemilik sumber daya memutuskan siapa lagi yang dapat mengakses sumber daya dan tindakan yang diizinkan dilakukan orang lain pada sumber daya. S3 memiliki berbagai alat manajemen akses yang dapat Anda gunakan untuk memberi orang lain akses ke sumber daya S3 Anda.

Bagian berikut memberi Anda gambaran umum tentang sumber daya S3, alat manajemen akses S3 yang tersedia, dan kasus penggunaan terbaik untuk setiap alat manajemen akses. Daftar di bagian ini bertujuan untuk menjadi komprehensif dan mencakup semua sumber daya S3, alat manajemen akses, dan kasus penggunaan manajemen akses umum. Pada saat yang sama, bagian ini dirancang untuk menjadi direktori yang mengarahkan Anda ke detail teknis yang Anda inginkan. Jika Anda memiliki pemahaman yang baik tentang beberapa topik berikut, Anda dapat melompat ke bagian yang berlaku untuk Anda.

Untuk informasi selengkapnya tentang izin operasi S3 API menurut jenis sumber daya S3, lihat. Izin yang diperlukan untuk operasi API Amazon S3

Topik

- Sumber daya S3
- Identitas
- Alat manajemen akses
- Tindakan
- Kasus penggunaan manajemen akses
- Pemecahan masalah manajemen akses

Sumber daya S3

Sumber daya Amazon S3 asli adalah ember dan objek yang dikandungnya. Karena fitur baru ditambahkan ke S3, sumber daya baru juga ditambahkan. Berikut ini adalah daftar lengkap sumber daya S3 dan fitur-fiturnya masing-masing.

Jenis sumber daya	Fitur Amazon S3	Deskripsi
`bucket`	Fitur inti	Bucket adalah kontainer untuk objek. Untuk menyimpan objek di S3, buat bucket lalu unggah satu atau beberapa objek ke bucket. Untuk informasi selengkapnya, lihat Membuat, mengonfigurasi, dan bekerja dengan bucket tujuan umum Amazon S3.
`object`		Objek dapat berupa file dan metadata apa pun yang menggambarkan file itu. Ketika sebuah objek ada di ember, Anda dapat membukanya, mengunduhnya, dan memindahk annya. Untuk informasi selengkapnya, lihat Bekerja dengan objek di Amazon S3.
`accesspoi nt`	Titik Akses	Titik Akses diberi nama titik akhir jaringan yang dilampirk an ke bucket yang dapat Anda gunakan untuk melakukan operasi objek Amazon S3, seperti dan. `GetObject` `PutObject` Setiap titik akses memiliki izin berbeda, kontrol jaringan, dan kebijakan jalur akses khusus yang berfungsi bersama dengan kebijakan bucket yang dilampirk an ke bucket yang mendasarinya. Anda dapat mengonfig urasi titik akses apa pun untuk menerima permintaan hanya dari virtual private cloud (VPC) atau mengonfigurasi pengaturan akses publik blok khusus untuk setiap titik akses. Untuk informasi selengkapnya, lihat Mengelola akses ke kumpulan data bersama dengan titik akses.
`objectlam bdaaccess point`		Object Lambda Access Point adalah titik akses untuk bucket yang juga terkait dengan fungsi Lambda. Dengan Object Lambda Access Point, Anda dapat menambahkan kode

Jenis sumber daya	Fitur Amazon S3	Deskripsi
		Anda sendiri ke Amazon `GET` `S3LIST`, `HEAD` dan meminta untuk memodifikasi dan memproses data saat dikembalikan ke aplikasi. Untuk informasi selengkapnya, lihat Membuat Titik Akses Objek Lambda.
`multiregionaccesspoint`		Titik Akses Multi-Wilayah menyediakan titik akhir global yang dapat digunakan aplikasi untuk memenuhi permintaan dari bucket Amazon S3 yang berlokasi di beberapa Wilayah. AWS Anda dapat menggunakan Titik Akses Multi-Wilayah untuk membangun aplikasi multi-Wilayah dengan arsitektur yang sama dengan yang digunakan di satu Wilayah, dan kemudian menjalankan aplikasi tersebut di mana saja di seluruh dunia. Alih-alih mengirim permintaan melalui internet publik yang padat, permintaan aplikasi yang dibuat ke titik akhir global Multi-Region Access Point secara otomatis merutekan melalui jaringan AWS global ke bucket Amazon S3 terdekat. Untuk informasi selengkapnya, lihat Mengelola lalu lintas Multi-wilayah dengan Titik Akses Multi-Wilayah.
`job`	Operasi Batch S3	Pekerjaan adalah sumber daya dari fitur Operasi Batch S3. Anda dapat menggunakan Operasi Batch S3 untuk melakukan operasi batch skala besar pada daftar objek Amazon S3 yang Anda tentukan. Amazon S3 melacak kemajuan pekerjaan operasi batch, mengirimkan pemberitahuan, dan menyimpan laporan penyelesaian terperinci dari semua tindakan, memberi Anda pengalaman yang sepenuhnya dikelola, dapat diaudit, dan tanpa server. Untuk informasi selengkapnya, lihat Melakukan operasi objek secara massal dengan Operasi Batch.

Jenis sumber daya	Fitur Amazon S3	Deskripsi
`storagelensconfiguration`	Lensa Penyimpanan S3	Konfigurasi Lensa Penyimpanan S3 mengumpulkan metrik penyimpanan di seluruh organisasi dan data pengguna di seluruh akun. S3 Storage Lens memberi admin satu tampilan penggunaan dan aktivitas penyimpanan objek di ratusan, atau bahkan ribuan, akun dalam suatu organisasi, dengan detail untuk menghasilkan wawasan di berbagai tingkat agregasi. Untuk informasi selengkapnya, lihat Menilai aktivitas penyimpanan dan penggunaan dengan Lensa Penyimpanan Amazon S3.
`storagelensgroup`		Grup Lensa Penyimpanan S3 menggabungkan metrik dengan menggunakan filter khusus berdasarkan metadata objek. Grup Lensa Penyimpanan S3 membantu Anda menyelidiki karakteristik data Anda, seperti distribusi objek berdasarkan usia, jenis file yang paling umum, dan banyak lagi. Untuk informasi selengkapnya, lihat Bekerja dengan grup Lensa Penyimpanan S3 untuk memfilter dan mengumpulkan metrik.
`accessgrantsinstance`	Pemberian Akses S3	Instance S3 Access Grants adalah wadah untuk hibah S3 yang Anda buat. Dengan S3 Access Grants, Anda dapat membuat hibah ke data Amazon S3 untuk identitas IAM dalam akun Anda, identitas IAM di akun lain (lintas akun), dan identitas direktori yang ditambahkan dari direktori perusahaan Anda. AWS IAM Identity Center Untuk informasi selengkapnya tentang Hibah Akses S3, lihat. Mengelola akses dengan S3 Access Grants

Jenis sumber daya	Fitur Amazon S3	Deskripsi
`accessgrantslocation`		Lokasi Access Grants adalah bucket, awalan dalam bucket, atau objek yang Anda daftarkan di instance S3 Access Grants. Anda harus mendaftarkan lokasi dalam instance S3 Access Grants sebelum Anda dapat membuat hibah ke lokasi tersebut. Kemudian, dengan S3 Access Grants, Anda dapat memberikan akses ke bucket, awalan, atau objek untuk identitas IAM dalam akun Anda, identitas IAM di akun lain (lintas akun), dan identitas direktori yang ditambahkan dari direktori perusahaan Anda. AWS IAM Identity Center Untuk informasi selengkapnya tentang Hibah Akses S3, lihat Mengelola akses dengan S3 Access Grants
`accessgrant`		Hibah Akses adalah hibah individu untuk data Amazon S3 Anda. Dengan S3 Access Grants, Anda dapat membuat hibah ke data Amazon S3 untuk identitas IAM dalam akun Anda, identitas IAM di akun lain (lintas akun), dan identitas direktori yang ditambahkan dari direktori perusahaan Anda. AWS IAM Identity Center Untuk informasi selengkapnya tentang Hibah Akses S3, lihat Mengelola akses dengan S3 Access Grants

Bucket

Ada dua jenis bucket Amazon S3: ember tujuan umum dan ember direktori.

- Bucket tujuan umum adalah jenis bucket S3 asli dan direkomendasikan untuk sebagian besar kasus penggunaan dan pola akses. Bucket tujuan umum juga memungkinkan objek yang disimpan di semua kelas penyimpanan, kecuali S3 Express One Zone. Untuk informasi selengkapnya tentang kelas penyimpanan S3, lihatMemahami dan mengelola kelas penyimpanan Amazon S3.
- Bucket direktori menggunakan kelas penyimpanan S3 Express One Zone, yang direkomendasikan jika aplikasi Anda peka terhadap kinerja dan mendapat manfaat dari milidetik dan latensi satu digit. `PUT GET` Lihat informasi selengkapnya di Bekerja dengan bucket direktori, S3 Express One Zone, dan Mengotorisasi operasi API endpoint Regional dengan IAM.

Mengkategorikan sumber daya S3

Amazon S3 menyediakan fitur untuk mengkategorikan dan mengatur sumber daya S3 Anda. Mengkategorikan sumber daya Anda tidak hanya berguna untuk mengaturnya, tetapi Anda juga dapat menetapkan aturan manajemen akses berdasarkan kategori sumber daya. Secara khusus, awalan dan penandaan adalah dua fitur organisasi penyimpanan yang dapat Anda gunakan saat mengatur izin manajemen akses.

> **Note**
>
> Informasi berikut berlaku untuk ember tujuan umum. Bucket direktori tidak mendukung penandaan, dan mereka memiliki batasan awalan. Untuk informasi selengkapnya, lihat Mengotorisasi operasi API endpoint Regional dengan IAM.

- Awalan - Awalan di Amazon S3 adalah string karakter di awal nama kunci objek yang digunakan untuk mengatur objek yang disimpan di bucket S3 Anda. Anda dapat menggunakan karakter pembatas, seperti garis miring maju (/), untuk menunjukkan akhir awalan dalam nama kunci objek. Misalnya, Anda mungkin memiliki nama kunci objek yang dimulai dengan `engineering/` awalan atau nama kunci objek yang dimulai dengan `marketing/campaigns/` awalan. Menggunakan delimeter di akhir awalan Anda, seperti karakter garis miring maju `/` mengemulasi folder dan konvensi penamaan file. Namun, di S3, awalan adalah bagian dari nama kunci objek. Dalam bucket S3 tujuan umum, tidak ada hierarki folder yang sebenarnya.

  Amazon S3 mendukung pengorganisasian dan pengelompokan objek dengan menggunakan awalan mereka. Anda juga dapat mengelola akses ke objek dengan awalan mereka. Misalnya, Anda dapat membatasi akses hanya ke objek dengan nama yang dimulai dengan awalan tertentu.

  Untuk informasi selengkapnya, lihat Organisasi objek menggunakan prefiks. Konsol S3 menggunakan konsep folder, yang, dalam bucket tujuan umum, pada dasarnya adalah awalan yang telah ditambahkan ke nama kunci objek. Untuk informasi selengkapnya, lihat Mengatur objek di konsol Amazon S3 dengan menggunakan folder.
- Tag - Setiap tag adalah pasangan nilai kunci yang Anda tetapkan ke sumber daya. Misalnya, Anda dapat menandai beberapa sumber daya dengan tag`topicCategory=engineering`. Anda dapat menggunakan penandaan untuk membantu alokasi biaya, pengkategorian dan pengorganisasian, dan kontrol akses. Bucket tagging hanya digunakan untuk alokasi biaya. Anda dapat menandai objek, Lensa Penyimpanan S3, pekerjaan, dan Hibah Akses S3 untuk keperluan pengorganisasian atau untuk kontrol akses. Di S3 Access Grants, Anda juga dapat menggunakan penandaan untuk

alokasi biaya. Sebagai contoh mengendalikan akses ke sumber daya dengan menggunakan tag mereka, Anda hanya dapat berbagi objek yang memiliki tag tertentu atau kombinasi tag.

Untuk informasi selengkapnya, lihat Mengontrol akses ke sumber daya AWS menggunakan tanda sumber daya di Panduan Pengguna IAM.

Identitas

Di Amazon S3, pemilik sumber daya adalah identitas yang menciptakan sumber daya, seperti ember atau objek. Secara default, hanya pengguna root akun yang membuat sumber daya dan identitas IAM dalam akun yang memiliki izin yang diperlukan dapat mengakses sumber daya S3. Pemilik sumber daya dapat memberikan identitas lain akses ke sumber daya S3 mereka.

Identitas yang tidak memiliki sumber daya dapat meminta akses ke sumber daya tersebut. Permintaan ke sumber daya diautentikasi atau tidak diautentikasi. Permintaan yang diautentikasi harus menyertakan nilai tanda tangan yang mengautentikasi pengirim permintaan, tetapi permintaan yang tidak diautentikasi tidak memerlukan tanda tangan. Kami menyarankan Anda memberikan akses hanya kepada pengguna yang diautentikasi. Untuk informasi selengkapnya tentang autentikasi permintaan, lihat Membuat permintaan di Referensi API Amazon S3.

> **Important**
>
> Kami menyarankan agar Anda tidak menggunakan kredensi pengguna Akun AWS root untuk membuat permintaan yang diautentikasi. Sebagai gantinya, buat peran IAM dan berikan akses penuh kepada peran tersebut. Kami merujuk ke pengguna dengan peran ini sebagai pengguna administrator. Anda dapat menggunakan kredensyal yang ditetapkan ke peran administrator, bukan kredensyal pengguna Akun AWS root, untuk berinteraksi AWS dan melakukan tugas, seperti membuat bucket, membuat pengguna, dan memberikan izin. Untuk informasi selengkapnya, lihat kredensyal pengguna Akun AWS root dan kredensyal pengguna IAM di Referensi Umum AWS, dan lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.

Identitas mengakses data Anda di Amazon S3 dapat menjadi salah satu dari yang berikut:

Akun AWS pemilik

Akun AWS Yang menciptakan sumber daya. Misalnya, akun yang membuat ember. Akun ini memiliki sumber daya. Untuk informasi selengkapnya, lihat pengguna root AWS akun.

Identitas IAM di akun pemilik yang sama Akun AWS

Saat menyiapkan akun untuk anggota tim baru yang memerlukan akses S3, Akun AWS pemilik dapat menggunakan AWS Identity and Access Management (IAM) untuk membuat pengguna, grup, dan peran. Akun AWS Pemilik kemudian dapat berbagi sumber daya dengan identitas IAM ini. Pemilik akun juga dapat menentukan izin untuk memberikan identitas IAM, yang memungkinkan atau menolak tindakan yang dapat dilakukan pada sumber daya bersama.

Identitas IAM memberikan peningkatan kemampuan, termasuk kemampuan untuk meminta pengguna memasukkan kredensi login sebelum mengakses sumber daya bersama. Dengan menggunakan identitas IAM, Anda dapat menerapkan bentuk otentikasi multi-faktor IAM (MFA) untuk mendukung fondasi identitas yang kuat. Praktik terbaik IAM adalah membuat peran untuk manajemen akses alih-alih memberikan izin kepada setiap pengguna individu. Anda menetapkan pengguna individu ke peran yang sesuai. Untuk informasi lebih lanjut, lihat Praktik terbaik keamanan di IAM.

Pemilik AWS akun lain dan identitas IAM mereka (akses lintas akun)

Akun AWS Pemilik juga dapat memberikan pemilik AWS akun lain, atau identitas IAM milik AWS akun lain, akses ke sumber daya.

> **Note**
>
> Delegasi izin — Jika Akun AWS memiliki sumber daya, ia dapat memberikan izin tersebut kepada yang lain. Akun AWS Akun itu kemudian dapat mendelegasikan izin tersebut, atau sebagian dari mereka, kepada pengguna di akun yang sama. Ini disebut sebagai delegasi izin. Tetapi akun yang menerima izin dari akun lain tidak dapat mendelegasikan izin tersebut "lintas akun" ke akun lain. Akun AWS

Pengguna anonim (akses publik)

Akun AWS Pemilik dapat membuat sumber daya publik. Membuat sumber daya publik secara teknis berbagi sumber daya dengan pengguna anonim. Bucket yang dibuat sejak April 2023 memblokir semua akses publik secara default, kecuali jika Anda mengubah pengaturan ini. Kami menyarankan Anda mengatur bucket untuk memblokir akses publik, dan Anda hanya memberikan akses ke

pengguna yang diautentikasi. Untuk informasi lebih lanjut tentang pemblokiran akses publik, lihat Melakukan blok akses publik ke penyimpanan Amazon S3 Anda.

Layanan AWS

Pemilik sumber daya dapat memberikan akses AWS layanan lain ke sumber daya Amazon S3. Misalnya, Anda dapat memberikan `s3:PutObject` izin AWS CloudTrail layanan untuk menulis file log ke bucket Anda. Untuk informasi selengkapnya, lihat Menyediakan akses ke AWS layanan.

Identitas direktori perusahaan

Pemilik sumber daya dapat memberikan pengguna atau peran dari direktori perusahaan Anda akses ke sumber daya S3 dengan menggunakan S3 Access Grants. Untuk informasi selengkapnya tentang menambahkan direktori perusahaan Anda AWS IAM Identity Center, lihat Apa itu Pusat Identitas IAM? .

Pemilik ember atau sumber daya

Akun AWS Yang Anda gunakan untuk membuat bucket dan mengunggah objek memiliki sumber daya tersebut. Pemilik bucket dapat memberikan izin lintas akun kepada Akun AWS lain (atau pengguna di akun lain) untuk unggah objek.

Ketika pemilik bucket mengizinkan akun lain untuk mengunggah objek ke bucket, pemilik bucket, secara default, memiliki semua objek yang diunggah ke bucket mereka. Namun, jika pengaturan bucket yang dipaksakan oleh pemilik Bucket dan pemilik Bucket dinonaktifkan, Akun AWS yang mengunggah objek memiliki objek tersebut, dan pemilik bucket tidak memiliki izin pada objek yang dimiliki oleh akun lain, dengan pengecualian berikut:

- Pemilik bucket membayar tagihan. Pemilik bucket dapat menolak akses ke objek apa pun, atau menghapus objek apa pun di dalam bucket, tanpa memandang siapa yang memilikinya.
- Pemilik bucket dapat mengarsipkan objek apa pun atau mengembalikan objek yang diarsipkan, terlepas dari siapa yang memilikinya. Pengarsipan mengacu pada kelas penyimpanan yang digunakan untuk menyimpan objek. Untuk informasi selengkapnya, lihat Mengelola siklus hidup objek.

Alat manajemen akses

Amazon S3 menyediakan beragam fitur dan alat keamanan. Berikut ini adalah daftar lengkap fitur dan alat ini. Anda tidak memerlukan semua alat manajemen akses ini, tetapi Anda harus

menggunakan satu atau lebih untuk memberikan akses ke sumber daya Amazon S3 Anda. Aplikasi yang tepat dari alat-alat ini dapat membantu memastikan bahwa sumber daya Anda hanya dapat diakses oleh pengguna yang dituju.

Alat manajemen akses yang paling umum digunakan adalah kebijakan akses. Kebijakan akses dapat berupa kebijakan berbasis sumber daya yang dilampirkan ke AWS sumber daya, seperti kebijakan bucket untuk bucket. Kebijakan akses juga dapat berupa kebijakan berbasis identitas yang dilampirkan pada identitas AWS Identity and Access Management (IAM), seperti pengguna, grup, atau peran IAM. Tulis kebijakan akses untuk memberikan Akun AWS izin kepada pengguna, grup, dan peran IAM untuk melakukan operasi pada sumber daya. Misalnya, Anda dapat memberikan `PUT Object` izin kepada yang lain Akun AWS sehingga akun lain dapat mengunggah objek ke bucket Anda.

Kebijakan akses menjelaskan siapa yang memiliki akses ke hal-hal apa. Saat Amazon S3 menerima permintaan, Amazon S3 harus mengevaluasi semua kebijakan akses untuk menentukan apakah akan mengotorisasi atau menolak permintaan tersebut. Untuk informasi lebih lanjut tentang cara Amazon S3 mengevaluasi kebijakan ini, lihat Bagaimana Amazon S3 Mengotorisasi Permintaan.

Berikut ini adalah alat manajemen akses yang tersedia di Amazon S3.

Kebijakan bucket

Kebijakan bucket Amazon S3 adalah kebijakan berbasis sumber daya berformat JSON AWS Identity and Access Management (IAM) yang dilampirkan ke bucket tertentu. Gunakan kebijakan bucket untuk memberikan izin identitas lain Akun AWS atau IAM untuk bucket dan objek di dalamnya. Banyak kasus penggunaan manajemen akses S3 dapat dipenuhi dengan menggunakan kebijakan bucket. Dengan kebijakan bucket, Anda dapat mempersonalisasi akses bucket untuk membantu memastikan bahwa hanya identitas yang telah Anda setujui yang dapat mengakses sumber daya dan melakukan tindakan di dalamnya. Untuk informasi selengkapnya, lihat Kebijakan bucket untuk Amazon S3.

Berikut ini adalah contoh kebijakan bucket. Anda mengekspresikan kebijakan bucket dengan menggunakan file JSON. Kebijakan contoh ini memberikan izin baca peran IAM ke semua objek di bucket. Ini berisi satu pernyataan bernama`BucketLevelReadPermissions`, yang memungkinkan `s3:GetObject` tindakan (izin baca) pada objek dalam ember bernama`amzn-s3-demo-bucket1`. Dengan menetapkan peran IAM sebagai`Principal`, kebijakan ini memberikan akses ke setiap pengguna IAM dengan peran ini. Untuk menggunakan kebijakan contoh ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid":"BucketLevelReadPermissions",
      "Effect":"Allow",
      "Principal": {
     "AWS": "arn:aws:iam::123456789101:role/s3-role"
      },
      "Action":["s3:GetObject"],
      "Resource":["arn:aws:s3:::amzn-s3-demo-bucket1/*"]
    }]
  }
```

> **Note**
>
> Saat membuat kebijakan, hindari penggunaan karakter wildcard (*) di elemen `Principal` karena penggunaan karakter wildcard memungkinkan siapa pun mengakses sumber daya Amazon S3 Anda. Sebagai gantinya, secara eksplisit mencantumkan pengguna atau grup yang diizinkan mengakses bucket, atau mencantumkan kondisi yang harus dipenuhi dengan menggunakan klausa kondisi dalam kebijakan. Selain itu, daripada menyertakan karakter wildcard untuk tindakan pengguna atau grup Anda, berikan mereka izin khusus jika berlaku.

Kebijakan berbasis identitas

Kebijakan pengguna berbasis identitas atau IAM adalah jenis kebijakan AWS Identity and Access Management (IAM). Kebijakan berbasis identitas adalah kebijakan berformat JSON yang dilampirkan ke pengguna, grup, atau peran IAM di akun Anda. AWS Anda dapat menggunakan kebijakan berbasis identitas untuk memberikan akses identitas IAM ke bucket atau objek Anda. Anda dapat membuat pengguna, grup, dan peran IAM di akun Anda dan melampirkan kebijakan akses kepada mereka. Anda kemudian dapat memberikan akses ke AWS sumber daya, termasuk sumber daya Amazon S3. Untuk informasi selengkapnya, lihat Kebijakan berbasis identitas untuk Amazon S3.

Berikut ini adalah contoh kebijakan berbasis identitas. Kebijakan contoh memungkinkan peran IAM terkait untuk melakukan enam tindakan (izin) Amazon S3 yang berbeda pada bucket dan objek di dalamnya. Jika Anda melampirkan kebijakan ini ke peran IAM di akun Anda dan menetapkan peran tersebut ke beberapa pengguna IAM Anda, pengguna dengan peran ini akan dapat melakukan

tindakan ini pada sumber daya (bucket) yang ditentukan dalam kebijakan Anda. Untuk menggunakan kebijakan contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
{
  "Version": "2012-10-17",
  "Statement": [
  {
    "Sid": "AssignARoleActions",
    "Effect": "Allow",
    "Action": [
   "s3:PutObject",
   "s3:GetObject",
   "s3:ListBucket",
   "s3:DeleteObject",
   "s3:GetBucketLocation"
    ],
    "Resource": [
   "arn:aws:s3:::amzn-s3-demo-bucket1/*",
   "arn:aws:s3:::amzn-s3-demo-bucket1"
    ]
    },
  {
    "Sid": "AssignARoleActions2",
    "Effect": "Allow",
    "Action": "s3:ListAllMyBuckets",
    "Resource": "*"
  }
  ]
}
```

Pemberian Akses S3

Gunakan Hibah Akses S3 untuk membuat hibah akses ke data Amazon S3 Anda untuk kedua identitas di direktori identitas perusahaan, sepertiActive Directory, dan identitas (IAM). AWS Identity and Access Management S3 Access Grants membantu Anda mengelola izin data dalam skala besar. Selain itu, S3 Access Grants mencatat identitas pengguna akhir dan aplikasi yang digunakan untuk mengakses data S3 di. AWS CloudTrail Ini memberikan riwayat audit terperinci hingga identitas pengguna akhir untuk semua akses ke data di bucket S3 Anda. Untuk informasi selengkapnya, lihat Mengelola akses dengan S3 Access Grants.

Titik Akses

Titik Akses Amazon S3 menyederhanakan pengelolaan akses data dalam skala besar untuk aplikasi yang menggunakan kumpulan data bersama di S3. Access Points diberi nama endpoint jaringan yang dilampirkan ke bucket. Anda dapat menggunakan titik akses untuk melakukan operasi objek S3 dalam skala besar, seperti mengunggah dan mengambil objek. Bucket dapat memiliki hingga 10.000 titik akses yang terpasang, dan untuk setiap titik akses, Anda dapat menerapkan izin dan kontrol jaringan yang berbeda untuk memberi Anda kontrol terperinci atas akses ke objek S3 Anda. Poin Akses S3 dapat dikaitkan dengan bucket di akun yang sama atau di akun tepercaya lainnya. Kebijakan Access Points adalah kebijakan berbasis sumber daya yang dievaluasi bersama dengan kebijakan bucket yang mendasarinya. Untuk informasi selengkapnya, lihat Mengelola akses ke kumpulan data bersama dengan titik akses.

Daftar kontrol akses (ACL)

ACL adalah daftar hibah yang mengidentifikasi penerima hibah dan izin yang diberikan. ACLs memberikan izin baca atau tulis dasar kepada yang lain Akun AWS. ACLs gunakan skema XML-khusus Amazon S3. ACL adalah jenis kebijakan AWS Identity and Access Management (IAM). Objek ACL digunakan untuk mengelola akses ke objek, dan bucket ACL digunakan untuk mengelola akses ke bucket. Dengan kebijakan bucket, ada satu kebijakan untuk seluruh bucket, tetapi objek ACLs ditentukan untuk setiap objek. Kami menyarankan agar Anda tetap ACLs dimatikan, kecuali dalam keadaan yang tidak biasa di mana Anda harus mengontrol akses secara individual untuk setiap objek. Untuk informasi selengkapnya tentang penggunaan ACLs, lihatMengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.

> **Warning**
>
> Mayoritas kasus penggunaan modern di Amazon S3 tidak memerlukan penggunaan. ACLs

Berikut ini adalah contoh bucket ACL. Hibah di ACL menunjukkan pemilik bucket yang memiliki izin kontrol penuh.

```
<?xml version="1.0" encoding="UTF-8"?>
 <AccessControlPolicy xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Owner>
   <ID>Owner-Canonical-User-ID</ID>
   <DisplayName>owner-display-name</DisplayName>
  </Owner>
```

```
 <AccessControlList>
  <Grant>
   <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Canonical
 User">
    <ID>Owner-Canonical-User-ID</ID>
    <DisplayName>display-name</DisplayName>
   </Grantee>
   <Permission>FULL_CONTROL</Permission>
  </Grant>
 </AccessControlList>
</AccessControlPolicy>
```

Kepemilikan Objek

Untuk mengelola akses ke objek Anda, Anda harus menjadi pemilik objek. Anda dapat menggunakan setelan tingkat ember Kepemilikan Objek untuk mengontrol kepemilikan objek yang diunggah ke bucket. Juga, gunakan Object Ownership untuk mengaktifkan ACLs. Secara default, Kepemilikan Objek diatur ke setelan diberlakukan pemilik Bucket dan ACLs semuanya dimatikan. Saat ACLs dimatikan, pemilik bucket memiliki semua objek di bucket dan secara eksklusif mengelola akses ke data. Untuk mengelola akses, pemilik bucket menggunakan kebijakan atau alat manajemen akses lainnya, tidak termasuk. ACLs Untuk informasi selengkapnya, lihat Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.

Object Ownership memiliki tiga pengaturan yang dapat Anda gunakan untuk mengontrol kepemilikan objek yang diunggah ke bucket dan untuk mengaktifkanACLs:

ACLs dimatikan

- Pemilik bucket diberlakukan (default) — ACLs dimatikan, dan pemilik bucket secara otomatis memiliki dan memiliki kontrol penuh atas setiap objek di bucket. ACLs tidak memengaruhi izin ke data di bucket S3. Bucket menggunakan kebijakan secara eksklusif untuk menentukan kontrol akses.

ACLs dihidupkan

- Pemilik bucket yang dipilih—Pemilik bucket memiliki dan diberikan kendali penuh atas objek baru yang ditulis akun lain ke bucket dengan ACL `bucket-owner-full-control` yang dibatasi.
- Penulis objek — Akun AWS Yang mengunggah objek memiliki objek, memiliki kontrol penuh atasnya, dan dapat memberikan pengguna lain akses ke sana melalui. ACLs

Praktik terbaik tambahan

Pertimbangkan untuk menggunakan pengaturan dan alat bucket berikut untuk membantu melindungi data saat transit dan saat istirahat, keduanya sangat penting dalam menjaga integritas dan aksesibilitas data Anda:

- Blokir Akses Publik - Jangan matikan pengaturan tingkat ember default Blokir Akses Publik. Pengaturan ini memblokir akses publik ke data Anda secara default. Untuk informasi lebih lanjut tentang pemblokiran akses publik, lihat Melakukan blok akses publik ke penyimpanan Amazon S3 Anda.
- Pembuatan Versi S3 — Untuk integritas data, Anda dapat menerapkan pengaturan bucket S3 Versioning, yang membuat versi objek Anda saat Anda melakukan pembaruan, alih-alih menimpanya. Anda dapat menggunakan S3 Versioning untuk mempertahankan, mengambil, dan memulihkan versi sebelumnya, jika diperlukan. Untuk informasi tentang Penentuan Versi S3, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.
- S3 Object Lock - S3 Object Lock adalah pengaturan lain yang dapat Anda terapkan untuk mencapai integritas data. Fitur ini dapat mengimplementasikan model write-once-read-many (WORM) untuk menyimpan objek secara abadi. Untuk informasi tentang Kunci Objek, lihat Mengunci objek dengan Object Lock.
- Enkripsi objek — Amazon S3 menawarkan beberapa opsi enkripsi objek yang melindungi data saat transit dan saat istirahat. Enkripsi sisi server mengenkripsi objek Anda sebelum menyimpannya di disk di pusat datanya dan kemudian mendekripsi saat Anda mengunduh objek. Jika Anda mengautentikasi permintaan Anda dan Anda memiliki izin akses, tidak ada perbedaan dalam cara Anda mengakses objek terenkripsi atau tidak terenkripsi. Untuk informasi selengkapnya, lihat Melindungi data dengan enkripsi di sisi klien. S3 mengenkripsi objek yang baru diunggah secara default. Untuk informasi selengkapnya, lihat Mengatur perilaku enkripsi sisi server default untuk bucket Amazon S3. Enkripsi sisi klien adalah tindakan mengenkripsi data sebelum mengirimkannya ke Amazon S3. Untuk informasi selengkapnya, lihat Melindungi data menggunakan enkripsi di sisi klien.
- Metode penandatanganan — Signature Version 4 adalah proses menambahkan informasi otentikasi ke AWS permintaan yang dikirim oleh HTTP. Untuk keamanan, sebagian besar permintaan AWS harus ditandatangani dengan kunci akses, yang terdiri dari ID kunci akses dan kunci akses rahasia. Kedua kunci ini umumnya disebut sebagai kredensial keamanan Anda. Untuk informasi selengkapnya, lihat Mengautentikasi Permintaan (Versi AWS Tanda Tangan 4) dan proses penandatanganan Versi Tanda Tangan 4.

Tindakan

Untuk daftar lengkap izin dan kunci kondisi S3, lihat Kunci tindakan, sumber daya, dan kondisi untuk Amazon S3 di Referensi Otorisasi Layanan.

Untuk informasi selengkapnya tentang izin operasi S3 API menurut jenis sumber daya S3, lihat. Izin yang diperlukan untuk operasi API Amazon S3

Tindakan

Tindakan AWS Identity and Access Management (IAM) untuk Amazon S3 adalah kemungkinan tindakan yang dapat dilakukan pada bucket atau objek S3. Anda memberikan tindakan ini ke identitas sehingga mereka dapat bertindak berdasarkan sumber daya S3 Anda. Contoh tindakan S3 adalah `s3:GetObject` membaca objek dalam ember, dan `s3:PutObject` menulis objek ke ember.

Kunci syarat

Selain tindakan, kunci kondisi IAM dibatasi untuk memberikan akses hanya ketika suatu kondisi terpenuhi. Kunci kondisi bersifat opsional.

> **Note**
>
> Dalam kebijakan akses berbasis sumber daya, seperti kebijakan bucket, atau dalam kebijakan berbasis identitas, Anda dapat menentukan hal berikut:
>
> - Tindakan atau serangkaian tindakan dalam `Action` elemen pernyataan kebijakan.
> - Dalam `Effect` elemen pernyataan kebijakan, Anda dapat menentukan `Allow` untuk memberikan tindakan yang tercantum, atau Anda dapat menentukan `Deny` untuk memblokir tindakan yang tercantum. Untuk lebih mempertahankan praktik hak istimewa terkecil, `Deny` pernyataan dalam `Effect` elemen kebijakan akses harus seluas mungkin, dan `Allow` pernyataan harus sesempit mungkin. `Deny`efek yang dipasangkan dengan `s3:*` tindakan adalah cara lain yang baik untuk menerapkan praktik terbaik opt-in untuk identitas yang termasuk dalam pernyataan kondisi kebijakan.
> - Kunci kondisi dalam `Condition` elemen pernyataan kebijakan.

Kasus penggunaan manajemen akses

Amazon S3 menyediakan pemilik sumber daya dengan berbagai alat untuk memberikan akses. Alat manajemen akses S3 yang Anda gunakan bergantung pada sumber daya S3 yang ingin Anda bagikan, identitas yang Anda berikan akses, dan tindakan yang ingin Anda izinkan atau tolak. Anda mungkin ingin menggunakan satu atau kombinasi alat manajemen akses S3 untuk mengelola akses ke sumber daya S3 Anda.

Dalam kebanyakan kasus, Anda dapat menggunakan kebijakan akses untuk mengelola izin. Kebijakan akses dapat berupa kebijakan berbasis sumber daya, yang dilampirkan ke sumber daya, seperti bucket, atau sumber daya Amazon S3 lainnya (). Sumber daya S3 Kebijakan akses juga dapat berupa kebijakan berbasis identitas, yang dilampirkan ke pengguna, grup, atau peran AWS Identity and Access Management (IAM) di akun Anda. Anda mungkin menemukan bahwa kebijakan bucket berfungsi lebih baik untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat Kebijakan bucket untuk Amazon S3. Atau, dengan AWS Identity and Access Management (IAM), Anda dapat membuat pengguna, grup, dan peran IAM dalam diri Anda Akun AWS dan mengelola akses mereka ke bucket dan objek melalui kebijakan berbasis identitas. Untuk informasi selengkapnya, lihat Kebijakan berbasis identitas untuk Amazon S3.

Untuk membantu Anda menavigasi opsi manajemen akses ini, berikut ini adalah kasus penggunaan pelanggan Amazon S3 yang umum dan rekomendasi untuk masing-masing alat manajemen akses S3.

Akun AWS Pemilik ingin berbagi bucket hanya dengan pengguna dalam akun yang sama

Semua alat manajemen akses dapat memenuhi kasus penggunaan dasar ini. Kami merekomendasikan alat manajemen akses berikut untuk kasus penggunaan ini:

- Kebijakan bucket — Jika Anda ingin memberikan akses ke satu bucket atau sejumlah kecil bucket, atau jika izin akses bucket Anda serupa dari bucket ke bucket, gunakan kebijakan bucket. Dengan kebijakan bucket, Anda mengelola satu kebijakan untuk setiap bucket. Untuk informasi selengkapnya, lihat Kebijakan bucket untuk Amazon S3.
- Kebijakan berbasis identitas — Jika Anda memiliki jumlah bucket yang sangat besar dengan izin akses berbeda untuk setiap bucket, dan hanya beberapa peran pengguna yang harus dikelola, Anda dapat menggunakan kebijakan IAM untuk pengguna, grup, atau peran. Kebijakan IAM juga merupakan pilihan yang baik jika Anda mengelola akses pengguna ke AWS sumber daya lain,

serta sumber daya Amazon S3. Untuk informasi selengkapnya, lihat Contoh 1: Pemilik bucket yang memberikan izin bucket kepada penggunanya.

- Hibah Akses S3 — Anda dapat menggunakan Hibah Akses S3 untuk memberikan akses ke bucket, awalan, atau objek S3 Anda. S3 Access Grants memungkinkan Anda menentukan berbagai izin tingkat objek dalam skala besar; sedangkan, kebijakan bucket dibatasi hingga 20 KB. Untuk informasi selengkapnya, lihat Memulai S3 Access Grants.
- Access Points — Anda dapat menggunakan Access Points, yang diberi nama endpoint jaringan yang dilampirkan ke bucket. Bucket dapat memiliki hingga 10.000 titik akses yang terpasang, dan untuk setiap titik akses, Anda dapat menerapkan izin dan kontrol jaringan yang berbeda untuk memberi Anda kontrol terperinci atas akses ke objek S3 Anda. Untuk informasi selengkapnya, lihat Mengelola akses ke kumpulan data bersama dengan titik akses.

Akun AWS Pemilik ingin berbagi bucket atau objek dengan pengguna dari AWS akun lain (cross-account)

Untuk memberikan izin kepada orang lain Akun AWS, Anda harus menggunakan kebijakan bucket atau salah satu alat manajemen akses yang disarankan berikut ini. Anda tidak dapat menggunakan kebijakan akses berbasis identitas untuk kasus penggunaan ini. Untuk informasi selengkapnya tentang pemberian akses lintas akun, lihat Bagaimana cara menyediakan akses lintas akun ke objek yang ada di bucket Amazon S3?

Kami merekomendasikan alat manajemen akses berikut untuk kasus penggunaan ini:

- Kebijakan bucket — Dengan kebijakan bucket, Anda mengelola satu kebijakan untuk setiap bucket. Untuk informasi selengkapnya, lihat Kebijakan bucket untuk Amazon S3.
- Hibah Akses S3 — Anda dapat menggunakan Hibah Akses S3 untuk memberikan izin lintas akun ke bucket, awalan, atau objek S3 Anda. Anda dapat menggunakan S3 Access Grants untuk menentukan berbagai izin tingkat objek pada skala; sedangkan, kebijakan bucket dibatasi hingga 20 KB dalam ukuran. Untuk informasi selengkapnya, lihat Memulai S3 Access Grants.
- Access Points — Anda dapat menggunakan Access Points, yang diberi nama endpoint jaringan yang dilampirkan ke bucket. Bucket dapat memiliki hingga 10.000 titik akses yang terpasang, dan untuk setiap titik akses, Anda dapat menerapkan izin dan kontrol jaringan yang berbeda untuk memberi Anda kontrol terperinci atas akses ke objek S3 Anda. Untuk informasi selengkapnya, lihat Mengelola akses ke kumpulan data bersama dengan titik akses.

Akun AWS Pemilik atau pemilik bucket harus memberikan izin di tingkat objek atau tingkat awalan, dan izin ini bervariasi dari objek ke objek atau awalan ke awalan

Dalam kebijakan bucket, misalnya, Anda dapat memberikan akses ke objek dalam bucket yang membagikan awalan nama kunci tertentu atau memiliki tag tertentu. Anda dapat memberikan izin baca pada objek yang dimulai dengan awalan `logs/` nama kunci. Namun, jika izin akses Anda bervariasi menurut objek, pemberian izin ke objek individual dengan menggunakan kebijakan bucket mungkin tidak praktis, terutama karena kebijakan bucket dibatasi hingga 20 KB.

Kami merekomendasikan alat manajemen akses berikut untuk kasus penggunaan ini:

- Hibah Akses S3 - Anda dapat menggunakan Hibah Akses S3 untuk mengelola izin tingkat objek atau tingkat awalan. Tidak seperti kebijakan bucket, Anda dapat menggunakan S3 Access Grants untuk menentukan berbagai izin tingkat objek dalam skala besar. Kebijakan bucket dibatasi hingga ukuran 20 KB. Untuk informasi selengkapnya, lihat Memulai S3 Access Grants.
- Access Points — Anda dapat menggunakan titik akses untuk mengelola izin tingkat objek atau tingkat awalan. Access Points diberi nama endpoint jaringan yang dilampirkan ke bucket. Bucket dapat memiliki hingga 10.000 titik akses yang terpasang, dan untuk setiap titik akses, Anda dapat menerapkan izin dan kontrol jaringan yang berbeda untuk memberi Anda kontrol terperinci atas akses ke objek S3 Anda. Untuk informasi selengkapnya, lihat Mengelola akses ke kumpulan data bersama dengan titik akses.
- ACLs— Kami tidak menyarankan menggunakan Access Control Lists (ACLs), terutama karena ACLs dibatasi hingga 100 hibah per objek. Namun, jika Anda memilih untuk mengaktifkan ACLs, di Pengaturan Bucket, setel Kepemilikan Objek ke pemilik Bucket yang disukai dan ACLsdiaktifkan. Dengan pengaturan ini, objek baru yang ditulis dengan ACL terekam `bucket-owner-full-control` secara otomatis dimiliki oleh pemilik bucket daripada penulis objek. Anda kemudian dapat menggunakan objek ACLs, yang merupakan kebijakan akses berformat XML, untuk memberikan pengguna lain akses ke objek. Untuk informasi selengkapnya, lihat Gambaran umum daftar kontrol akses (ACL).

Akun AWS Pemilik atau pemilik bucket ingin membatasi akses bucket hanya ke akun tertentu IDs

Kami merekomendasikan alat manajemen akses berikut untuk kasus penggunaan ini:

- Kebijakan bucket — Dengan kebijakan bucket, Anda mengelola satu kebijakan untuk setiap bucket. Untuk informasi selengkapnya, lihat Kebijakan bucket untuk Amazon S3.

- Access Points — Access Points diberi nama endpoint jaringan yang dilampirkan ke bucket. Bucket dapat memiliki hingga 10.000 titik akses yang terpasang, dan untuk setiap titik akses, Anda dapat menerapkan izin dan kontrol jaringan yang berbeda untuk memberi Anda kontrol terperinci atas akses ke objek S3 Anda. Untuk informasi selengkapnya, lihat Mengelola akses ke kumpulan data bersama dengan titik akses.

Akun AWS Pemilik atau pemilik bucket menginginkan titik akhir yang berbeda untuk setiap pengguna atau aplikasi yang mengakses data mereka

Kami merekomendasikan alat manajemen akses berikut untuk kasus penggunaan ini:

- Access Points — Access Points diberi nama endpoint jaringan yang dilampirkan ke bucket. Bucket dapat memiliki hingga 10.000 titik akses yang terpasang, dan untuk setiap titik akses, Anda dapat menerapkan izin dan kontrol jaringan yang berbeda untuk memberi Anda kontrol terperinci atas akses ke objek S3 Anda. Setiap titik akses memberlakukan kebijakan titik akses khusus yang bekerja bersama kebijakan bucket yang melekat pada bucket dasar. Untuk informasi selengkapnya, lihat Mengelola akses ke kumpulan data bersama dengan titik akses.

Akun AWS Pemilik atau pemilik bucket harus mengelola akses dari titik akhir Virtual Private Cloud (VPC) untuk S3

Titik akhir Virtual Private Cloud (VPC) untuk Amazon S3 adalah entitas logis dalam VPC yang memungkinkan konektivitas hanya ke S3. Kami merekomendasikan alat manajemen akses berikut untuk kasus penggunaan ini:

- Bucket dalam pengaturan VPC — Anda dapat menggunakan kebijakan bucket untuk mengontrol siapa yang diizinkan mengakses bucket Anda dan titik akhir VPC mana yang dapat mereka akses. Untuk informasi selengkapnya, lihat Megendalikan akses dari titik akhir VPC dengan kebijakan bucket.
- Access Points — Jika Anda memilih untuk mengatur jalur akses, Anda dapat menggunakan kebijakan titik akses. Anda dapat mengonfigurasi titik akses apa pun untuk menerima permintaan hanya dari cloud privat virtual (VPC) untuk membatasi akses data Amazon S3 ke jaringan pribadi. Anda juga dapat mengonfigurasi pengaturan blok akses publik khusus untuk setiap titik akses. Untuk informasi selengkapnya, lihat Mengelola akses ke kumpulan data bersama dengan titik akses.

Akun AWS Pemilik atau pemilik bucket harus membuat situs web statis tersedia untuk umum

Dengan S3, Anda dapat meng-host situs web statis dan memungkinkan siapa saja untuk melihat konten situs web, yang di-host dari ember S3.

Kami merekomendasikan alat manajemen akses berikut untuk kasus penggunaan ini:

- Amazon CloudFront - Solusi ini memungkinkan Anda untuk meng-host situs web statis Amazon S3 ke publik sambil juga terus memblokir semua akses publik ke konten bucket. Jika Anda ingin mengaktifkan keempat pengaturan Akses Publik Blok S3 dan menghosting situs web statis S3, Anda dapat menggunakan kontrol akses CloudFront asal Amazon (OAC). Amazon CloudFront menyediakan kemampuan yang diperlukan untuk membuat situs web statis yang aman. Selain itu, situs web statis Amazon S3 yang tidak menggunakan solusi ini hanya dapat mendukung titik akhir HTTP. CloudFront menggunakan penyimpanan Amazon S3 yang tahan lama sambil menyediakan header keamanan tambahan, seperti HTTPS. HTTPS menambahkan keamanan dengan mengenkripsi permintaan HTTP normal, dan melindungi dari serangan siber umum.

  Untuk informasi selengkapnya, lihat Memulai situs web statis aman di Panduan CloudFront Pengembang Amazon.
- Membuat bucket Amazon S3 Anda dapat diakses publik — Anda dapat mengonfigurasi bucket untuk digunakan sebagai situs web statis yang diakses publik.

  > **Warning**
  >
  > Kami tidak merekomendasikan metode ini. Sebagai gantinya, kami sarankan Anda menggunakan situs web statis Amazon S3 sebagai bagian dari Amazon. CloudFront Untuk informasi selengkapnya, lihat opsi sebelumnya, atau lihat Memulai situs web statis yang aman.

  Untuk membuat situs web statis Amazon S3, tanpa Amazon CloudFront, pertama, Anda harus mematikan semua pengaturan Blokir Akses Publik. Saat menulis kebijakan bucket untuk situs web statis, pastikan Anda hanya mengizinkan tindakan `s3:GetObject`, bukan izin `ListObject` atau `PutObject`. Ini membantu memastikan bahwa pengguna tidak dapat melihat semua objek di bucket Anda atau menambahkan konten mereka sendiri. Untuk informasi selengkapnya, lihat Mengatur izin untuk akses situs web.

Akun AWS Pemilik atau pemilik ember ingin membuat konten ember tersedia untuk umum

Saat membuat bucket Amazon S3 baru, pengaturan Blokir Akses Publik diaktifkan secara default. Untuk informasi lebih lanjut tentang pemblokiran akses publik, lihat Melakukan blok akses publik ke penyimpanan Amazon S3 Anda.

Kami tidak menyarankan mengizinkan akses publik ke bucket Anda. Namun, jika Anda harus melakukannya untuk kasus penggunaan tertentu, kami merekomendasikan alat manajemen akses berikut untuk kasus penggunaan ini:

- Nonaktifkan pengaturan Blokir Akses Publik — Pemilik bucket dapat mengizinkan permintaan yang tidak diautentikasi ke bucket. Misalnya, permintaan Objek PUT yang tidak diautentikasi diizinkan saat bucket memiliki kebijakan bucket publik, atau saat bucket ACL memberikan akses publik. Semua permintaan yang tidak diautentikasi dibuat oleh pengguna sewenang-wenang lainnya, atau bahkan AWS pengguna anonim yang tidak diautentikasi. Pengguna ini diwakili ACLs oleh ID pengguna kanonik tertentu. `65a011a29cdf8ec533ec3d1ccaae921c` Jika objek diunggah ke `WRITE` atau`FULL_CONTROL`, maka ini secara khusus memberikan akses ke grup Semua Pengguna atau pengguna anonim. Untuk informasi selengkapnya tentang kebijakan bucket publik dan daftar kontrol akses publik (ACLs), lihatArti “publik”.

Akun AWS Pemilik atau pemilik bucket telah melampaui batas ukuran kebijakan akses

Kebijakan bucket dan kebijakan berbasis identitas memiliki batas ukuran 20 KB. Jika persyaratan izin akses Anda rumit, Anda mungkin melebihi batas ukuran ini.

Kami merekomendasikan alat manajemen akses berikut untuk kasus penggunaan ini:

- Titik Akses — Gunakan titik akses jika ini berfungsi dengan kasus penggunaan Anda. Dengan titik akses, setiap bucket memiliki beberapa titik akhir jaringan bernama, masing-masing dengan kebijakan jalur aksesnya sendiri yang sesuai dengan kebijakan bucket yang mendasarinya. Namun, titik akses hanya dapat bertindak pada objek, bukan ember, dan tidak mendukung replikasi lintas wilayah. Untuk informasi selengkapnya, lihat Mengelola akses ke kumpulan data bersama dengan titik akses.
- Hibah Akses S3 — Gunakan Hibah Akses S3, yang mendukung sejumlah besar hibah yang memberikan akses ke bucket, awalan, atau objek. Untuk informasi selengkapnya, lihat Memulai S3 Access Grants.

Peran Akun AWS pemilik atau admin ingin memberikan akses bucket, awalan, atau objek langsung ke pengguna atau grup di direktori perusahaan

Alih-alih mengelola pengguna, grup, dan peran melalui AWS Identity and Access Management (IAM), Anda dapat menambahkan direktori perusahaan Anda. AWS IAM Identity Center Untuk informasi lebih lanjut, lihat Apa itu Pusat Identitas IAM? .

Setelah Anda menambahkan direktori perusahaan Anda AWS IAM Identity Center, kami sarankan Anda menggunakan alat manajemen akses berikut untuk memberikan identitas direktori perusahaan akses ke sumber daya S3 Anda:

- Hibah Akses S3 — Gunakan Hibah Akses S3, yang mendukung pemberian akses ke pengguna atau peran di direktori perusahaan Anda. Untuk informasi selengkapnya, lihat Memulai S3 Access Grants.

Akun AWS Pemilik atau pemilik bucket ingin memberikan akses AWS CloudFront layanan untuk menulis CloudFront log ke bucket S3

Kami merekomendasikan alat manajemen akses berikut untuk kasus penggunaan ini:

- Bucket ACL — Satu-satunya kasus penggunaan yang disarankan untuk bucket ACLs adalah memberikan izin kepada tertentu Layanan AWS, seperti akun Amazon CloudFront `awslogsdelivery`. Saat Anda membuat atau memperbarui distribusi dan mengaktifkan CloudFront logging, CloudFront perbarui bucket ACL untuk memberikan `FULL_CONTROL` izin `awslogsdelivery` akun untuk menulis log ke bucket Anda. Untuk informasi selengkapnya, lihat Izin yang diperlukan untuk mengonfigurasi pencatatan standar dan mengakses file log Anda di Panduan CloudFront Pengembang Amazon. Jika bucket yang menyimpan log menggunakan pengaturan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek S3 untuk dimatikan ACLs, CloudFront tidak dapat menulis log ke bucket. Untuk informasi selengkapnya, lihat Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.

Anda, sebagai pemilik bucket, ingin mempertahankan kontrol penuh atas objek yang ditambahkan ke bucket oleh pengguna lain

Anda dapat memberikan akses akun lain untuk mengunggah objek ke bucket menggunakan kebijakan bucket, titik akses, atau S3 Access Grants. Jika Anda telah memberikan akses lintas akun

ke bucket, Anda dapat memastikan bahwa objek apa pun yang diunggah ke bucket tetap berada di bawah kendali penuh Anda.

Kami merekomendasikan alat manajemen akses berikut untuk kasus penggunaan ini:

- Kepemilikan Objek - Pertahankan pengaturan tingkat ember Kepemilikan Objek pada pengaturan default pemilik Bucket yang diberlakukan.

Pemecahan masalah manajemen akses

Sumber daya berikut dapat membantu Anda memecahkan masalah apa pun dengan manajemen akses S3:

Pemecahan Masalah Kesalahan Akses Ditolak (403 Terlarang)

Jika Anda mengalami masalah penolakan akses, periksa pengaturan tingkat akun dan tingkat ember. Juga, periksa fitur manajemen akses yang Anda gunakan untuk memberikan akses untuk memastikan bahwa kebijakan, pengaturan, atau konfigurasi sudah benar. Untuk informasi selengkapnya tentang penyebab umum kesalahan Akses Ditolak (403 Terlarang) di Amazon S3, lihat. Memecahkan masalah kesalahan akses ditolak (403 Forbidden) di Amazon S3

Penganalisis Akses IAM untuk S3

Jika Anda tidak ingin membuat sumber daya Anda tersedia untuk umum, atau jika Anda ingin membatasi akses publik ke sumber daya Anda, Anda dapat menggunakan IAM Access Analyzer untuk S3. Di konsol Amazon S3, gunakan IAM Access Analyzer untuk S3 untuk meninjau semua bucket yang memiliki daftar kontrol akses bucket (ACLs), kebijakan bucket, atau kebijakan titik akses yang memberikan akses publik atau bersama. IAM Access Analyzer for S3 memberi tahu Anda tentang bucket yang dikonfigurasi untuk memungkinkan akses ke siapa pun di internet atau lainnya Akun AWS, termasuk Akun AWS di luar organisasi Anda. Untuk setiap bucket publik atau bucket bersama, Anda akan menerima temuan yang melaporkan sumber dan tingkat akses publik atau akses bersama.

Di IAM Access Analyzer untuk S3, Anda dapat memblokir semua akses publik ke bucket dengan satu tindakan. Kami menyarankan Anda memblokir semua akses publik ke bucket Anda, kecuali Anda memerlukan akses publik untuk mendukung kasus penggunaan tertentu. Sebelum Anda memblokir semua akses publik, pastikan aplikasi Anda akan terus berfungsi dengan benar tanpa akses publik. Untuk informasi selengkapnya, lihat Melakukan blok akses publik ke penyimpanan Amazon S3 Anda.

Anda juga dapat meninjau pengaturan izin tingkat ember Anda untuk mengonfigurasi tingkat akses terperinci. Untuk kasus penggunaan tertentu dan terverifikasi yang memerlukan akses publik atau akses bersama, Anda dapat menyatakan dan mencatat maksud Anda untuk bucket agar tetap sebagai bucket dengan akses publik atau akses bersama dengan mengarsipkan temuan untuk bucket tersebut. Anda dapat mempertahankan dan memodifikasi konfigurasi bucket ini kapan saja. Anda juga dapat mengunduh temuan Anda dalam bentuk laporan CSV untuk keperluan audit.

IAM Access Analyzer untuk S3 tersedia tanpa biaya tambahan, yang berada di konsol Amazon S3. Penganalisis Akses IAM untuk S3 didukung oleh Penganalisis Akses IAM (IAM) AWS Identity and Access Management . Untuk menggunakan IAM Access Analyzer untuk S3 di konsol Amazon S3, Anda harus mengunjungi Konsol IAM dan membuat penganalisis tingkat akun di IAM Access Analyzer untuk setiap Wilayah.

Untuk informasi selengkapnya tentang IAM Access Analyzer untuk Amazon S3, lihat Meninjau akses bucket menggunakan IAM Access Analyzer untuk S3.

Pencatatan log dan pemantauan

Pemantauan adalah bagian penting dalam menjaga keandalan, ketersediaan, dan kinerja solusi Amazon S3 Anda sehingga Anda dapat lebih mudah men-debug kegagalan akses. Logging dapat memberikan wawasan tentang kesalahan apa pun yang diterima pengguna, dan kapan dan permintaan apa yang dibuat. AWS menyediakan beberapa alat untuk memantau sumber daya Amazon S3 Anda, seperti berikut ini:

- AWS CloudTrail
- Log Akses Amazon S3
- AWS Trusted Advisor
- Amazon CloudWatch

Untuk informasi selengkapnya, lihat Pencatatan log dan pemantauan di Amazon S3.

Identity and Access Management untuk Amazon S3

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diotorisasi (memiliki izin) untuk menggunakan sumber daya Amazon S3. IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Untuk informasi selengkapnya tentang izin operasi S3 API menurut jenis sumber daya S3, lihat. Izin yang diperlukan untuk operasi API Amazon S3

> **Note**
>
> Untuk informasi selengkapnya tentang menggunakan kelas penyimpanan Amazon S3 Express One Zone dengan bucket direktori, lihat S3 Express One Zone dan Bekerja dengan bucket direktori.

Topik

- Audiens
- Mengautentikasi dengan identitas
- Mengelola akses menggunakan kebijakan
- Bagaimana Amazon S3 bekerja dengan IAM
- Bagaimana Amazon S3 Mengotorisasi Permintaan
- Izin yang diperlukan untuk operasi API Amazon S3
- Kebijakan dan izin di Amazon S3
- Kebijakan bucket untuk Amazon S3
- Kebijakan berbasis identitas untuk Amazon S3
- Panduan yang menggunakan kebijakan untuk mengelola akses ke sumber daya Amazon S3
- Menggunakan Peran Terkait Layanan untuk Lensa Penyimpanan Amazon S3
- Memecahkan masalah identitas dan akses Amazon S3
- AWS kebijakan terkelola untuk Amazon S3

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan di Amazon S3.

Pengguna layanan - Jika Anda menggunakan layanan Amazon S3 untuk melakukan pekerjaan Anda, administrator Anda memberi Anda kredensi dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak fitur Amazon S3 untuk melakukan pekerjaan Anda, Anda mungkin memerlukan izin tambahan. Memahami cara akses dikelola dapat membantu Anda meminta izin yang tepat dari

administrator Anda. Jika Anda tidak dapat mengakses fitur di Amazon S3, lihat. Memecahkan masalah identitas dan akses Amazon S3

Administrator layanan - Jika Anda bertanggung jawab atas sumber daya Amazon S3 di perusahaan Anda, Anda mungkin memiliki akses penuh ke Amazon S3. Tugas Anda adalah menentukan fitur dan sumber daya Amazon S3 mana yang harus diakses pengguna layanan Anda. Kemudian, Anda harus mengirimkan permintaan kepada administrator IAM untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep dasar IAM. Untuk mempelajari lebih lanjut tentang bagaimana perusahaan Anda dapat menggunakan IAM dengan Amazon S3, lihat. Bagaimana Amazon S3 bekerja dengan IAM

Administrator IAM - Jika Anda administrator IAM, Anda mungkin ingin mempelajari detail tentang cara menulis kebijakan untuk mengelola akses ke Amazon S3. Untuk melihat contoh kebijakan berbasis identitas Amazon S3 yang dapat Anda gunakan di IAM, lihat. Kebijakan berbasis identitas untuk Amazon S3

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensyal identitas Anda. Anda harus diautentikasi (masuk ke AWS) sebagai Pengguna root akun AWS, sebagai pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk AWS sebagai identitas federasi dengan menggunakan kredensil yang disediakan melalui sumber identitas. AWS IAM Identity Center Pengguna (IAM Identity Center), autentikasi masuk tunggal perusahaan Anda, dan kredensyal Google atau Facebook Anda adalah contoh identitas federasi. Saat Anda masuk sebagai identitas terfederasi, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan peran IAM. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil peran.

Bergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal AWS akses. Untuk informasi selengkapnya tentang masuk AWS, lihat Cara masuk ke Panduan AWS Sign-In Pengguna Anda Akun AWS.

Jika Anda mengakses AWS secara terprogram, AWS sediakan kit pengembangan perangkat lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensil Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Guna mengetahui informasi selengkapnya tentang penggunaan metode yang disarankan untuk menandatangani permintaan sendiri, lihat AWS Signature Version 4 untuk permintaan API dalam Panduan Pengguna IAM.

Apa pun metode autentikasi yang digunakan, Anda mungkin diminta untuk menyediakan informasi keamanan tambahan. Misalnya, AWS merekomendasikan agar Anda menggunakan otentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari selengkapnya, lihat Autentikasi multi-faktor dalam Panduan Pengguna AWS IAM Identity Center dan Autentikasi multi-faktor AWS di IAM dalam Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya di akun. Identitas ini disebut pengguna Akun AWS root dan diakses dengan masuk dengan alamat email dan kata sandi yang Anda gunakan untuk membuat akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar lengkap tugas yang mengharuskan Anda masuk sebagai pengguna root, lihat Tugas yang memerlukan kredensial pengguna root dalam Panduan Pengguna IAM.

Identitas gabungan

Sebagai praktik terbaik, mewajibkan pengguna manusia, termasuk pengguna yang memerlukan akses administrator, untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS dengan menggunakan kredensi sementara.

Identitas federasi adalah pengguna dari direktori pengguna perusahaan Anda, penyedia identitas web, direktori Pusat Identitas AWS Directory Service, atau pengguna mana pun yang mengakses Layanan AWS dengan menggunakan kredensil yang disediakan melalui sumber identitas. Ketika identitas federasi mengakses Akun AWS, mereka mengambil peran, dan peran memberikan kredensi sementara.

Untuk manajemen akses terpusat, kami sarankan Anda menggunakan AWS IAM Identity Center. Anda dapat membuat pengguna dan grup di Pusat Identitas IAM, atau Anda dapat menghubungkan dan menyinkronkan ke sekumpulan pengguna dan grup di sumber identitas Anda sendiri untuk digunakan di semua aplikasi Akun AWS dan aplikasi Anda. Untuk informasi tentang Pusat Identitas IAM, lihat Apakah itu Pusat Identitas IAM? dalam Panduan Pengguna AWS IAM Identity Center .

Pengguna dan grup IAM

Pengguna IAM adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, kami merekomendasikan untuk mengandalkan kredensial

sementara, bukan membuat pengguna IAM yang memiliki kredensial jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan tertentu yang memerlukan kredensial jangka panjang dengan pengguna IAM, kami merekomendasikan Anda merotasi kunci akses. Untuk informasi selengkapnya, lihat Merotasi kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensial jangka panjang dalam Panduan Pengguna IAM.

Grup IAM adalah identitas yang menentukan sekumpulan pengguna IAM. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin bagi beberapa pengguna sekaligus. Grup mempermudah manajemen izin untuk sejumlah besar pengguna sekaligus. Misalnya, Anda dapat meminta kelompok untuk menyebutkan IAMAdmins dan memberikan izin kepada grup tersebut untuk mengelola sumber daya IAM.

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran dimaksudkan untuk dapat digunakan oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk mempelajari selengkapnya, lihat Kasus penggunaan untuk pengguna IAM dalam Panduan Pengguna IAM.

Peran IAM

Peran IAM adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus. Peran ini mirip dengan pengguna IAM, tetapi tidak terkait dengan orang tertentu. Untuk mengambil peran IAM sementara AWS Management Console, Anda dapat beralih dari pengguna ke peran IAM (konsol). Anda dapat mengambil peran dengan memanggil operasi AWS CLI atau AWS API atau dengan menggunakan URL kustom. Untuk informasi selengkapnya tentang cara menggunakan peran, lihat Metode untuk mengambil peran dalam Panduan Pengguna IAM.

Peran IAM dengan kredensial sementara berguna dalam situasi berikut:

- Akses pengguna terfederasi – Untuk menetapkan izin ke identitas terfederasi, Anda membuat peran dan menentukan izin untuk peran tersebut. Ketika identitas terfederasi mengautentikasi, identitas tersebut terhubung dengan peran dan diberi izin yang ditentukan oleh peran. Untuk informasi tentang peran untuk federasi, lihat Buat peran untuk penyedia identitas pihak ketiga dalam Panduan Pengguna IAM. Jika menggunakan Pusat Identitas IAM, Anda harus mengonfigurasi set izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah identitas tersebut diautentikasi, Pusat Identitas IAM akan mengorelasikan set izin ke peran dalam IAM. Untuk informasi tentang set izin, lihat Set izin dalam Panduan Pengguna AWS IAM Identity Center .
- Izin pengguna IAM sementara – Pengguna atau peran IAM dapat mengambil peran IAM guna mendapatkan berbagai izin secara sementara untuk tugas tertentu.

- Akses lintas akun – Anda dapat menggunakan peran IAM untuk mengizinkan seseorang (prinsipal tepercaya) di akun lain untuk mengakses sumber daya di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, dengan beberapa Layanan AWS, Anda dapat melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan peran sebagai proxy). Untuk mempelajari perbedaan antara peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Akses sumber daya lintas akun di IAM dalam Panduan Pengguna IAM.
- Akses lintas layanan — Beberapa Layanan AWS menggunakan fitur lain Layanan AWS. Misalnya, saat Anda melakukan panggilan dalam suatu layanan, biasanya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Sebuah layanan mungkin melakukannya menggunakan izin prinsipal yang memanggil, menggunakan peran layanan, atau peran terkait layanan.
  - Sesi akses teruskan (FAS) — Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat Sesi akses maju.
  - Peran layanan – Peran layanan adalah peran IAM yang dijalankan oleh layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Buat sebuah peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.
  - Peran terkait layanan — Peran terkait layanan adalah jenis peran layanan yang ditautkan ke peran layanan. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.
- Aplikasi yang berjalan di Amazon EC2 — Anda dapat menggunakan peran IAM untuk mengelola kredensyal sementara untuk aplikasi yang berjalan pada EC2 instance dan membuat AWS CLI atau AWS permintaan API. Ini lebih baik untuk menyimpan kunci akses dalam EC2 instance. Untuk menetapkan AWS peran ke EC2 instance dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instance berisi peran dan memungkinkan program yang berjalan pada EC2 instance untuk mendapatkan kredensyal

sementara. Untuk informasi selengkapnya, lihat Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan di EC2 instans Amazon di Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan adalah objek AWS yang, ketika dikaitkan dengan identitas atau sumber daya, menentukan izinnya. AWS mengevaluasi kebijakan ini ketika prinsipal (pengguna, pengguna root, atau sesi peran) membuat permintaan. Izin dalam kebijakan menentukan apakah permintaan diizinkan atau ditolak. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang struktur dan isi dokumen kebijakan JSON, lihat Gambaran umum kebijakan JSON dalam Panduan Pengguna IAM.

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk melakukan operasinya. Misalnya, anggaplah Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan tersebut bisa mendapatkan informasi peran dari AWS Management Console, API AWS CLI, atau AWS API.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan terkelola pelanggan dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan yang dikelola. Kebijakan inline disematkan langsung ke satu pengguna, grup, atau peran. Kebijakan

terkelola adalah kebijakan mandiri yang dapat dilampirkan ke beberapa pengguna, grup, dan peran dalam. Akun AWS Kebijakan AWS terkelola mencakup kebijakan terkelola dan kebijakan yang dikelola pelanggan. Untuk mempelajari cara memilih antara kebijakan yang dikelola atau kebijakan inline, lihat Pilih antara kebijakan yang dikelola dan kebijakan inline dalam Panduan Pengguna IAM.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan prinsipal dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola dari IAM dalam kebijakan berbasis sumber daya.

Daftar kontrol akses (ACLs)

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan JSON.

Amazon S3, AWS WAF, dan VPC adalah contoh layanan yang mendukung. ACLs Untuk mempelajari selengkapnya ACLs, lihat Ringkasan daftar kontrol akses (ACL) di Panduan Pengembang Layanan Penyimpanan Sederhana Amazon.

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang kurang umum. Jenis-jenis kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda oleh jenis kebijakan yang lebih umum.

- Batasan izin – Batasan izin adalah fitur lanjutan tempat Anda mengatur izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas ke entitas IAM (pengguna IAM atau peran IAM). Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas milik entitas dan batasan izinnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang `Principal` tidak dibatasi oleh batasan izin. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian

izin. Untuk informasi selengkapnya tentang batasan izin, lihat Batasan izin untuk entitas IAM dalam Panduan Pengguna IAM.

- Kebijakan kontrol layanan (SCPs) — SCPs adalah kebijakan JSON yang menentukan izin maksimum untuk organisasi atau unit organisasi (OU) di. AWS Organizations AWS Organizations adalah layanan untuk mengelompokkan dan mengelola secara terpusat beberapa Akun AWS yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur dalam suatu organisasi, maka Anda dapat menerapkan kebijakan kontrol layanan (SCPs) ke salah satu atau semua akun Anda. SCP membatasi izin untuk entitas di akun anggota, termasuk masing-masing. Pengguna root akun AWS Untuk informasi selengkapnya tentang Organizations dan SCPs, lihat Kebijakan kontrol layanan di Panduan AWS Organizations Pengguna.
- Kebijakan kontrol sumber daya (RCPs) — RCPs adalah kebijakan JSON yang dapat Anda gunakan untuk menetapkan izin maksimum yang tersedia untuk sumber daya di akun Anda tanpa memperbarui kebijakan IAM yang dilampirkan ke setiap sumber daya yang Anda miliki. RCP membatasi izin untuk sumber daya di akun anggota dan dapat memengaruhi izin efektif untuk identitas, termasuk Pengguna root akun AWS, terlepas dari apakah itu milik organisasi Anda. Untuk informasi selengkapnya tentang Organizations dan RCPs, termasuk daftar dukungan Layanan AWS tersebut RCPs, lihat Kebijakan kontrol sumber daya (RCPs) di Panduan AWS Organizations Pengguna.
- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda berikan sebagai parameter ketika Anda membuat sesi sementara secara programatis untuk peran atau pengguna terfederasi. Izin sesi yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga bisa datang dari kebijakan berbasis sumber daya. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya, lihat Kebijakan sesi dalam Panduan Pengguna IAM.

Berbagai jenis kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat Logika evaluasi kebijakan di Panduan Pengguna IAM.

Bagaimana Amazon S3 bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses ke Amazon S3, pelajari fitur IAM apa yang tersedia untuk digunakan dengan Amazon S3.

Fitur IAM yang dapat Anda gunakan dengan Amazon S3

Fitur IAM	Dukungan Amazon S3
Kebijakan berbasis identitas	Ya
Kebijakan berbasis sumber daya	Ya
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya
kunci-kunci persyaratan kebijakan (spesifik layanan)	Ya
ACLs	Ya
ABAC (tanda dalam kebijakan)	Parsial
Kredensial sementara	Ya
Sesi akses teruskan (FAS)	Ya
Peran layanan	Ya
Peran terkait layanan	Parsial

Untuk mendapatkan tampilan tingkat tinggi tentang cara kerja Amazon S3 dan layanan AWS lainnya dengan sebagian besar fitur IAM, AWS lihat layanan yang bekerja dengan IAM di Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang izin operasi S3 API menurut jenis sumber daya S3, lihat. Izin yang diperlukan untuk operasi API Amazon S3

Kebijakan berbasis identitas untuk Amazon S3

Mendukung kebijakan berbasis identitas: Ya

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana,

dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan terkelola pelanggan dalam Panduan Pengguna IAM.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan secara spesifik apakah tindakan dan sumber daya diizinkan atau ditolak, serta kondisi yang menjadi dasar dikabulkan atau ditolaknya tindakan tersebut. Anda tidak dapat menentukan secara spesifik prinsipal dalam sebuah kebijakan berbasis identitas karena prinsipal berlaku bagi pengguna atau peran yang melekat kepadanya. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam kebijakan JSON, lihat Referensi elemen kebijakan JSON IAM dalam Panduan Pengguna IAM.

Contoh kebijakan berbasis identitas untuk Amazon S3

Untuk melihat contoh kebijakan berbasis identitas Amazon S3, lihat. Kebijakan berbasis identitas untuk Amazon S3

Kebijakan berbasis sumber daya dalam Amazon S3

Mendukung kebijakan berbasis sumber daya: Ya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan prinsipal dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan secara spesifik seluruh akun atau entitas IAM di akun lain sebagai prinsipal dalam kebijakan berbasis sumber daya. Menambahkan prinsipal akun silang ke kebijakan berbasis sumber daya hanya setengah dari membangun hubungan kepercayaan. Ketika prinsipal dan sumber daya berbeda Akun AWS, administrator IAM di akun tepercaya juga harus memberikan izin entitas utama (pengguna atau peran) untuk mengakses sumber daya. Mereka memberikan izin dengan melampirkan kebijakan berbasis identitas kepada entitas. Namun, jika kebijakan berbasis sumber daya memberikan akses ke principal dalam akun yang sama, tidak diperlukan kebijakan berbasis identitas tambahan. Untuk informasi selengkapnya, lihat Akses sumber daya lintas akun di IAM dalam Panduan Pengguna IAM.

Layanan Amazon S3 mendukung kebijakan bucket, kebijakan titik akses, dan hibah akses:

- Kebijakan bucket adalah kebijakan berbasis sumber daya yang dilampirkan ke bucket Amazon S3. Kebijakan bucket menentukan prinsipal mana yang dapat melakukan tindakan di bucket.
- Kebijakan titik akses adalah kebijakan berbasis sumber daya yang dievaluasi bersama dengan kebijakan bucket yang mendasarinya.
- Hibah akses adalah model yang disederhanakan untuk menentukan izin akses ke data di Amazon S3 berdasarkan awalan, bucket, atau objek. Untuk informasi tentang Hibah Akses S3, lihat. Mengelola akses dengan S3 Access Grants

Prinsipal untuk kebijakan bucket

`Principal`Elemen menentukan pengguna, akun, layanan, atau entitas lain yang diizinkan atau ditolak akses ke sumber daya. Berikut ini adalah contoh-contoh menentukan `Principal`. Untuk informasi selengkapnya, lihat Pengguna Utama dalam Panduan Pengguna IAM.

Berikan izin ke Akun AWS

Untuk memberikan izin ke Akun AWS, identifikasi akun menggunakan format berikut.

```
"AWS":"account-ARN"
```

Berikut ini adalah beberapa contohnya.

```
"Principal":{"AWS":"arn:aws:iam::AccountIDWithoutHyphens:root"}
```

```
"Principal":{"AWS":
["arn:aws:iam::AccountID1WithoutHyphens:root","arn:aws:iam::AccountID2WithoutHyphens:root"]}
```

> **Note**
>
> Contoh di atas memberikan izin kepada pengguna root, yang mendelegasikan izin ke tingkat akun. Namun, kebijakan IAM masih diperlukan pada peran dan pengguna tertentu di akun.

Berikan Izin kepada Pengguna IAM

Untuk memberikan izin kepada pengguna IAM dalam akun Anda, Anda harus memberikan pasangan nama-nilai `"AWS":"user-ARN"`.

```
"Principal":{"AWS":"arn:aws:iam::account-number-without-hyphens:user/username"}
```

Untuk contoh rinci yang memberikan step-by-step instruksi, lihat Contoh 1: Pemilik bucket yang memberikan izin bucket kepada penggunanya dan Contoh 3: Pemilik bucket yang memberikan izin kepada penggunanya ke objek yang bukan miliknya.

> **Note**
>
> Jika identitas IAM dihapus setelah Anda memperbarui kebijakan bucket, kebijakan bucket akan menampilkan pengenal unik di elemen utama, bukan ARN. Keunikan ini tidak pernah IDs digunakan kembali, sehingga Anda dapat dengan aman menghapus prinsipal dengan pengenal unik dari semua pernyataan kebijakan Anda. Untuk informasi selengkapnya tentang pengidentifikasi unik, lihat pengidentifikasi IAM di Panduan Pengguna IAM.

Memberikan izin anonim

> **Warning**
>
> Berhati-hatilah saat memberikan akses anonim ke bucket Amazon S3. Saat Anda memberikan akses anonim, siapa pun di dunia dapat mengakses bucket Anda. Kami sangat menyarankan agar Anda tidak pernah memberikan akses menulis anonim apa pun ke bucket S3 Anda.

Untuk memberikan izin kepada semua orang, juga disebut sebagai akses anonim, Anda atur wildcard (`"*"`) sebagai nilai `Principal`. Misalnya, jika Anda mengonfigurasi bucket sebagai situs web, maka Anda ingin semua objek yang ada dalam bucket dapat diakses publik.

```
"Principal":"*"
```

```
"Principal":{"AWS":"*"}
```

Menggunakan `"Principal": "*"` dengan `Allow` efek dalam kebijakan berbasis sumber daya memungkinkan siapa pun, meskipun mereka tidak masuk AWS, untuk mengakses sumber daya Anda.

Menggunakan `"Principal" : { "AWS" : "*" }` dengan efek `Allow` dalam kebijakan berbasis sumber daya memungkinkan setiap pengguna root, pengguna IAM, sesi peran yang dianggap, atau pengguna gabungan di akun apa pun di dalam partisi yang sama untuk mengakses sumber daya Anda.

Untuk pengguna anonim, kedua metode ini setara. Untuk informasi selengkapnya, lihat Semua pengguna utama di Panduan Pengguna IAM.

Anda tidak dapat menggunakan wildcard untuk mencocokkan sebagian nama pengguna utama atau ARN.

> **Important**
>
> Dalam kebijakan kontrol AWS akses, Prinsipal “*” dan {”AWS“: “*"} berperilaku identik.

Batasi izin sumber daya

Anda juga dapat menggunakan kebijakan sumber daya untuk membatasi akses ke sumber daya yang seharusnya tersedia untuk pengguna utama IAM. Gunakan `Deny` pernyataan untuk mencegah akses.

Contoh berikut memblokir akses jika protokol transport aman tidak digunakan:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyBucketAccessIfSTPNotUsed",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
      "Condition": {
        "Bool": {
          "aws:SecureTransport": "false"
        }
      }
    }
  ]
}
```

Menggunakan `"Principal": "*"` sehingga pembatasan ini berlaku untuk semua orang adalah praktik terbaik untuk kebijakan ini, daripada mencoba menolak akses hanya ke akun atau pengguna utama tertentu yang menggunakan metode ini.

Membutuhkan akses melalui CloudFront URLs

Anda dapat meminta pengguna mengakses konten Amazon S3 Anda hanya dengan menggunakan CloudFront URLs alih-alih Amazon S3. URLs Untuk melakukan ini, buat kontrol akses CloudFront asal (OAC). Kemudian, ubah izin pada data S3 Anda. Dalam kebijakan bucket, Anda dapat menetapkan CloudFront Principal sebagai berikut:

```
"Principal":{"Service":"cloudfront.amazonaws.com"}
```

Gunakan `Condition` elemen dalam kebijakan CloudFront untuk mengizinkan akses bucket hanya jika permintaan tersebut atas nama CloudFront distribusi yang berisi asal S3.

```
        "Condition": {
           "StringEquals": {
              "AWS:SourceArn":
 "arn:aws:cloudfront::111122223333:distribution/CloudFront-distribution-ID"
           }
        }
```

Untuk informasi selengkapnya tentang mewajibkan akses S3 CloudFront URLs, lihat Membatasi akses ke asal Layanan Penyimpanan Sederhana Amazon di Panduan CloudFront Pengembang Amazon. Untuk informasi selengkapnya tentang manfaat keamanan dan privasi menggunakan Amazon CloudFront, lihat Mengonfigurasi akses aman dan membatasi akses ke konten.

Contoh kebijakan berbasis sumber daya untuk Amazon S3

- Untuk melihat contoh kebijakan untuk bucket Amazon S3, lihat. Kebijakan bucket untuk Amazon S3
- Untuk melihat contoh kebijakan untuk titik akses, lihatMengonfigurasi kebijakan IAM untuk menggunakan titik akses.

Tindakan kebijakan untuk Amazon S3

Mendukung tindakan kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama dengan operasi AWS API terkait. Ada beberapa pengecualian, misalnya tindakan hanya izin yang tidak memiliki operasi API yang cocok. Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam suatu kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Sertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Berikut ini menunjukkan berbagai jenis hubungan pemetaan antara operasi API S3 dan tindakan kebijakan yang diperlukan.

- One-to-one pemetaan dengan nama yang sama. Misalnya, untuk menggunakan operasi `PutBucketPolicy` API, tindakan `s3:PutBucketPolicy` kebijakan diperlukan.
- One-to-one pemetaan dengan nama yang berbeda. Misalnya, untuk menggunakan operasi `ListObjectsV2` API, tindakan `s3:ListBucket` kebijakan diperlukan.
- One-to-many pemetaan. Misalnya, untuk menggunakan operasi `HeadObject` API, `s3:GetObject` diperlukan. Selain itu, saat Anda menggunakan Kunci Objek S3 dan ingin mendapatkan status Penahanan Hukum atau pengaturan penyimpanan objek, tindakan terkait `s3:GetObjectLegalHold` atau `s3:GetObjectRetention` kebijakan juga diperlukan sebelum Anda dapat menggunakan operasi `HeadObject` API.
- Many-to-one pemetaan. Misalnya, untuk menggunakan operasi `ListObjectsV2` atau `HeadBucket` API, tindakan `s3:ListBucket` kebijakan diperlukan.

Untuk melihat daftar tindakan Amazon S3 untuk digunakan dalam kebijakan, lihat Tindakan yang ditentukan oleh Amazon S3 di Referensi Otorisasi Layanan. Untuk daftar lengkap operasi Amazon S3 API, lihat Tindakan API Amazon S3 di Referensi API Layanan Penyimpanan Sederhana Amazon.

Untuk informasi selengkapnya tentang izin operasi S3 API menurut jenis sumber daya S3, lihat. Izin yang diperlukan untuk operasi API Amazon S3

Tindakan kebijakan di Amazon S3 menggunakan awalan berikut sebelum tindakan:

```
s3
```

Untuk menetapkan secara spesifik beberapa tindakan dalam satu pernyataan, pisahkan tindakan tersebut dengan koma.

```
"Action": [
      "s3:action1",
      "s3:action2"
         ]
```

Operasi bucket

Operasi bucket adalah operasi API S3 yang beroperasi pada tipe sumber daya bucket. Misalnya,`CreateBucket`,`ListObjectsV2`, dan`PutBucketPolicy`. Tindakan kebijakan S3 untuk operasi bucket memerlukan `Resource` elemen dalam kebijakan bucket atau kebijakan berbasis identitas IAM sebagai pengidentifikasi Amazon Resource Name (ARN) tipe bucket S3 dalam format contoh berikut.

```
"Resource": "arn:aws:s3:::amzn-s3-demo-bucket"
```

Kebijakan bucket berikut *Akua* memberi pengguna `s3:ListBucket` izin akun *12345678901* untuk melakukan operasi ListObjectsV2API dan mencantumkan objek dalam bucket S3.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow Akua to list objects in the bucket",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::12345678901:user/Akua"
            },
            "Action": [
            "s3:ListBucket"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket"
        }
    ]
}
```

Operasi bucket dalam kebijakan untuk titik akses untuk bucket tujuan umum

Izin yang diberikan di jalur akses untuk kebijakan bucket tujuan umum hanya berlaku jika bucket yang mendasarinya mengizinkan izin yang sama. Saat Anda menggunakan Titik Akses S3, Anda harus mendelegasikan kontrol akses dari bucket ke titik akses atau menambahkan izin yang sama dalam kebijakan titik akses ke kebijakan bucket yang mendasarinya. Untuk informasi selengkapnya, lihat Mengonfigurasi kebijakan IAM untuk menggunakan titik akses. Dalam kebijakan titik akses, tindakan kebijakan S3 untuk operasi bucket mengharuskan Anda menggunakan titik akses ARN untuk elemen `Resource` dalam format berikut.

```
"Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/example-access-point"
```

Kebijakan titik akses berikut *Akua* memberi pengguna akun *12345678901* `s3:ListBucket` izin untuk melakukan operasi ListObjectsV2API melalui titik akses S3 bernama. *example-access-point* Izin ini memungkinkan *Akua* untuk membuat daftar objek di bucket yang terkait dengannya*example-access-point*.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow Akua to list objects in the bucket through access point",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::12345678901:user/Akua"
            },
            "Action": [
            "s3:ListBucket"
            ],
            "Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/example-access-
point"
        }
    ]
}
```

> **Note**
>
> Tidak semua operasi bucket didukung oleh titik akses untuk bucket tujuan umum. Untuk informasi selengkapnya, lihat Kompatibilitas titik akses dengan operasi S3.

Operasi bucket dalam kebijakan untuk titik akses untuk bucket direktori

Izin yang diberikan dalam jalur akses untuk kebijakan bucket direktori hanya efektif jika bucket yang mendasarinya mengizinkan izin yang sama. Saat Anda menggunakan Titik Akses S3, Anda harus mendelegasikan kontrol akses dari bucket ke titik akses atau menambahkan izin yang sama dalam kebijakan titik akses ke kebijakan bucket yang mendasarinya. Untuk informasi selengkapnya, lihat Mengkonfigurasi kebijakan IAM untuk menggunakan titik akses untuk bucket direktori. Dalam kebijakan titik akses, tindakan kebijakan S3 untuk operasi bucket mengharuskan Anda menggunakan titik akses ARN untuk elemen `Resource` dalam format berikut.

```
"Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/example-access-point--usw2-az1--xa-s3"
```

Kebijakan titik akses berikut *Akua* memberi pengguna akun *12345678901* `s3:ListBucket` izin untuk melakukan operasi ListObjectsV2API melalui titik akses bernama*example-access-point--usw2-az1--xa-s3*. Izin ini memungkinkan *Akua* untuk membuat daftar objek di bucket yang terkait dengannya*example-access-point--usw2-az1--xa-s3*.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow Akua to list objects in the bucket through access point",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::12345678901:user/Akua"
            },
            "Action": [
            "s3:ListBucket"
            ],
            "Resource": "arn:aws:s3express:us-west-2:123456789012:accesspoint/example-access-point--usw2-az1--xa-s3"
        }
    ]
}
```

> **Note**
>
> Tidak semua operasi bucket didukung oleh titik akses untuk bucket direktori. Untuk informasi selengkapnya, lihat Operasi objek untuk titik akses untuk bucket direktori.

Operasi objek

Operasi objek adalah operasi API S3 yang bertindak berdasarkan jenis sumber daya objek. Misalnya,`GetObject`,`PutObject`, dan`DeleteObject`. Tindakan kebijakan S3 untuk operasi objek memerlukan `Resource` elemen dalam kebijakan menjadi objek ARN S3 dalam format contoh berikut.

```
"Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
```

```
"Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/*"
```

> **Note**
>
> Objek ARN harus berisi garis miring ke depan setelah nama bucket, seperti yang terlihat pada contoh sebelumnya.

Kebijakan bucket berikut *Akua* memberi pengguna `s3:PutObject` izin akun*12345678901*. Izin ini memungkinkan *Akua* untuk menggunakan operasi PutObjectAPI untuk mengunggah objek ke bucket S3 bernama*amzn-s3-demo-bucket*.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow Akua to upload objects",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::12345678901:user/Akua"
            },
            "Action": [
            "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
        }
    ]
}
```

Operasi objek dalam kebijakan titik akses

Bila Anda menggunakan S3 Access Points untuk mengontrol akses ke operasi objek, Anda dapat menggunakan kebijakan titik akses. Saat Anda menggunakan kebijakan titik akses, tindakan kebijakan S3 untuk operasi objek mengharuskan Anda menggunakan titik akses ARN untuk elemen `Resource` dalam format berikut:. `arn:aws:s3:region:account-id:accesspoint/access-point-name/object/resource` Untuk operasi objek yang menggunakan titik akses, Anda harus menyertakan `/object/` nilai setelah seluruh titik akses ARN dalam elemen. `Resource` Berikut ini adalah beberapa contoh.

```
"Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/example-access-point/object/
*"
```

```
"Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/example-access-point/
object/prefix/*"
```

Kebijakan jalur akses berikut *Akua* memberi pengguna `s3:GetObject` izin akun*12345678901*. Izin ini memungkinkan *Akua* untuk melakukan operasi GetObjectAPI melalui titik akses yang diberi nama *example-access-point* pada semua objek di bucket yang terkait dengan titik akses.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow Akua to get objects through access point",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::12345678901:user/Akua"
            },
            "Action": [
            "s3:GetObject"
            ],
            "Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/example-access-
point/object/*"
        }
    ]
}
```

> **Note**
>
> Tidak semua operasi objek didukung oleh titik akses. Untuk informasi selengkapnya, lihat Kompatibilitas titik akses dengan operasi S3.

Operasi objek dalam kebijakan untuk titik akses untuk bucket direktori

Saat Anda menggunakan titik akses untuk bucket direktori untuk mengontrol akses ke operasi objek, Anda dapat menggunakan kebijakan titik akses. Saat Anda menggunakan kebijakan titik akses, tindakan kebijakan S3 untuk operasi objek mengharuskan Anda menggunakan titik akses ARN untuk elemen `Resource` dalam format berikut:. `arn:aws:s3:region:account-id:accesspoint/access-point-name/object/resource` Untuk operasi objek yang menggunakan titik akses, Anda harus menyertakan `/object/` nilai setelah seluruh titik akses ARN dalam elemen. `Resource` Berikut ini adalah beberapa contoh.

```
"Resource": "arn:aws:s3express:us-west-2:123456789012:accesspoint/example-access-point--usw2-az1--xa-s3/object/*"
```

```
"Resource": "arn:aws:s3express:us-west-2:123456789012:accesspoint/example-access-point--usw2-az1--xa-s3/object/prefix/*"
```

Kebijakan jalur akses berikut *Akua* memberi pengguna `s3:GetObject` izin akun*12345678901*. Izin ini memungkinkan *Akua* untuk melakukan operasi GetObjectAPI melalui titik akses yang diberi nama *example-access-point--usw2-az1--xa-s3* pada semua objek di bucket yang terkait dengan titik akses.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow Akua to get objects through access point",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::12345678901:user/Akua"
            },
            "Action": "s3express:CreateSession","s3:GetObject"
            "Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/example-access-point--usw2-az1--xa-s3/object/*"
        }
```

```
    ]
}
```

> **Note**
>
> Tidak semua operasi objek didukung oleh titik akses untuk bucket direktori. Untuk informasi selengkapnya, lihat Operasi objek untuk titik akses untuk bucket direktori.

Titik akses untuk operasi bucket tujuan umum

Operasi titik akses adalah operasi API S3 yang beroperasi pada jenis `accesspoint` sumber daya. Misalnya,`CreateAccessPoint`,`DeleteAccessPoint`, dan`GetAccessPointPolicy`. Tindakan kebijakan S3 untuk operasi titik akses hanya dapat digunakan dalam kebijakan berbasis identitas IAM, bukan dalam kebijakan bucket atau kebijakan jalur akses. Operasi titik akses memerlukan `Resource` elemen untuk menjadi titik akses ARN dalam format contoh berikut.

```
"Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/example-access-point"
```

Kebijakan berbasis identitas IAM berikut memberikan `s3:GetAccessPointPolicy` izin untuk melakukan operasi GetAccessPointPolicyAPI pada titik akses S3 bernama. *`example-access-point`*

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Grant permission to retrieve the access point policy of access
 point example-access-point",
            "Effect": "Allow",
            "Action": [
            "s3:GetAccessPointPolicy"
            ],
            "Resource": "arn:aws:s3:*:123456789012:accesspoint/example-access-point"
        }
    ]
}
```

Saat Anda menggunakan Titik Akses, untuk mengontrol akses ke operasi bucket, lihatOperasi bucket dalam kebijakan untuk titik akses untuk bucket tujuan umum; untuk mengontrol akses ke operasi

objek, lihatOperasi objek dalam kebijakan titik akses. Untuk informasi selengkapnya tentang cara mengonfigurasi kebijakan titik akses, lihatMengonfigurasi kebijakan IAM untuk menggunakan titik akses.

Titik akses untuk operasi bucket direktori

Titik akses untuk operasi bucket direktori adalah operasi API S3 yang beroperasi pada jenis `accesspoint` sumber daya. Misalnya,`CreateAccessPoint`,`DeleteAccessPoint`, dan`GetAccessPointPolicy`. Tindakan kebijakan S3 untuk operasi titik akses hanya dapat digunakan dalam kebijakan berbasis identitas IAM, bukan dalam kebijakan bucket atau kebijakan jalur akses. Titik akses untuk operasi bucket direktori memerlukan `Resource` elemen untuk menjadi titik akses ARN dalam format contoh berikut.

```
"Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/example-access-point--usw2-az1--xa-s3"
```

Kebijakan berbasis identitas IAM berikut memberikan `s3express:GetAccessPointPolicy` izin untuk melakukan operasi GetAccessPointPolicyAPI pada titik akses bernama. *`example-access-point--usw2-az1--xa-s3`*

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Grant permission to retrieve the access point policy of access
 point example-access-point--usw2-az1--xa-s3",
            "Effect": "Allow",
            "Action": [
            "s3express:CreateSession","s3express:GetAccessPointPolicy"
            ],
            "Resource": "arn:aws:s3:*:123456789012:accesspoint/example-access-point--
usw2-az1--xa-s3"
        }
    ]
}
```

Kebijakan berbasis identitas IAM berikut memberikan `s3express:CreateAccessPoint` izin untuk membuat titik akses untuk bucket direktori.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Grant CreateAccessPoint.",
            "Principal": "*",
            "Action": "s3express:CreateSession",
            "s3express:CreateAccessPoint""Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Kebijakan berbasis identitas IAM berikut memberikan `s3express:PutAccessPointScope` izin untuk membuat cakupan titik akses untuk titik akses untuk bucket direktori.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Grant PutAccessPointScope",
            "Principal": "*",
            "Action": "s3express:CreateSession",
            "s3express:CreateAccessPoint",
            "S3Express:PutAccessPointScope""Effect": "Allow",
            "Resource": "*",
        }
    ]
}
```

Saat Anda menggunakan titik akses untuk bucket direktori untuk mengontrol akses ke operasi bucket, lihatOperasi bucket dalam kebijakan untuk titik akses untuk bucket direktori; untuk mengontrol akses ke operasi objek, lihatOperasi objek dalam kebijakan untuk titik akses untuk bucket direktori. Untuk informasi selengkapnya tentang cara mengonfigurasi titik akses untuk kebijakan bucket direktori, lihatMengkonfigurasi kebijakan IAM untuk menggunakan titik akses untuk bucket direktori.

Operasi Objek Lambda Access Point

Dengan Lambda Objek Amazon S3, Anda dapat menambahkan kode Anda sendiri ke `GET`, `LIST`, dan `HEAD` Amazon S3 dan meminta untuk mengubah dan memproses data saat dikembalikan ke

aplikasi. Anda dapat membuat permintaan melalui Object Lambda Access Point, yang berfungsi sama seperti membuat permintaan melalui titik akses lainnya. Untuk informasi selengkapnya, lihat Mengubah objek dengan S3 Lambda Objek.

Untuk informasi selengkapnya tentang cara mengonfigurasi kebijakan untuk operasi Titik Akses Objek Lambda, lihat. Mengonfigurasi kebijakan IAM untuk Titik Akses Lambda Objek

Operasi Titik Akses Multi-Wilayah

Titik Akses Multi-Region menyediakan titik akhir global yang dapat digunakan aplikasi untuk memenuhi permintaan dari bucket S3 yang terletak di beberapa. Wilayah AWS Anda dapat menggunakan Titik Akses Multi-Wilayah untuk membangun aplikasi Multi-wilayah dengan arsitektur yang sama yang digunakan di satu Wilayah, dan kemudian menjalankan aplikasi tersebut di mana saja di dunia. Untuk informasi selengkapnya, lihat Mengelola lalu lintas Multi-wilayah dengan Titik Akses Multi-Wilayah.

Untuk informasi selengkapnya tentang cara mengonfigurasi kebijakan untuk operasi Titik Akses Multi-Wilayah, lihatContoh kebijakan Titik Akses Multi-Wilayah.

Operasi pekerjaan Batch

Operasi pekerjaan (Operasi Batch) adalah operasi API S3 yang beroperasi pada jenis sumber daya pekerjaan. Misalnya, `DescribeJob` dan `CreateJob`. Tindakan kebijakan S3 untuk operasi pekerjaan hanya dapat digunakan dalam kebijakan berbasis identitas IAM, bukan dalam kebijakan bucket. Selain itu, operasi pekerjaan memerlukan `Resource` elemen dalam kebijakan berbasis identitas IAM menjadi `job` ARN dalam format contoh berikut.

```
"Resource": "arn:aws:s3:*:123456789012:job/*"
```

Kebijakan berbasis identitas IAM berikut memberikan `s3:DescribeJob` izin untuk melakukan operasi DescribeJobAPI pada pekerjaan Operasi Batch S3 yang diberi nama. *`example-job`*

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow describing the Batch operation job example-job",
            "Effect": "Allow",
            "Action": [
            "s3:DescribeJob"
            ],
```

```
            "Resource": "arn:aws:s3:*:123456789012:job/example-job"
        }
    ]
}
```

Operasi konfigurasi Lensa Penyimpanan S3

Untuk informasi selengkapnya tentang cara mengonfigurasi operasi konfigurasi Lensa Penyimpanan S3, lihatMenyetel izin Lensa Penyimpanan Amazon S3.

Operasi akun

Operasi akun adalah operasi API S3 yang beroperasi pada tingkat akun. Misalnya, `GetPublicAccessBlock` (untuk akun). Akun bukanlah jenis sumber daya yang ditentukan oleh Amazon S3. Tindakan kebijakan S3 untuk operasi akun hanya dapat digunakan dalam kebijakan berbasis identitas IAM, bukan dalam kebijakan bucket. Selain itu, operasi akun memerlukan `Resource` elemen dalam kebijakan berbasis identitas IAM. `"*"`

Kebijakan berbasis identitas IAM berikut memberikan `s3:GetAccountPublicAccessBlock` izin untuk melakukan operasi GetPublicAccessBlockAPI tingkat akun dan mengambil pengaturan Blok Akses Publik tingkat akun.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"Allow retrieving the account-level Public Access Block settings",
         "Effect":"Allow",
         "Action":[
            "s3:GetAccountPublicAccessBlock"
         ],
         "Resource":[
            "*"
         ]
       }
    ]
}
```

Contoh kebijakan untuk Amazon S3

- Untuk melihat contoh kebijakan berbasis identitas Amazon S3, lihat. Kebijakan berbasis identitas untuk Amazon S3

- Untuk melihat contoh kebijakan berbasis sumber daya Amazon S3, lihat dan. Kebijakan bucket untuk Amazon S3 Mengonfigurasi kebijakan IAM untuk menggunakan titik akses

Sumber daya kebijakan untuk Amazon S3

Mendukung sumber daya kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan elemen `Resource` atau `NotResource`. Praktik terbaiknya, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Anda dapat melakukan ini untuk tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, misalnya operasi pencantuman, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

```
"Resource": "*"
```

Beberapa tindakan API Amazon S3 mendukung banyak sumber daya. Misalnya, `s3:GetObject` mengakses *`example-resource-1`* dan*`example-resource-2`*, jadi prinsipal harus memiliki izin untuk mengakses kedua sumber daya. Untuk menentukan beberapa sumber daya dalam satu pernyataan, pisahkan ARNs dengan koma, seperti yang ditunjukkan pada contoh berikut.

```
"Resource": [
      "example-resource-1",
      "example-resource-2"
```

Sumber daya di Amazon S3 adalah bucket, objek, titik akses, atau pekerjaan. Dalam kebijakan, gunakan Amazon Resource Name (ARN) bucket, objek, access point, atau job untuk mengidentifikasi sumber daya.

Untuk melihat daftar lengkap jenis sumber daya Amazon S3 dan jenisnya ARNs, lihat Sumber daya yang ditentukan oleh Amazon S3 di Referensi Otorisasi Layanan. Untuk mempelajari tindakan

mana yang dapat Anda tentukan ARN dari setiap sumber daya, lihat Tindakan yang ditentukan oleh Amazon S3.

Untuk informasi selengkapnya tentang izin operasi S3 API menurut jenis sumber daya S3, lihat. Izin yang diperlukan untuk operasi API Amazon S3

Karakter wildcard dalam sumber daya ARNs

Anda dapat menggunakan karakter wildcard sebagai bagian dari ARN sumber daya. Anda dapat menggunakan karakter wildcard (*dan?) dalam setiap segmen ARN (bagian dipisahkan oleh titik dua). Tanda bintang (*) menunjukkan kombinasi apa pun dari nol karakter atau lebih, dan tanda tanya (?) menunjukkan karakter tunggal. Anda dapat menggunakan beberapa * atau ? karakter di setiap segmen. Namun, karakter wildcard tidak dapat menjangkau segmen.

- ARN berikut menggunakan karakter * wildcard di `relative-ID` bagian ARN untuk mengidentifikasi semua objek dalam ember. *amzn-s3-demo-bucket*

```
arn:aws:s3:::amzn-s3-demo-bucket/*
```

- ARN berikut digunakan * untuk menunjukkan semua ember dan objek S3.

```
arn:aws:s3:::*
```

- ARN berikut menggunakan kedua karakter wildcard, * dan?, di bagian tersebut. `relative-ID` ARN ini mengidentifikasi semua objek dalam ember seperti*amzn-s3-demo-example1bucket*,, *amzn-s3-demo-example2bucketamzn-s3-demo-example3bucket*, dan sebagainya.

```
arn:aws:s3:::amzn-s3-demo-example?bucket/*
```

Variabel kebijakan untuk sumber daya ARNs

Anda dapat menggunakan variabel kebijakan di Amazon S3 ARNs. Pada waktu evaluasi kebijakan, variabel-variabel yang telah ditentukan ini digantikan oleh nilai-nilai yang sesuai. Misalkan Anda mengatur bucket Anda sebagai kumpulan folder, dengan satu folder untuk setiap pengguna Anda. Nama folder sama dengan nama pengguna. Untuk memberikan pengguna izin ke folder mereka, Anda dapat menentukan variabel kebijakan di ARN sumber daya:

```
arn:aws:s3:::bucket_name/developers/${aws:username}/
```

Saat runtime, ketika kebijakan dievaluasi, variabel `${aws:username}` dalam ARN sumber daya diganti dengan nama pengguna orang yang membuat permintaan.

Contoh kebijakan untuk Amazon S3

- Untuk melihat contoh kebijakan berbasis identitas Amazon S3, lihat. Kebijakan berbasis identitas untuk Amazon S3
- Untuk melihat contoh kebijakan berbasis sumber daya Amazon S3, lihat dan. Kebijakan bucket untuk Amazon S3 Mengonfigurasi kebijakan IAM untuk menggunakan titik akses

Kunci kondisi kebijakan untuk Amazon S3

Mendukung kunci kondisi kebijakan khusus layanan: Yes

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen `Condition` (atau blok `Condition`) akan memungkinkan Anda menentukan kondisi yang menjadi dasar suatu pernyataan berlaku. Elemen `Condition` bersifat opsional. Anda dapat membuat ekspresi bersyarat yang menggunakan operator kondisi, misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang diminta.

Jika Anda menentukan beberapa elemen `Condition` dalam sebuah pernyataan, atau beberapa kunci dalam elemen `Condition` tunggal, maka AWS akan mengevaluasinya menggunakan operasi `AND` logis. Jika Anda menentukan beberapa nilai untuk satu kunci kondisi, AWS mengevaluasi kondisi menggunakan `OR` operasi logis. Semua kondisi harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Sebagai contoh, Anda dapat memberikan izin kepada pengguna IAM untuk mengakses sumber daya hanya jika izin tersebut mempunyai tanda yang sesuai dengan nama pengguna IAM mereka. Untuk informasi selengkapnya, lihat Elemen kebijakan IAM: variabel dan tanda dalam Panduan Pengguna IAM.

AWS mendukung kunci kondisi global dan kunci kondisi khusus layanan. Untuk melihat semua kunci kondisi AWS global, lihat kunci konteks kondisi AWS global di Panduan Pengguna IAM.

Setiap kunci kondisi Amazon S3 memetakan ke header permintaan nama yang sama yang diizinkan oleh API tempat kondisinya dapat disetel. Kunci kondisi spesifik Amazon S3 menentukan perilaku

header permintaan nama yang sama. Misalnya, kunci kondisi yang `s3:VersionId` digunakan untuk memberikan izin bersyarat untuk
`s3:GetObjectVersion`
izin menentukan perilaku parameter `versionId` kueri yang Anda tetapkan dalam permintaan GET Object.

Untuk melihat daftar kunci kondisi Amazon S3, lihat Kunci kondisi untuk Amazon S3 di Referensi Otorisasi Layanan. Untuk mempelajari tindakan dan sumber daya yang dapat Anda gunakan kunci kondisi, lihat Tindakan yang ditentukan oleh Amazon S3.

Contoh: Membatasi unggahan objek ke objek dengan kelas penyimpanan tertentu

Misalkan Akun A, yang diwakili oleh ID akun*123456789012*, memiliki ember. Administrator Akun A ingin membatasi*Dave*, pengguna di Akun A, sehingga *Dave* dapat mengunggah objek ke bucket hanya jika objek disimpan di kelas `STANDARD_IA` penyimpanan. Untuk membatasi unggahan objek ke kelas penyimpanan tertentu, administrator Akun A dapat menggunakan kunci `s3:x-amz-storage-class` kondisi, seperti yang ditunjukkan dalam contoh kebijakan bucket berikut.

```
{
                "Version": "2012-10-17",
                "Statement": [
                  {
                    "Sid": "statement1",
                    "Effect": "Allow",
                    "Principal": {
                      "AWS": "arn:aws:iam::123456789012:user/Dave"
                    },
                    "Action": "s3:PutObject",
                    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*",
                    "Condition": {
                      "StringEquals": {
                        "s3:x-amz-storage-class": [
                          "STANDARD_IA"
                        ]
                      }
                    }
                  }
                ]
            }
```

Dalam contoh ini, blok `Condition` menentukan syarat `StringEquals` yang diterapkan untuk pasangan nilai kunci tertentu, `"s3:x-amz-acl":["public-read"]`. Ada satu pengaturan kunci yang telah ditetapkan yang dapat Anda gunakan dalam mengekspresikan syarat. Contoh tersebut menggunakan kunci kondisi `s3:x-amz-acl`. Kondisi ini mengharuskan pengguna untuk menyertakan `x-amz-acl` header dengan nilai `public-read` di setiap `PutObject` permintaan.

Contoh kebijakan untuk Amazon S3

- Untuk melihat contoh kebijakan berbasis identitas Amazon S3, lihat. Kebijakan berbasis identitas untuk Amazon S3
- Untuk melihat contoh kebijakan berbasis sumber daya Amazon S3, lihat dan. Kebijakan bucket untuk Amazon S3 Mengonfigurasi kebijakan IAM untuk menggunakan titik akses

ACLs di Amazon S3

Mendukung ACLs: Ya

Di Amazon S3, kontrol daftar kontrol akses (ACLs) yang Akun AWS memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan JSON.

> **Important**
>
> Mayoritas kasus penggunaan modern di Amazon S3 tidak lagi memerlukan penggunaan. ACLs

Untuk informasi tentang penggunaan ACLs untuk mengontrol akses di Amazon S3, lihat. Mengelola akses dengan ACLs

ABAC dengan Amazon S3

Mendukung ABAC (tag dalam kebijakan): Sebagian

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang menentukan izin berdasarkan atribut. Dalam AWS, atribut ini disebut tag. Anda dapat melampirkan tag ke entitas IAM (pengguna atau peran) dan ke banyak AWS sumber daya. Penandaan ke entitas dan sumber daya adalah langkah pertama dari ABAC. Kemudian rancanglah kebijakan ABAC untuk mengizinkan operasi ketika tanda milik prinsipal cocok dengan tanda yang ada di sumber daya yang ingin diakses.

ABAC sangat berguna di lingkungan yang berkembang dengan cepat dan berguna di situasi saat manajemen kebijakan menjadi rumit.

Untuk mengendalikan akses berdasarkan tanda, berikan informasi tentang tanda di elemen kondisi dari kebijakan menggunakan kunci kondisi `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, atau `aws:TagKeys`.

Jika sebuah layanan mendukung ketiga kunci kondisi untuk setiap jenis sumber daya, nilainya adalah Ya untuk layanan tersebut. Jika suatu layanan mendukung ketiga kunci kondisi untuk hanya beberapa jenis sumber daya, nilainya adalah Parsial.

Untuk informasi selengkapnya tentang ABAC, lihat Tentukan izin dengan otorisasi ABAC dalam Panduan Pengguna IAM. Untuk melihat tutorial yang menguraikan langkah-langkah pengaturan ABAC, lihat Menggunakan kontrol akses berbasis atribut (ABAC) dalam Panduan Pengguna IAM.

Untuk melihat contoh kebijakan berbasis identitas untuk membatasi akses ke pekerjaan Operasi Batch S3 berdasarkan tag, lihat. Mengontrol izin untuk Operasi Batch menggunakan tag pekerjaan

ABAC dan tag objek

Dalam kebijakan ABAC, objek menggunakan `s3:` tag, bukan `aws:` tag. Untuk mengontrol akses ke objek berdasarkan tag objek, Anda memberikan informasi tag dalam Conditionelemen kebijakan menggunakan tag berikut:

- `s3:ExistingObjectTag/tag-key`
- `s3:RequestObjectTagKeys`
- `s3:RequestObjectTag/tag-key`

Untuk informasi tentang penggunaan tag objek untuk mengontrol akses, termasuk contoh kebijakan izin, lihatKebijakan pemberian tag dan kontrol akses.

Menggunakan kredensyal sementara dengan Amazon S3

Mendukung kredensial sementara: Ya

Beberapa Layanan AWS tidak berfungsi saat Anda masuk menggunakan kredensil sementara. Untuk informasi tambahan, termasuk yang Layanan AWS bekerja dengan kredensi sementara, lihat Layanan AWS yang bekerja dengan IAM di Panduan Pengguna IAM.

Anda menggunakan kredensi sementara jika Anda masuk AWS Management Console menggunakan metode apa pun kecuali nama pengguna dan kata sandi. Misalnya, ketika Anda mengakses AWS

menggunakan tautan masuk tunggal (SSO) perusahaan Anda, proses tersebut secara otomatis membuat kredensil sementara. Anda juga akan secara otomatis membuat kredensial sementara ketika Anda masuk ke konsol sebagai seorang pengguna lalu beralih peran. Untuk informasi selengkapnya tentang peralihan peran, lihat Beralih dari pengguna ke peran IAM (konsol) dalam Panduan Pengguna IAM.

Anda dapat membuat kredensil sementara secara manual menggunakan API AWS CLI atau AWS . Anda kemudian dapat menggunakan kredensil sementara tersebut untuk mengakses. AWS AWS merekomendasikan agar Anda secara dinamis menghasilkan kredensi sementara alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat Kredensial keamanan sementara di IAM.

Teruskan sesi akses untuk Amazon S3

Mendukung sesi akses maju (FAS): Ya

Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat Sesi akses maju.

- FAS digunakan oleh Amazon S3 untuk melakukan panggilan AWS KMS untuk mendekripsi objek ketika SSE-KMS digunakan untuk mengenkripsi itu. Untuk informasi selengkapnya, lihat Menggunakan enkripsi sisi server dengan AWS KMS kunci (SSE-KMS).
- S3 Access Grants juga menggunakan FAS. Setelah Anda membuat hibah akses ke data S3 Anda untuk identitas tertentu, penerima hibah meminta kredensi sementara dari S3 Access Grants. S3 Access Grants memperoleh kredensi sementara untuk pemohon dari AWS STS dan memberikan kredensi kepada pemohon. Untuk informasi selengkapnya, lihat Minta akses ke data Amazon S3 melalui S3 Access Grants.

Peran layanan untuk Amazon S3

Mendukung peran layanan: Ya

Peran layanan adalah peran IAM yang diambil oleh sebuah layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Buat sebuah peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.

> **Warning**
>
> Mengubah izin untuk peran layanan dapat merusak fungsionalitas Amazon S3. Edit peran layanan hanya jika Amazon S3 memberikan panduan untuk melakukannya.

Peran terkait layanan untuk Amazon S3

Mendukung peran terkait layanan: Sebagian

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

Amazon S3 mendukung peran terkait layanan untuk Lensa Penyimpanan Amazon S3. Untuk detail tentang membuat atau mengelola peran terkait layanan Amazon S3, lihat. Menggunakan Peran Terkait Layanan untuk Lensa Penyimpanan Amazon S3

Layanan Amazon S3 sebagai Principal

Nama layanan dalam kebijakan	Fitur S3	Informasi selengkapnya
`s3.amazonaws.com`	Replikasi S3	Menyiapkan ikhtisar replikasi langsung
`s3.amazonaws.com`	Pemberitahuan acara S3	Notifikasi Peristiwa Amazon S3
`s3.amazonaws.com`	Inventaris S3	Membuat katalog dan menganalisis data Anda dengan S3 Inventory

Nama layanan dalam kebijakan	Fitur S3	Informasi selengkapnya
`access-grants.s3.amazonaws.com`	Pemberian Akses S3	Mendaftarkan lokasi
`batchoperations.s3.amazonaws.com`	Operasi Batch S3	Memberikan izin untuk Batch Operations
`logging.s3.amazonaws.com`	Pencatatan Akses Server S3	Mengaktifkan pencatatan akses server Amazon S3
`storage-lens.s3.amazonaws.com`	Lensa Penyimpanan S3	Melihat metrik Lensa Penyimpanan Amazon S3 menggunakan ekspor data

Bagaimana Amazon S3 Mengotorisasi Permintaan

Ketika Amazon S3 menerima permintaan—misalnya, operasi bucket atau operasi objek—pertama-tama akan memverifikasi apakah pemohon memiliki izin yang diperlukan. Amazon S3 mengevaluasi semua kebijakan akses yang relevan, kebijakan pengguna, dan kebijakan berbasis sumber daya (kebijakan bucket, daftar kontrol akses bucket (ACL), dan objek ACL) dalam memutuskan apakah akan mengotorisasi permintaan.

> **Note**
>
> Jika pemeriksaan izin Amazon S3 gagal menemukan izin yang valid, kesalahan Akses Ditolak (403 Terlarang) ditolak akan dikembalikan. Untuk informasi selengkapnya, lihat Memecahkan masalah kesalahan Akses Ditolak (403 Terlarang) di Amazon S3.

Untuk menentukan apakah pemohon memiliki izin untuk melakukan operasi tertentu, Amazon S3 melakukan hal berikut, secara berurutan, saat menerima permintaan:

1. Mengonversi semua kebijakan akses yang relevan (kebijakan pengguna, kebijakan bucket, dan ACLs) pada waktu berjalan menjadi serangkaian kebijakan untuk evaluasi.
2. Mengevaluasi serangkaian kebijakan yang dihasilkan dalam langkah-langkah berikut. Dalam setiap langkah, Amazon S3 akan mengevaluasi sebagian dari kebijakan dalam konteks tertentu, berdasarkan otoritas konteks.
   a. Konteks pengguna–Dalam konteks pengguna, akun induk yang merupakan pemilik pengguna adalah otoritas konteks.

      Amazon S3 mengevaluasi sebagian kebijakan yang dimiliki oleh akun induk. Sebagian kebijakan ini mencakup kebijakan pengguna yang dilampirkan oleh akun induk kepada pengguna. Jika induk juga memiliki sumber daya dalam permintaan (bucket atau objek), Amazon S3 juga mengevaluasi kebijakan sumber daya terkait (kebijakan bucket, bucket ACL, dan objek ACL) secara bersamaan.

      Pengguna harus memiliki izin dari akun induk untuk melakukan operasi.

      Langkah ini hanya berlaku jika permintaan diajukan oleh pengguna dalam Akun AWS. Jika permintaan dibuat dengan menggunakan kredensi pengguna root dari sebuah Akun AWS, Amazon S3 melewatkan langkah ini.

b. Konteks bucket — Dalam konteks bucket, Amazon S3 mengevaluasi kebijakan yang dimiliki oleh pemilik Akun AWS bucket.

   Apabila permintaan digunakan untuk operasi bucket, maka pemohon harus memiliki izin dari pemilik bucket. Jika permintaan untuk sebuah objek, maka Amazon S3 akan mengevaluasi semua kebijakan yang dimiliki oleh pemilik bucket untuk memeriksa apakah pemilik bucket belum secara jelas menolak akses ke objek tersebut. Jika ada penolakan jelas yang ditetapkan, Amazon S3 tidak mengotorisasi permintaan.

c. Konteks objek–Jika permintaan untuk objek, maka Amazon S3 akan mengevaluasi sebagian kebijakan yang dimiliki oleh pemilik objek.

Berikut adalah beberapa contoh skenario yang menggambarkan bagaimana Amazon S3 mengotorisasi permintaan.

Example — Pemohon adalah prinsipal IAM

Jika pemohon adalah prinsipal IAM, Amazon S3 harus menentukan apakah Akun AWS induk tempat prinsipal tersebut telah memberikan izin pokok yang diperlukan untuk melakukan operasi. Selain itu, jika permintaan tersebut untuk operasi bucket, seperti permintaan untuk mencantumkan konten bucket, maka Amazon S3 harus memverifikasi bahwa pemilik bucket telah memberikan izin bagi pemohon untuk melakukan operasi tersebut. Untuk melakukan operasi tertentu pada sumber daya, prinsipal IAM memerlukan izin dari induk Akun AWS yang dimilikinya dan Akun AWS yang memiliki sumber daya tersebut.

Example — Pemohon adalah prinsipal IAM — Jika permintaan adalah untuk operasi pada objek yang tidak dimiliki pemilik bucket

Jika permintaan adalah untuk operasi pada objek yang tidak dimiliki pemilik bucket, selain memastikan pemohon memiliki izin dari pemilik objek, Amazon S3 juga harus memeriksa kebijakan bucket untuk memastikan pemilik bucket belum menetapkan penolakan eksplisit pada objek. Pemilik bucket (yang membayar tagihan) dapat dengan tegas menolak akses ke objek di dalam bucket, terlepas dari siapa yang memilikinya. Pemilik bucket juga dapat menghapus objek apa pun yang ada dalam bucket.

Secara default, ketika orang lain Akun AWS mengunggah objek ke bucket tujuan umum S3 Anda, akun tersebut (penulis objek) memiliki objek, memiliki akses ke objek tersebut, dan dapat memberi pengguna lain akses ke sana melalui daftar kontrol akses (). ACLs Anda dapat menggunakan Object

Ownership untuk mengubah perilaku default ini sehingga ACLs dinonaktifkan dan Anda, sebagai pemilik bucket, secara otomatis memiliki setiap objek di bucket tujuan umum Anda. Akibatnya, kontrol akses untuk data Anda didasarkan pada kebijakan, seperti kebijakan pengguna IAM, kebijakan bucket S3, kebijakan titik akhir virtual private cloud (VPC), dan kebijakan kontrol AWS Organizations layanan (). SCPs Untuk informasi selengkapnya, lihat Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.

Untuk informasi lebih lanjut tentang bagaimana Amazon S3 mengevaluasi kebijakan akses untuk mengotorisasi atau tolak permintaan untuk operasi bucket dan operasi objek, lihat topik berikut:

Topik

- Bagaimana Amazon S3 Mengotorisasi Permintaan Operasi bucket
- Bagaimana Amazon S3 Memberikan Otorisasi Permintaan Operasi Objek

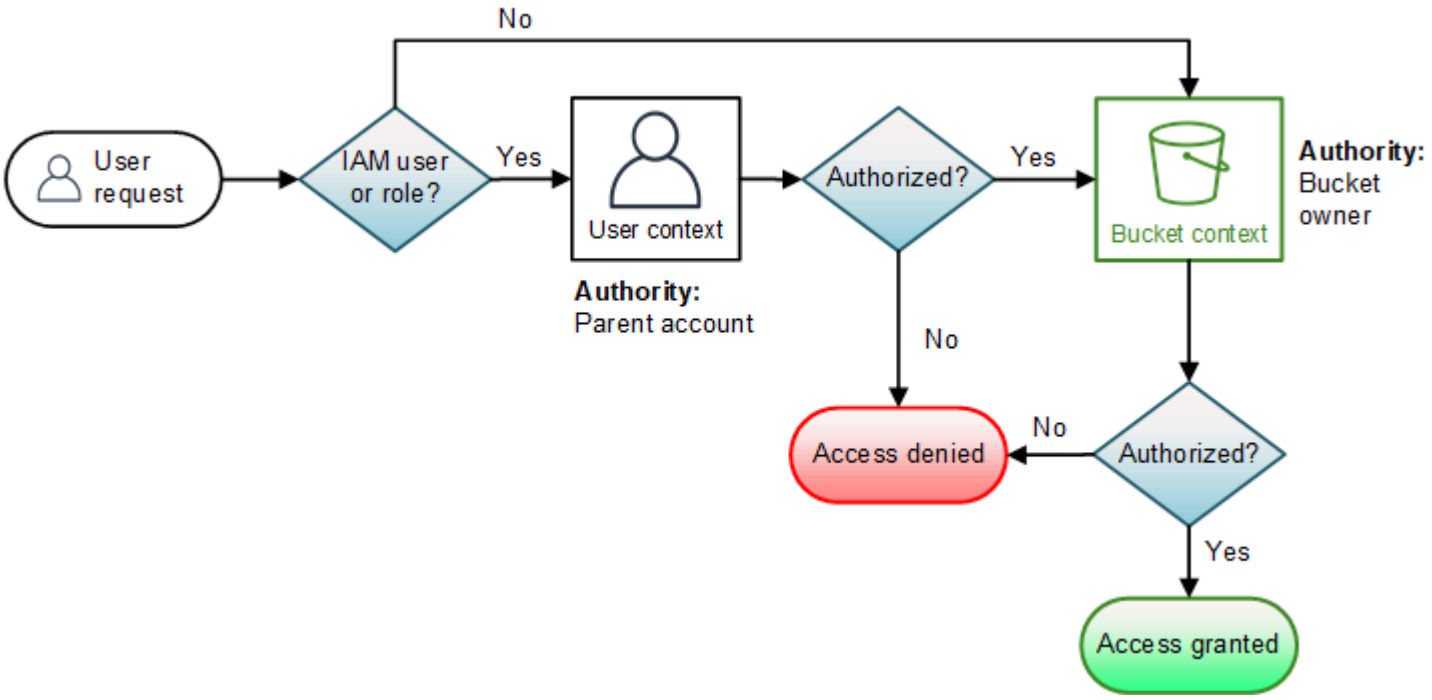
Bagaimana Amazon S3 Mengotorisasi Permintaan Operasi bucket

Ketika Amazon S3 menerima permintaan untuk operasi bucket, Amazon S3 mengubah semua izin yang relevan menjadi serangkaian kebijakan untuk mengevaluasi pada runtime. Izin yang relevan termasuk izin berbasis sumber daya (misalnya, kebijakan bucket dan daftar kontrol akses bucket) dan kebijakan pengguna IAM jika permintaan dari pengguna utama IAM. Amazon S3 kemudian mengevaluasi serangkaian kebijakan yang dihasilkan dalam serangkaian langkah sesuai dengan konteks tertentu—konteks pengguna atau konteks bucket:

1. Konteks pengguna - Jika pemohon adalah prinsipal IAM, kepala sekolah harus memiliki izin dari orang tua yang Akun AWS menjadi miliknya. Dalam langkah ini, Amazon S3 mengevaluasi sebagian kebijakan yang dimiliki oleh akun induk (juga disebut sebagai otoritas konteks). Bagian dari kebijakan ini mencakup kebijakan pengguna yang dilampirkan oleh akun induk pada pengguna utama. Jika induk juga memiliki sumber daya dalam permintaan (dalam hal ini, bucket), maka Amazon S3 juga akan mengevaluasi kebijakan sumber daya yang sesuai (kebijakan bucket dan ACL bucket) pada waktu yang bersamaan. Setiap kali permintaan untuk operasi bucket dibuat, log akses server akan mencatat ID kanonik dari pemohon. Untuk informasi selengkapnya, lihat Pencatatan permintaan dengan pencatatan akses server.
2. Konteks bucket Pemohon harus memiliki izin dari pemilik bucket untuk melakukan operasi bucket tertentu. Pada langkah ini, Amazon S3 mengevaluasi subset kebijakan yang dimiliki oleh pemilik bucket Akun AWS .

Pemilik bucket dapat memberikan izin dengan menggunakan kebijakan bucket atau ACL bucket. Jika Akun AWS yang memiliki bucket juga merupakan akun induk dari prinsipal IAM, maka bucket tersebut dapat mengonfigurasi izin bucket dalam kebijakan pengguna.

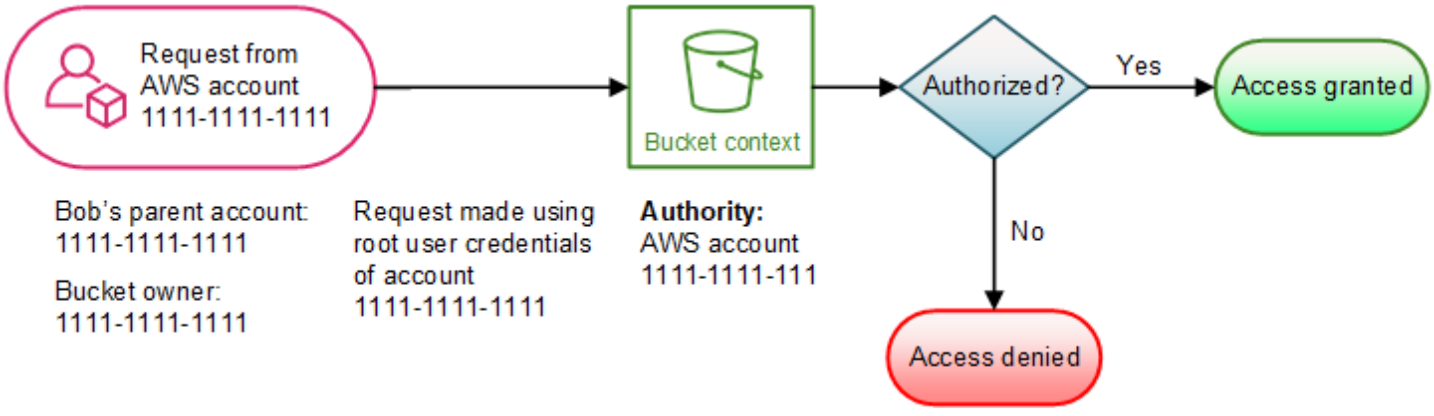
Berikut ini adalah ilustrasi grafis evaluasi berbasis konteks untuk operasi bucket.



Contoh-contoh berikut ini menggambarkan logika evaluasi.

Contoh 1: Operasi bucket Diminta oleh pemilik bucket

Dalam contoh ini, pemilik bucket mengirimkan permintaan operasi bucket dengan menggunakan kredensial root Akun AWS.

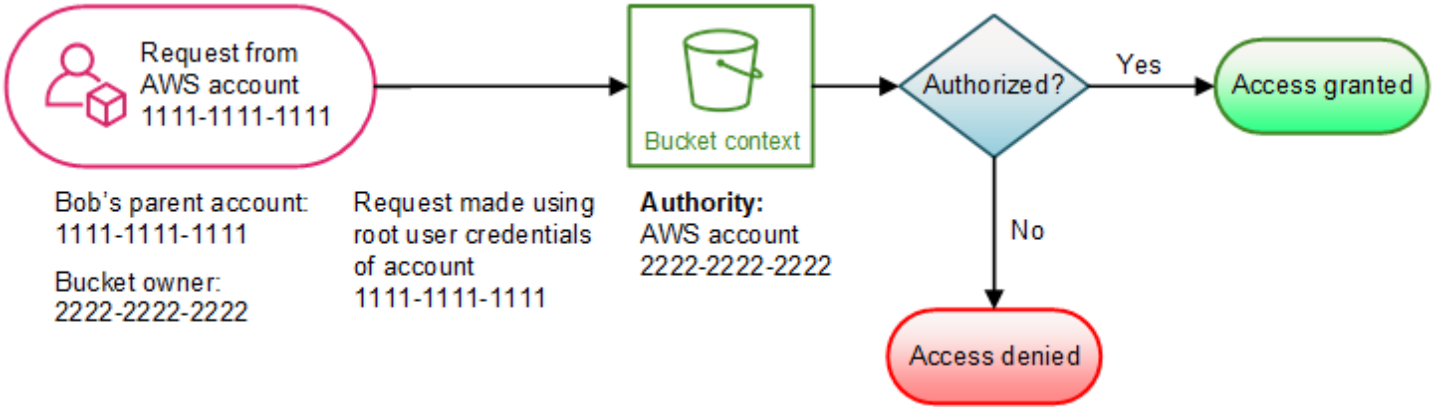


Amazon S3 melakukan evaluasi konteks sebagai berikut:

1. Karena permintaan dibuat dengan menggunakan kredensial pengguna root Akun AWS, maka konteks pengguna tidak dievaluasi.
2. Dalam konteks bucket, Amazon S3 akan meninjau kebijakan bucket untuk menentukan apakah pemohon memiliki izin untuk melakukan operasi atau tidak. Amazon S3 mengotorisasi permintaan tersebut.

Contoh 2: Operasi bucket yang diminta oleh pemilik bucket Akun AWS yang bukan pemilik bucket

Dalam contoh ini, permintaan dibuat dengan menggunakan kredensial pengguna root Akun AWS 1111-1111-1111 untuk operasi bucket yang dimiliki oleh 2222-2222-2222 Akun AWS . Tidak ada pengguna IAM yang dilibatkan dalam permintaan ini.

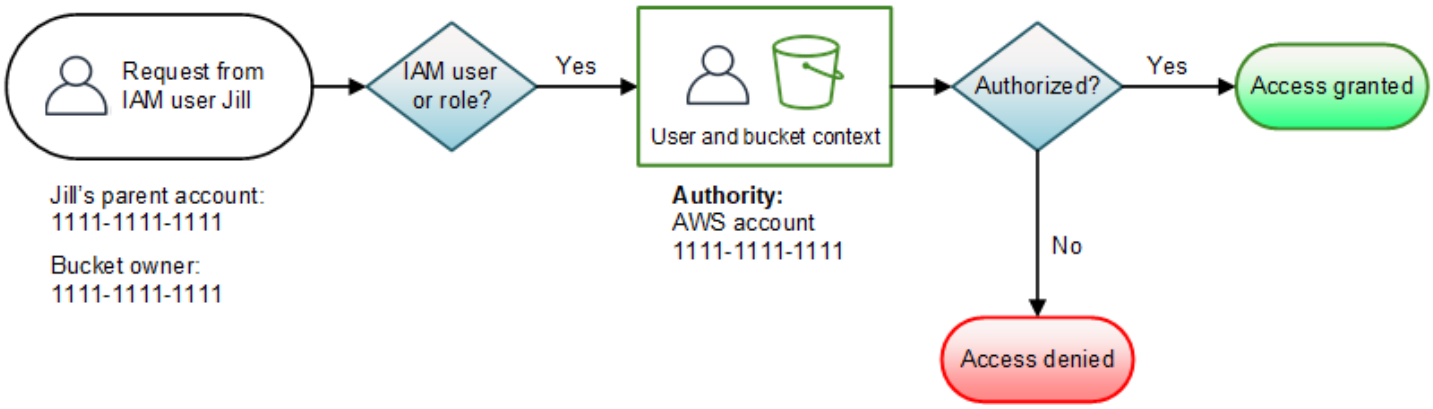


Dalam contoh ini, Amazon S3 mengevaluasi konteksnya sebagai berikut:

1. Karena permintaan dibuat dengan menggunakan kredensi pengguna root dari sebuah Akun AWS, konteks pengguna tidak dievaluasi.
2. Dalam konteks bucket, Amazon S3 memeriksa kebijakan bucket. Jika pemilik bucket (Akun AWS 2222-2222-2222) belum mengizinkan Akun AWS 1111-1111-1111 untuk melakukan operasi yang diminta, Amazon S3 menolak permintaan tersebut. Jika tidak, Amazon S3 akan memberikan permintaan dan melakukan operasi.

Contoh 3: Operasi bucket yang diminta oleh kepala sekolah IAM yang Akun AWS induknya juga pemilik bucket

Sebagai contoh, permintaan dikirim oleh Jill, pengguna IAM di Akun AWS 1111-1111-1111, yang juga merupakan pemilik bucket.



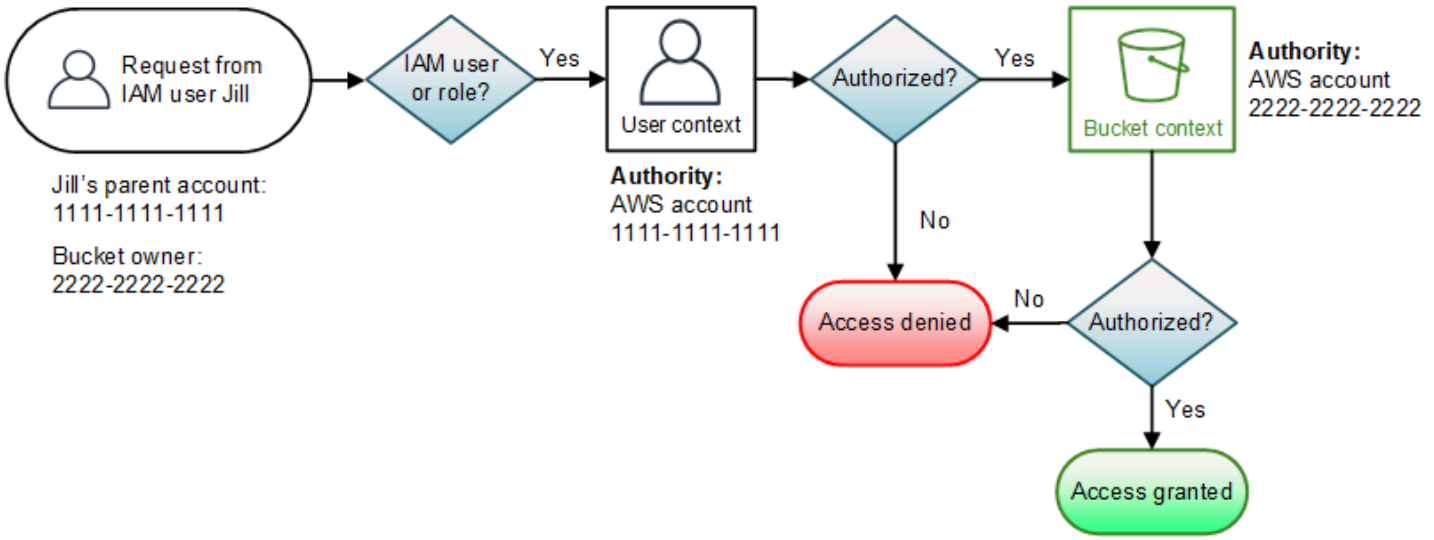
Amazon S3 akan melakukan evaluasi konteks sebagai berikut:

1. Karena permintaan berasal dari prinsipal IAM, dalam konteks pengguna, Amazon S3 mengevaluasi semua kebijakan milik Akun AWS induk untuk menentukan apakah Jill memiliki izin untuk melakukan operasi.

   Dalam contoh ini, induk Akun AWS 1111-1111-1111, yang menjadi milik prinsipal, juga merupakan pemilik ember. Akibatnya, selain kebijakan pengguna, Amazon S3 juga mengevaluasi kebijakan bucket dan bucket ACL dalam konteks yang sama karena mereka termasuk dalam akun yang sama.
2. Karena Amazon S3 mengevaluasi kebijakan bucket dan ACL bucket sebagai bagian dari konteks pengguna, maka itu tidak akan mengevaluasi konteks bucket.

Contoh 4: Operasi bucket yang diminta oleh kepala sekolah IAM yang induknya Akun AWS bukan pemilik bucket

Dalam contoh ini, permintaan dikirim oleh Jill, pengguna IAM yang induknya Akun AWS adalah 1111-1111-1111, tetapi ember tersebut dimiliki oleh yang lain, 2222-2222-2222. Akun AWS



Jill akan membutuhkan izin dari induk Akun AWS dan pemilik ember. Amazon S3 mengevaluasi konteks seperti berikut ini:

1. Karena permintaan tersebut berasal dari pengguna utama IAM, Amazon S3 melakukan evaluasi konteks pengguna dengan meninjau kebijakan yang ditulis oleh akun tersebut untuk memverifikasi bahwa Jill memiliki izin yang diperlukan. Jika Jill memiliki izin, Amazon S3 melanjutkan untuk mengevaluasi konteks bucket. Jika Jill tidak memiliki izin, ia menolak permintaan tersebut.
2. Dalam konteks bucket, Amazon S3 memverifikasi bahwa pemilik bucket 2222-2222-2222 telah memberikan izin kepada Jill (atau orang tuanya) untuk melakukan operasi yang diminta. Akun AWS Jika dia memiliki izin itu, Amazon S3 memberikan permintaan dan melakukan operasi. Jika tidak, maka Amazon S3 akan menolak permintaan tersebut.

Bagaimana Amazon S3 Memberikan Otorisasi Permintaan Operasi Objek

Saat Amazon S3 menerima permintaan untuk operasi objek, itu akan mengubah semua izin yang relevan— izin berbasis sumber daya (daftar kontrol akses (ACL) objek, kebijakan bucket, ACL bucket) dan kebijakan pengguna IAM—menjadi serangkaian kebijakan yang akan dievaluasi pada waktu berjalan. Kemudian, itu akan mengevaluasi serangkaian kebijakan yang dihasilkan dengan serangkaian langkah. Dalam setiap langkah, ia mengevaluasi subset kebijakan dalam tiga konteks tertentu—konteks pengguna, konteks bucket, dan konteks objek:

1. Konteks pengguna - Jika pemohon adalah prinsipal IAM, kepala sekolah harus memiliki izin dari orang tua yang Akun AWS menjadi miliknya. Dalam langkah ini, Amazon S3 mengevaluasi sebagian kebijakan yang dimiliki oleh akun induk (juga disebut sebagai otoritas konteks). Sebagian dari kebijakan ini mencakup kebijakan pengguna yang dilampirkan oleh akun induk pada

pengguna utama. Jika induk juga memiliki sumber daya dalam permintaan (bucket atau objek), Amazon S3 mengevaluasi kebijakan sumber daya terkait (kebijakan bucket, bucket ACL, dan objek ACL) secara bersamaan.

> **Note**
>
> Jika Akun AWS induk memiliki sumber daya (bucket atau objek), ia dapat memberikan izin sumber daya ke prinsipal IAM-nya dengan menggunakan kebijakan pengguna atau kebijakan sumber daya.

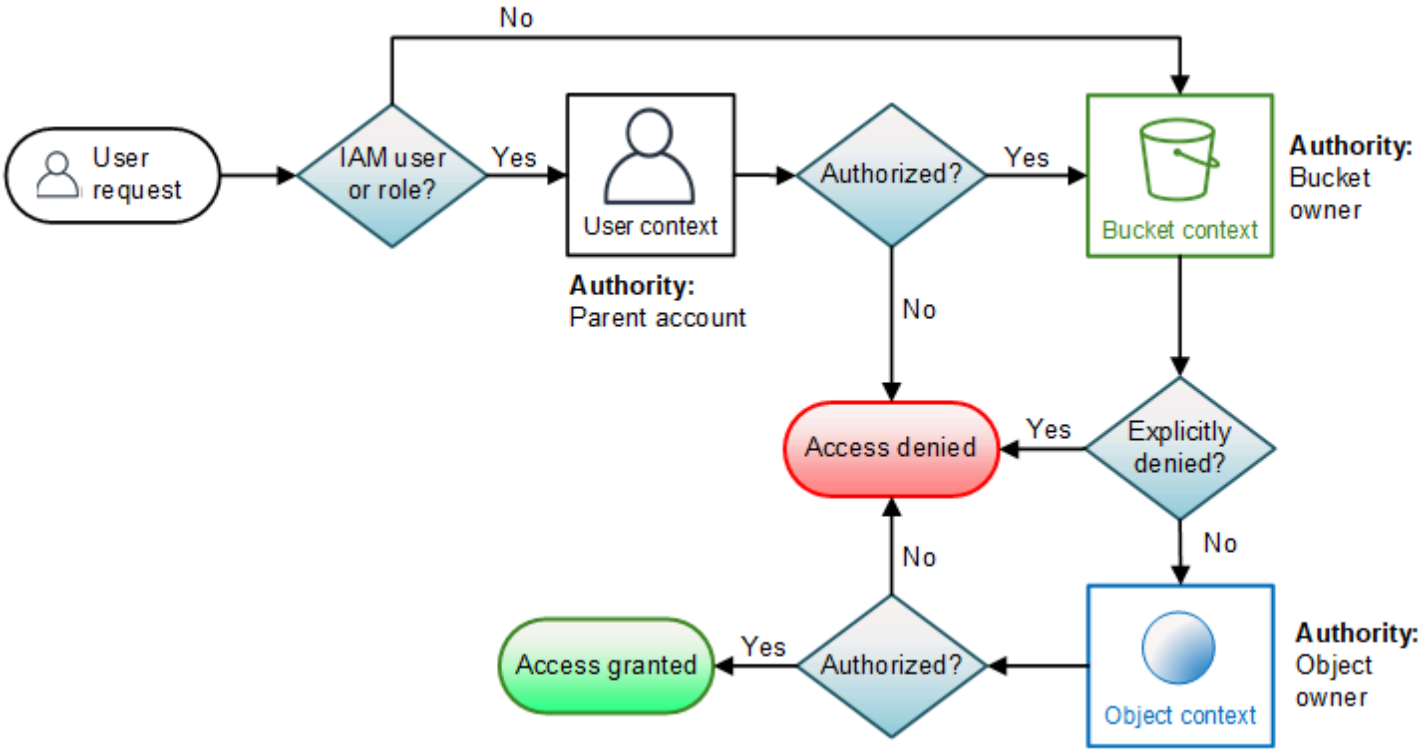
2. Konteks bucket–Dalam konteks ini, Amazon S3 mengevaluasi kebijakan yang dimiliki oleh Akun AWS yang memiliki bucket tersebut.

   Jika pemilik Akun AWS objek dalam permintaan tidak sama dengan pemilik bucket, Amazon S3 akan memeriksa kebijakan jika pemilik bucket secara eksplisit menolak akses ke objek tersebut. Jika ada penolakan tegas yang ditetapkan pada objek, Amazon S3 tidak mengizinkan permintaan tersebut.

3. Konteks objek–Pemohon harus memiliki izin dari pemilik objek untuk melakukan operasi objek tertentu. Dalam langkah ini, Amazon S3 mengevaluasi ACL objek.

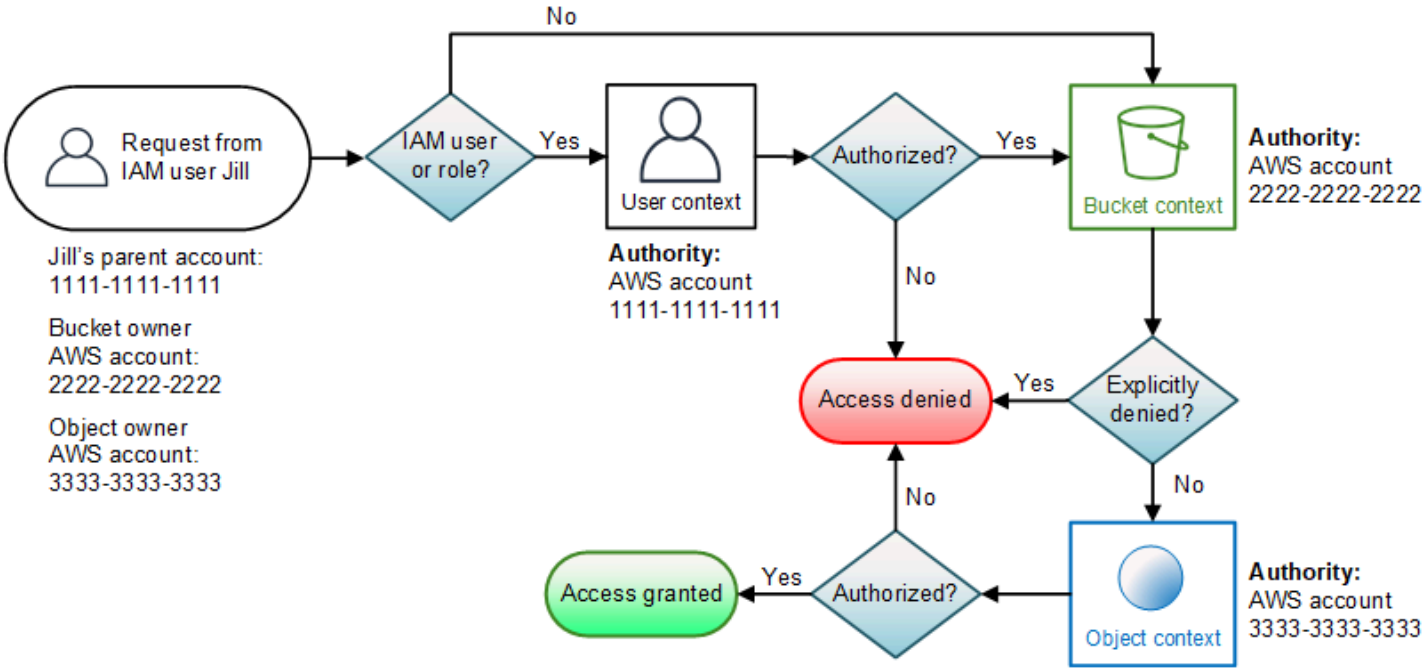
> **Note**
>
> Jika pemilik bucket dan objek sama, maka akses ke objek dapat diberikan dalam kebijakan bucket, yang dievaluasi dalam konteks bucket. Jika pemiliknya berbeda, maka pemilik objek harus menggunakan ACL objek untuk memberikan izin. Jika Akun AWS yang memiliki objek juga merupakan akun induk tempat prinsipal IAM berada, ia dapat mengonfigurasi izin objek dalam kebijakan pengguna, yang dievaluasi pada konteks pengguna. Untuk informasi lebih lanjut tentang penggunaan berbagai alternatif kebijakan akses ini, lihat Panduan yang menggunakan kebijakan untuk mengelola akses ke sumber daya Amazon S3.
>
> Jika Anda sebagai pemilik bucket ingin memiliki semua objek di bucket dan menggunakan kebijakan atau kebijakan bucket berdasarkan IAMto kelola akses ke objek tersebut, Anda dapat menerapkan pengaturan yang diberlakukan pemilik bucket untuk Kepemilikan Objek. Dengan pengaturan ini, Anda sebagai pemilik bucket secara otomatis memiliki dan memiliki kontrol penuh atas setiap objek di bucket Anda. Bucket dan objek tidak ACLs dapat diedit dan tidak lagi dipertimbangkan untuk diakses. Untuk informasi selengkapnya, lihat Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.

Berikut ini adalah ilustrasi dari evaluasi berbasis konteks untuk operasi objek.



Contoh permintaan operasi objek

Dalam contoh ini, pengguna IAM Jill, yang induknya Akun AWS adalah 1111-1111-1111, mengirimkan permintaan operasi objek (misalnya,`GetObject`) untuk objek yang dimiliki oleh Akun AWS 3333-3333-3333 dalam ember yang dimiliki oleh 2222-2222-2222. Akun AWS



Jill akan membutuhkan izin dari orang tua Akun AWS, pemilik ember, dan pemilik objek. Amazon S3 mengevaluasi konteksnya seperti berikut:

1. Karena permintaan tersebut berasal dari prinsipal IAM, Amazon S3 mengevaluasi konteks pengguna untuk memverifikasi bahwa Akun AWS induk 1111-1111-1111 telah memberikan izin kepada Jill untuk melakukan operasi yang diminta. Jika dia memiliki izin tersebut, maka Amazon S3 akan mengevaluasi konteks bucket. Jika tidak, maka Amazon S3 akan menolak permintaan tersebut.
2. Dalam konteks bucket, pemilik bucket, Akun AWS 2222-2222-2222, adalah otoritas konteks. Amazon S3 mengevaluasi kebijakan bucket untuk menentukan apakah pemilik bucket telah secara jelas menolak akses Jill ke objek.
3. Dalam konteks objek, otorisasi konteks adalah Akun AWS 3333-3333-3333, pemilik objek. Amazon S3 mengevaluasi ACL objek untuk menentukan apakah Jill memiliki izin untuk mengakses objek tersebut. Jika ya, maka Amazon S3 akan mengesahkan permintaan tersebut.

Izin yang diperlukan untuk operasi API Amazon S3

> **Note**
>
> Halaman ini berisi uraian tentang tindakan kebijakan Amazon S3 untuk bucket tujuan umum. Untuk mempelajari selengkapnya tentang tindakan kebijakan Amazon S3 untuk bucket direktori, lihat. Tindakan untuk ember direktori

Untuk melakukan operasi API S3, Anda harus memiliki izin yang tepat. Halaman ini memetakan operasi API S3 ke izin yang diperlukan. Untuk memberikan izin untuk melakukan operasi API S3, Anda harus membuat kebijakan yang valid (seperti kebijakan bucket S3 atau kebijakan berbasis identitas IAM), dan menentukan tindakan terkait dalam elemen kebijakan. `Action` Tindakan ini disebut tindakan kebijakan. Tidak setiap operasi API S3 diwakili oleh satu izin (satu tindakan kebijakan), dan beberapa izin (beberapa tindakan kebijakan) diperlukan untuk banyak operasi API yang berbeda.

Saat membuat kebijakan, Anda harus menentukan `Resource` elemen berdasarkan jenis sumber daya yang benar yang diperlukan oleh tindakan kebijakan Amazon S3 terkait. Halaman ini mengkategorikan izin untuk operasi API S3 berdasarkan jenis sumber daya. Untuk informasi selengkapnya tentang jenis sumber daya, lihat Jenis sumber daya yang ditentukan oleh Amazon S3 di Referensi Otorisasi Layanan. Untuk daftar lengkap tindakan kebijakan Amazon S3, sumber daya,

dan kunci kondisi untuk digunakan dalam kebijakan, lihat Tindakan, sumber daya, dan kunci kondisi untuk Amazon S3 di Referensi Otorisasi Layanan. Untuk daftar lengkap operasi Amazon S3 API, lihat Tindakan API Amazon S3 di Referensi API Layanan Penyimpanan Sederhana Amazon.

Untuk informasi selengkapnya tentang cara mengatasi `403 Forbidden` kesalahan HTTP di S3, lihatMemecahkan masalah kesalahan akses ditolak (403 Forbidden) di Amazon S3. Untuk informasi selengkapnya tentang fitur IAM yang akan digunakan dengan S3, lihat. Bagaimana Amazon S3 bekerja dengan IAM Untuk informasi selengkapnya tentang praktik terbaik keamanan S3, lihatPraktik Terbaik Keamanan untuk Amazon S3.

Topik

- Operasi dan izin bucket
- Operasi dan izin objek
- Titik akses untuk operasi dan izin bucket tujuan umum
- Operasi dan izin Objek Lambda Access Point
- Operasi dan izin Jalur Akses Multi-Wilayah
- Operasi dan izin pekerjaan Batch
- Operasi dan izin konfigurasi Lensa Penyimpanan S3
- Lensa Penyimpanan S3 mengelompokkan operasi dan izin
- S3 Access Memberikan operasi dan izin instans
- Akses S3 Memberikan operasi dan izin lokasi
- S3 Access Grants memberikan operasi dan izin
- Operasi dan izin akun

Operasi dan izin bucket

Operasi bucket adalah operasi API S3 yang beroperasi pada tipe sumber daya bucket. Anda harus menentukan tindakan kebijakan S3 untuk operasi bucket dalam kebijakan bucket atau kebijakan berbasis identitas IAM.

Dalam kebijakan, `Resource` elemen harus berupa bucket Amazon Resource Name (ARN). Untuk informasi selengkapnya tentang format `Resource` elemen dan kebijakan contoh, lihatOperasi bucket.

> **Note**
>
> Untuk memberikan izin ke operasi bucket dalam kebijakan jalur akses, perhatikan hal berikut:

- Izin yang diberikan untuk operasi bucket dalam kebijakan jalur akses hanya efektif jika bucket yang mendasarinya mengizinkan izin yang sama. Saat menggunakan titik akses, Anda harus mendelegasikan kontrol akses dari bucket ke titik akses atau menambahkan izin yang sama dalam kebijakan titik akses ke kebijakan bucket yang mendasarinya.
- Dalam kebijakan titik akses yang memberikan izin untuk operasi bucket, `Resource` elemen harus `accesspoint` ARN. Untuk informasi selengkapnya tentang format `Resource` elemen dan kebijakan contoh, lihatOperasi bucket dalam kebijakan untuk titik akses untuk bucket tujuan umum. Untuk informasi selengkapnya tentang kebijakan jalur akses, lihatMengonfigurasi kebijakan IAM untuk menggunakan titik akses.
- Tidak semua operasi bucket didukung oleh titik akses. Untuk informasi selengkapnya, lihat Kompatibilitas titik akses dengan operasi S3.

Berikut ini adalah pemetaan operasi bucket dan tindakan kebijakan yang diperlukan.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
CreateBucket	(Diperlukan) `s3:CreateBucket`	Diperlukan untuk membuat bucket s3 baru.
	(Diperlukan secara kondisional) `s3:PutBucketAcl`	Diperlukan jika Anda ingin menggunakan daftar kontrol akses (ACL) untuk menentukan izin pada bucket saat Anda membuat `CreateBucket` permintaan.
	(Diperlukan secara kondisional)`s3:PutBucketObjectLockConfiguration` , `s3:PutBucketVersioning`	Diperlukan jika Anda ingin mengaktifkan Object Lock saat membuat bucket.
	(Diperlukan secara kondisional) `s3:PutBucketOwnershipControls`	Diperlukan jika Anda ingin menentukan Kepemilikan

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
		Objek S3 saat Anda membuat bucket.
CreateBucketMetadataTableConfiguration	(Diperlukan)`s3:CreateBucketMetadataTableConfiguration` ,`s3tables:CreateNamespace` ,`s3tables:CreateTable` ,`s3tables:GetTable` , `s3tables:PutTablePolicy`	Diperlukan untuk membuat konfigurasi tabel metadata pada bucket tujuan umum. Untuk membuat tabel metadata di keranjang tabel yang ditentukan dalam konfigurasi tabel metadata, Anda harus memiliki izin yang ditentukan. `s3tables` Jika Anda juga ingin mengintegrasikan bucket tabel Anda dengan layanan AWS analitik sehingga Anda dapat menanyakan tabel metadata Anda, Anda memerlukan izin tambahan. Untuk informasi selengkapnya, lihat Mengintegrasikan Tabel Amazon S3 AWS dengan layanan analitik.
DeleteBucket	(Diperlukan) `s3:DeleteBucket`	Diperlukan untuk menghapus bucket S3.
DeleteBucketAnalyticsConfiguration	(Diperlukan) `s3:PutAnalyticsConfiguration`	Diperlukan untuk menghapus konfigurasi analitik S3 dari bucket S3.
DeleteBucketCors	(Diperlukan) `s3:PutBucketCORS`	Diperlukan untuk menghapus konfigurasi cross-origin resource sharing (CORS) untuk bucket.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
DeleteBucketEncryption	(Diperlukan) `s3:PutEncryptionConfiguration`	Diperlukan untuk mengatur ulang konfigurasi enkripsi default untuk bucket S3 sebagai enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3).
DeleteBucketIntelligentTieringConfiguration	(Diperlukan) `s3:PutIntelligentTieringConfiguration`	Diperlukan untuk menghapus konfigurasi S3 Intelligent-Tiering yang ada dari bucket S3.
DeleteBucketInventoryConfiguration	(Diperlukan) `s3:PutInventoryConfiguration`	Diperlukan untuk menghapus konfigurasi S3 Inventory dari bucket S3.
DeleteBucketLifecycle	(Diperlukan) `s3:PutLifecycleConfiguration`	Diperlukan untuk menghapus konfigurasi Siklus Hidup S3 untuk bucket S3.
DeleteBucketMetadataTableConfiguration	(Diperlukan) `s3:DeleteBucketMetadataTableConfiguration`	Diperlukan untuk menghapus konfigurasi tabel metadata dari bucket tujuan umum.
DeleteBucketMetricsConfiguration	(Diperlukan) `s3:PutMetricsConfiguration`	Diperlukan untuk menghapus konfigurasi metrik untuk metrik CloudWatch permintaan Amazon dari bucket S3.
DeleteBucketOwnershipControls	(Diperlukan) `s3:PutBucketOwnershipControls`	Diperlukan untuk menghapus pengaturan Object Ownership untuk bucket S3. Setelah penghapusan, pengaturan Object Ownership menjadi`Object writer`.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
DeleteBucketPolicy	(Diperlukan) `s3:DeleteBucketPolicy`	Diperlukan untuk menghapus kebijakan bucket S3.
DeleteBucketReplication	(Diperlukan) `s3:PutReplicationConfiguration`	Diperlukan untuk menghapus konfigurasi replikasi bucket S3.
DeleteBucketTagging	(Diperlukan) `s3:PutBucketTagging`	Diperlukan untuk menghapus tag dari bucket S3.
DeleteBucketWebsite	(Diperlukan) `s3:DeleteBucketWebsite`	Diperlukan untuk menghapus konfigurasi situs web untuk bucket S3.
DeletePublicAccessBlock(Tingkat ember)	(Diperlukan) `s3:PutBucketPublicAccessBlock`	Diperlukan untuk menghapus konfigurasi blok akses publik untuk bucket S3.
GetBucketAccelerateConfiguration	(Diperlukan) `s3:GetAccelerateConfiguration`	Diperlukan untuk menggunakan subsumber daya akselerasi untuk mengembalikan status Amazon S3 Transfer Acceleration dari bucket, yang diaktifkan atau ditangguhkan.
GetBucketAcl	(Diperlukan) `s3:GetBucketAcl`	Diperlukan untuk mengembalikan daftar kontrol akses (ACL) dari bucket S3.
GetBucketAnalyticsConfiguration	(Diperlukan) `s3:GetAnalyticsConfiguration`	Diperlukan untuk mengembalikan konfigurasi analitik yang diidentifikasi oleh ID konfigurasi analitik dari bucket S3.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
GetBucketCors	(Diperlukan) `s3:GetBucketCORS`	Diperlukan untuk mengembalikan konfigurasi cross-origin resource sharing (CORS) untuk bucket S3.
GetBucketEncryption	(Diperlukan) `s3:GetEncryptionConfiguration`	Diperlukan untuk mengembalikan konfigurasi enkripsi default untuk bucket S3.
GetBucketIntelligentTieringConfiguration	(Diperlukan) `s3:GetIntelligentTieringConfiguration`	Diperlukan untuk mendapatkan konfigurasi S3 Intelligent-Tiering dari bucket S3.
GetBucketInventoryConfiguration	(Diperlukan) `s3:GetInventoryConfiguration`	Diperlukan untuk mengembalikan konfigurasi inventaris yang diidentifikasi oleh ID konfigurasi inventaris dari bucket.
GetBucketLifecycle	(Diperlukan) `s3:GetLifecycleConfiguration`	Diperlukan untuk mengembalikan konfigurasi Siklus Hidup S3 bucket.
GetBucketLocation	(Diperlukan) `s3:GetBucketLocation`	Diperlukan untuk mengembalikan Wilayah AWS tempat ember S3 berada.
GetBucketLogging	(Diperlukan) `s3:GetBucketLogging`	Diperlukan untuk mengembalikan status logging bucket S3 dan izin yang dimiliki pengguna untuk melihat dan memodifikasi status tersebut.
GetBucketMetadataTableConfiguration	(Diperlukan) `s3:GetBucketMetadataTableConfiguration`	Diperlukan untuk mengambil konfigurasi tabel metadata untuk bucket tujuan umum.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
GetBucketMetricsConfiguration	(Diperlukan) `s3:GetMetricsConfiguration`	Diperlukan untuk mendapatkan konfigurasi metrik yang ditentukan oleh ID konfigurasi metrik dari bucket.
GetBucketNotificationConfiguration	(Diperlukan) `s3:GetBucketNotification`	Diperlukan untuk mengembalikan konfigurasi notifikasi bucket S3.
GetBucketOwnershipControls	(Diperlukan) `s3:GetBucketOwnershipControls`	Diperlukan untuk mengambil pengaturan Object Ownership untuk bucket S3.
GetBucketPolicy	(Diperlukan) `s3:GetBucketPolicy`	Diperlukan untuk mengembalikan kebijakan bucket S3.
GetBucketPolicyStatus	(Diperlukan) `s3:GetBucketPolicyStatus`	Diperlukan untuk mengambil status kebijakan bucket S3, yang menunjukkan apakah bucket bersifat publik.
GetBucketReplication	(Diperlukan) `s3:GetReplicationConfiguration`	Diperlukan untuk mengembalikan konfigurasi replikasi bucket S3.
GetBucketRequestPayment	(Diperlukan) `s3:GetBucketRequestPayment`	Diperlukan untuk mengembalikan konfigurasi pembayaran permintaan untuk bucket S3.
GetBucketVersioning	(Diperlukan) `s3:GetBucketVersioning`	Diperlukan untuk mengembalikan status pembuatan versi bucket S3.
GetBucketTagging	(Diperlukan) `s3:GetBucketTagging`	Diperlukan untuk mengembalikan set tag yang terkait dengan bucket S3.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
GetBucketWebsite	(Diperlukan) `s3:GetBucketWebsite`	Diperlukan untuk mengembalikan konfigurasi situs web untuk bucket S3.
GetObjectLockConfiguration	(Diperlukan) `s3:GetBucketObjectLockConfiguration`	Diperlukan untuk mendapatkan konfigurasi Object Lock untuk bucket S3.
GetPublicAccessBlock(Tingkat ember)	(Diperlukan) `s3:GetBucketPublicAccessBlock`	Diperlukan untuk mengambil konfigurasi akses publik blok untuk bucket S3.
HeadBucket	(Diperlukan) `s3:ListBucket`	Diperlukan untuk menentukan apakah ada bucket dan apakah Anda memiliki izin untuk mengaksesnya.
ListBucketAnalyticsConfigurations	(Diperlukan) `s3:GetAnalyticsConfiguration`	Diperlukan untuk membuat daftar konfigurasi analitik untuk bucket S3.
ListBucketIntelligentTieringConfigurations	(Diperlukan) `s3:GetIntelligentTieringConfiguration`	Diperlukan untuk membuat daftar konfigurasi S3 Intelligent-Tiering dari bucket S3.
ListBucketInventoryConfigurations	(Diperlukan) `s3:GetInventoryConfiguration`	Diperlukan untuk mengembalikan daftar konfigurasi inventaris untuk bucket S3.
ListBucketMetricsConfigurations	(Diperlukan) `s3:GetMetricsConfiguration`	Diperlukan untuk membuat daftar konfigurasi metrik untuk bucket S3.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
ListObjects	(Diperlukan) `s3:ListBucket`	Diperlukan untuk mencantumkan beberapa atau semua (hingga 1.000) objek dalam ember S3.
	(Diperlukan secara kondisional) `s3:GetObjectAcl`	Diperlukan jika Anda ingin menampilkan informasi pemilik objek.
ListObjectsV2	(Diperlukan) `s3:ListBucket`	Diperlukan untuk mencantumkan beberapa atau semua (hingga 1.000) objek dalam ember S3.
	(Diperlukan secara kondisional) `s3:GetObjectAcl`	Diperlukan jika Anda ingin menampilkan informasi pemilik objek.
ListObjectVersions	(Diperlukan) `s3:ListBucketVersions`	Diperlukan untuk mendapatkan metadata tentang semua versi objek dalam bucket S3.
PutBucketAccelerateConfiguration	(Diperlukan) `s3:PutAccelerateConfiguration`	Diperlukan untuk mengatur konfigurasi percepatan bucket yang ada.
PutBucketAcl	(Diperlukan) `s3:PutBucketAcl`	Diperlukan untuk menggunakan daftar kontrol akses (ACLs) untuk menyetel izin pada bucket yang ada.
PutBucketAnalyticsConfiguration	(Diperlukan) `s3:PutAnalyticsConfiguration`	Diperlukan untuk mengatur konfigurasi analitik untuk bucket S3.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
PutBucketCors	(Diperlukan) `s3:PutBucketCORS`	Diperlukan untuk menyetel konfigurasi cross-origin resource sharing (CORS) untuk bucket S3.
PutBucketEncryption	(Diperlukan) `s3:PutEncryptionConfiguration`	Diperlukan untuk mengonfigurasi enkripsi default untuk bucket S3.
PutBucketIntelligentTieringConfiguration	(Diperlukan) `s3:PutIntelligentTieringConfiguration`	Diperlukan untuk menempatkan konfigurasi S3 Intelligent-Tiering ke bucket S3.
PutBucketInventoryConfiguration	(Diperlukan) `s3:PutInventoryConfiguration`	Diperlukan untuk menambahkan konfigurasi inventaris ke bucket S3.
PutBucketLifecycle	(Diperlukan) `s3:PutLifecycleConfiguration`	Diperlukan untuk membuat konfigurasi Siklus Hidup S3 baru atau mengganti konfigurasi siklus hidup yang ada untuk bucket S3.
PutBucketLogging	(Diperlukan) `s3:PutBucketLogging`	Diperlukan untuk mengatur parameter logging untuk bucket S3 dan menentukan izin untuk siapa yang dapat melihat dan memodifikasi parameter logging.
PutBucketMetricsConfiguration	(Diperlukan) `s3:PutMetricsConfiguration`	Diperlukan untuk menyetel atau memperbarui konfigurasi metrik untuk metrik CloudWatch permintaan Amazon dari bucket S3.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
PutBucketNotificationConfiguration	(Diperlukan) `s3:PutBucketNotification`	Diperlukan untuk mengaktifkan pemberitahuan peristiwa tertentu untuk bucket S3.
PutBucketOwnershipControls	(Diperlukan) `s3:PutBucketOwnershipControls`	Diperlukan untuk membuat atau memodifikasi pengaturan Object Ownership untuk bucket S3.
PutBucketPolicy	(Diperlukan) `s3:PutBucketPolicy`	Diperlukan untuk menerapkan kebijakan bucket S3 ke bucket.
PutBucketReplication	(Diperlukan) `s3:PutReplicationConfiguration`	Diperlukan untuk membuat konfigurasi replikasi baru atau mengganti yang sudah ada untuk bucket S3.
PutBucketRequestPayment	(Diperlukan) `s3:PutBucketRequestPayment`	Diperlukan untuk mengatur konfigurasi pembayaran permintaan untuk ember.
PutBucketTagging	(Diperlukan) `s3:PutBucketTagging`	Diperlukan untuk menambahkan satu set tag ke bucket S3.
PutBucketVersioning	(Diperlukan) `s3:PutBucketVersioning`	Diperlukan untuk mengatur status pembuatan versi bucket S3.
PutBucketWebsite	(Diperlukan) `s3:PutBucketWebsite`	Diperlukan untuk mengkonfigurasi bucket sebagai situs web dan mengatur konfigurasi situs web.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
PutObjectLockConfiguration	(Diperlukan) `s3:PutBucketObjectLockConfiguration`	Diperlukan untuk menempatkan konfigurasi Object Lock pada bucket S3.
PutPublicAccessBlock(Tingkatember)	(Diperlukan) `s3:PutBucketPublicAccessBlock`	Diperlukan untuk membuat atau memodifikasi konfigurasi blokir akses publik untuk bucket S3.

Operasi dan izin objek

Operasi objek adalah operasi API S3 yang beroperasi pada jenis sumber daya objek. Anda harus menentukan tindakan kebijakan S3 untuk operasi objek dalam kebijakan berbasis sumber daya (seperti kebijakan bucket, kebijakan titik akses, kebijakan Titik Akses Multi-Wilayah, kebijakan titik akhir VPC) atau kebijakan berbasis identitas IAM.

Dalam kebijakan, `Resource` elemen harus menjadi objek ARN. Untuk informasi selengkapnya tentang format `Resource` elemen dan kebijakan contoh, lihatOperasi objek.

> **Note**
>
> - AWS KMS tindakan kebijakan (`kms:GenerateDataKey`dan`kms:Decrypt`) hanya berlaku untuk jenis AWS KMS sumber daya dan harus ditentukan dalam kebijakan berbasis identitas IAM dan kebijakan berbasis AWS KMS sumber daya (kebijakan utama).AWS KMS Anda tidak dapat menentukan tindakan AWS KMS kebijakan dalam kebijakan berbasis sumber daya S3, seperti kebijakan bucket S3.
> - Saat Anda menggunakan titik akses untuk mengontrol akses ke operasi objek, Anda dapat menggunakan kebijakan titik akses. Untuk memberikan izin ke operasi objek dalam kebijakan jalur akses, perhatikan hal berikut:
>   - Dalam kebijakan titik akses yang memberikan izin untuk operasi objek, `Resource` elemen harus ARNs untuk objek yang diakses melalui titik akses. Untuk informasi selengkapnya tentang format `Resource` elemen dan kebijakan contoh, lihatOperasi objek dalam kebijakan titik akses.

- Tidak semua operasi objek didukung oleh titik akses. Untuk informasi selengkapnya, lihat Kompatibilitas titik akses dengan operasi S3.
- Tidak semua operasi objek didukung oleh Titik Akses Multi-Wilayah. Untuk informasi selengkapnya, lihat Kompatibilitas Titik Akses Multi-Wilayah dengan operasi S3.

Berikut ini adalah pemetaan operasi objek dan tindakan kebijakan yang diperlukan.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
AbortMultipartUpload	(Diperlukan) `s3:AbortMultipartUpload`	Diperlukan untuk membatalkan unggahan multipart.
CompleteMultipartUpload	(Diperlukan) `s3:PutObject`	Diperlukan untuk menyelesaikan unggahan multipart.
	(Diperlukan secara kondisional) `kms:Decrypt`	Diperlukan jika Anda ingin menyelesaikan unggahan multipart untuk objek terenkripsi kunci yang dikelola AWS KMS pelanggan.
CopyObject	Untuk objek sumber:	Untuk objek sumber:
	(Diperlukan) Salah satu `s3:GetObject` atau `s3:GetObjectVersion`	• `s3:GetObject` — Diperlukan jika Anda ingin menyalin objek dari bucket sumber tanpa menentukan `versionId` permintaan. • `s3:GetObjectVersion` — Diperlukan jika Anda ingin menyalin versi objek tertentu dari bucket sumber dengan menentukan `versionId` dalam permintaan.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
	(Diperlukan secara kondisional) `kms:Decrypt`	Diperlukan jika Anda ingin menyalin objek terenkripsi kunci terkelola AWS KMS pelanggan dari bucket sumber.
	Untuk objek tujuan:	Untuk objek tujuan:
	(Diperlukan) `s3:PutObject`	Diperlukan untuk meletakkan objek yang disalin di ember tujuan.
	(Diperlukan secara kondisional) `s3:PutObjectAcl`	Diperlukan jika Anda ingin meletakkan objek yang disalin dengan daftar kontrol akses objek (ACL) ke bucket tujuan saat Anda membuat `CopyObject` permintaan.
	(Diperlukan secara kondisional) `s3:PutObjectTagging`	Diperlukan jika Anda ingin meletakkan objek yang disalin dengan penandaan objek ke bucket tujuan saat Anda membuat `CopyObject` permintaan.
	(Diperlukan secara kondisional) `kms:GenerateDataKey`	Diperlukan jika Anda ingin mengenkripsi objek yang disalin dengan kunci yang dikelola AWS KMS pelanggan dan memasukkannya ke bucket tujuan.
	(Diperlukan secara kondisional) `s3:PutObjectRetention`	Diperlukan jika Anda ingin mengatur konfigurasi retensi Object Lock untuk objek baru.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
	(Diperlukan secara kondisional) `s3:PutObjectLegalHold`	Diperlukan jika Anda ingin menempatkan penahanan hukum Object Lock pada objek baru.
CreateMultipartUpload	(Diperlukan) `s3:PutObject`	Diperlukan untuk membuat unggahan multipart.
	(Diperlukan secara kondisional) `s3:PutObjectAcl`	Diperlukan jika Anda ingin mengatur izin daftar kontrol akses objek (ACL) untuk objek yang diunggah.
	(Diperlukan secara kondisional) `s3:PutObjectTagging`	Diperlukan jika Anda ingin menambahkan penandaan objek ke objek yang diunggah.
	(Diperlukan secara kondisional) `kms:GenerateDataKey`	Diperlukan jika Anda ingin menggunakan kunci yang dikelola AWS KMS pelanggan untuk mengenkripsi objek saat Anda memulai unggahan multibagian.
	(Diperlukan secara kondisional) `s3:PutObjectRetention`	Diperlukan jika Anda ingin mengatur konfigurasi retensi Object Lock untuk objek yang diunggah.
	(Diperlukan secara kondisional) `s3:PutObjectLegalHold`	Diperlukan jika Anda ingin menerapkan penahanan hukum Object Lock ke objek yang diunggah.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
DeleteObject	(Diperlukan) Baik `s3:DeleteObject` atau `s3:DeleteObjectVersion`	• `s3:DeleteObject` — Diperlukan jika Anda ingin menghapus objek tanpa menentukan `versionId` dalam permintaan. • `s3:DeleteObjectVersion` — Diperlukan jika Anda ingin menghapus versi tertentu dari suatu objek dengan menentukan `versionId` dalam permintaan.
	(Diperlukan secara kondisional) `s3:BypassGovernanceRetention`	Diperlukan jika Anda ingin menghapus objek yang dilindungi oleh mode tata kelola untuk retensi Object Lock.
DeleteObjects	(Diperlukan) Baik `s3:DeleteObject` atau `s3:DeleteObjectVersion`	• `s3:DeleteObject` — Diperlukan jika Anda ingin menghapus objek tanpa menentukan `versionId` dalam permintaan. • `s3:DeleteObjectVersion` — Diperlukan jika Anda ingin menghapus versi tertentu dari suatu objek dengan menentukan `versionId` dalam permintaan.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
	(Diperlukan secara kondisional) `s3:BypassGovernanceRetention`	Diperlukan jika Anda ingin menghapus objek yang dilindungi oleh modus tata kelola untuk retensi Object Lock.
DeleteObjectTagging	(Diperlukan) Baik `s3:DeleteObjectTagging` atau `s3:DeleteObjectVersionTagging`	• `s3:DeleteObjectTagging` — Diperlukan jika Anda ingin menghapus seluruh kumpulan tag objek tanpa menentukan `versionId` dalam permintaan. • `s3:DeleteObjectVersionTagging` — Diperlukan jika Anda ingin menghapus tag dari versi objek tertentu dengan menentukan `versionId` dalam permintaan.
GetObject	(Diperlukan) Baik `s3:GetObject` atau `s3:GetObjectVersion`	• `s3:GetObject` — Diperlukan jika Anda ingin mendapatkan objek tanpa menentukan `versionId` dalam permintaan. • `s3:GetObjectVersion` — Diperlukan jika Anda ingin mendapatkan versi tertentu dari suatu objek dengan menentukan `versionId` dalam permintaan.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
	(Diperlukan secara kondisional) `kms:Decrypt`	Diperlukan jika Anda ingin mendapatkan dan mendekripsi objek terenkripsi kunci yang dikelola AWS KMS pelanggan.
	(Diperlukan secara kondisional) `s3:GetObjectTagging`	Diperlukan jika Anda ingin mendapatkan tag-set objek saat Anda membuat `GetObject` permintaan.
	(Diperlukan secara kondisional) `s3:GetObjectLegalHold`	Diperlukan jika Anda ingin mendapatkan status penahanan hukum Object Lock objek saat ini.
	(Diperlukan secara kondisional) `s3:GetObjectRetention`	Diperlukan jika Anda ingin mengambil pengaturan retensi Object Lock untuk objek.
GetObjectAcl	(Diperlukan) Baik `s3:GetObjectAcl` atau `s3:GetObjectVersionAcl`	• `s3:GetObjectAcl` — Diperlukan jika Anda ingin mendapatkan daftar kontrol akses (ACL) dari suatu objek tanpa menentukan `versionId` dalam permintaan. • `s3:GetObjectVersionAcl` — Diperlukan jika Anda ingin mendapatkan daftar kontrol akses (ACL) dari suatu objek dengan menentukan `versionId` dalam permintaan.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
GetObjectAttributes	(Diperlukan) Baik `s3:GetObject` atau `s3:GetObjectVersion`	• `s3:GetObject` — Diperlukan jika Anda ingin mengambil atribut yang terkait dengan objek tanpa menentukan `versionId` dalam permintaan. • `s3:GetObjectVersion` — Diperlukan jika Anda ingin mengambil atribut yang terkait dengan versi objek tertentu dengan menentukan `versionId` dalam permintaan.
	(Diperlukan secara kondisional) `kms:Decrypt`	Diperlukan jika Anda ingin mengambil atribut yang terkait dengan objek terenkripsi kunci yang dikelola AWS KMS pelanggan.
GetObjectLegalHold	(Diperlukan) `s3:GetObjectLegalHold`	Diperlukan untuk mendapatkan status penahanan hukum Object Lock objek saat ini.
GetObjectRetention	(Diperlukan) `s3:GetObjectRetention`	Diperlukan untuk mengambil pengaturan retensi Object Lock untuk objek.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
GetObjectTagging	(Diperlukan) Baik `s3:GetObjectTagging` atau `s3:GetObjectVersionTagging`	• `s3:GetObjectTagging` — Diperlukan jika Anda ingin mendapatkan set tag objek tanpa menentukan `versionId` dalam permintaan. • `s3:GetObjectVersionTagging` — Diperlukan jika Anda ingin mendapatkan tag dari versi objek tertentu dengan menentukan `versionId` dalam permintaan.
GetObjectTorrent	(Diperlukan) `s3:GetObject`	Diperlukan untuk mengembalikan file torrent dari suatu objek.
HeadObject	(Diperlukan) `s3:GetObject`	Diperlukan untuk mengambil metadata dari objek tanpa mengembalikan objek itu sendiri.
	(Diperlukan secara kondisional) `s3:GetObjectLegalHold`	Diperlukan jika Anda ingin mendapatkan status penahanan hukum Object Lock objek saat ini.
	(Diperlukan secara kondisional) `s3:GetObjectRetention`	Diperlukan jika Anda ingin mengambil pengaturan retensi Object Lock untuk objek.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
ListMultipartUploads	(Diperlukan) `s3:ListBucketMultipartUploads`	Diperlukan untuk mencantumkan unggahan multipart yang sedang berlangsung dalam bucket.
ListParts	(Diperlukan) `s3:ListMultipartUploadParts`	Diperlukan untuk membuat daftar bagian yang telah diunggah untuk unggahan multibagian tertentu.
	(Diperlukan secara kondisional) `kms:Decrypt`	Diperlukan jika Anda ingin membuat daftar bagian dari unggahan multibagian kunci terenkripsi yang dikelola AWS KMS pelanggan.
PutObject	(Diperlukan) `s3:PutObject`	Diperlukan untuk meletakkan objek.
	(Diperlukan secara kondisional) `s3:PutObjectAcl`	Diperlukan jika Anda ingin menempatkan daftar kontrol akses objek (ACL) saat Anda membuat `PutObject` permintaan.
	(Diperlukan secara kondisional) `s3:PutObjectTagging`	Diperlukan jika Anda ingin menempatkan penandaan objek saat Anda membuat `PutObject` permintaan.
	(Diperlukan secara kondisional) `kms:GenerateDataKey`	Diperlukan jika Anda ingin mengenkripsi objek dengan kunci yang dikelola AWS KMS pelanggan.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
	(Diperlukan secara kondisional) `s3:PutObjectRetention`	Diperlukan jika Anda ingin mengatur konfigurasi retensi Object Lock pada objek.
	(Diperlukan secara kondisional) `s3:PutObjectLegalHold`	Diperlukan jika Anda ingin menerapkan konfigurasi penahanan hukum Object Lock ke objek tertentu.
PutObjectAcl	(Diperlukan) Baik `s3:PutObjectAcl` atau `s3:PutObjectVersionAcl`	• `s3:PutObjectAcl` — Diperlukan jika Anda ingin mengatur izin daftar kontrol akses (ACL) untuk objek baru atau yang sudah ada tanpa menentukan permintaan`versionId` . • `s3:PutObjectVersionAcl` — Diperlukan jika Anda ingin mengatur izin daftar kontrol akses (ACL) untuk objek baru atau yang sudah ada dengan menentukan `versionId` dalam permintaan.
PutObjectLegalHold	(Diperlukan) `s3:PutObjectLegalHold`	Diperlukan untuk menerapkan konfigurasi penahanan legal Object Lock ke objek.
PutObjectRetention	(Diperlukan) `s3:PutObjectRetention`	Diperlukan untuk menerapkan konfigurasi retensi Object Lock ke objek.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
	(Diperlukan secara kondisional) `s3:BypassGovernanceRetention`	Diperlukan jika Anda ingin melewati mode tata kelola konfigurasi retensi Object Lock.
PutObjectTagging	(Diperlukan) Baik `s3:PutObjectTagging` atau `s3:PutObjectVersionTagging`	• `s3:PutObjectTagging` — Diperlukan jika Anda ingin menyetel set tag yang disediakan ke objek yang sudah ada di bucket tanpa menentukan `versionId` permintaan. • `s3:PutObjectVersionTagging` — Diperlukan jika Anda ingin menyetel set tag yang disediakan ke objek yang sudah ada di bucket dengan menentukan `versionId` dalam permintaan.
RestoreObject	(Diperlukan) `s3:RestoreObject`	Diperlukan untuk mengembalikan salinan objek yang diarsipkan.
SelectObjectContent	(Diperlukan) `s3:GetObject`	Diperlukan untuk memfilter isi objek S3 berdasarkan pernyataan bahasa kueri terstruktur sederhana (SQL).
	(Diperlukan secara kondisional) `kms:Decrypt`	Diperlukan jika Anda ingin memfilter konten objek S3 yang dienkripsi dengan kunci yang dikelola AWS KMS pelanggan.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
UploadPart	(Diperlukan) `s3:PutObject`	Diperlukan untuk mengunggah bagian dalam unggahan multibagian.
	(Diperlukan secara kondisional) `kms:GenerateDataKey`	Diperlukan jika Anda ingin meletakkan bagian unggahan dan mengenkripsi dengan kunci yang dikelola AWS KMS pelanggan.
UploadPartCopy	Untuk objek sumber:	Untuk objek sumber:
	(Diperlukan) Baik `s3:GetObject` atau `s3:GetObjectVersion`	• `s3:GetObject` — Diperlukan jika Anda ingin menyalin objek dari bucket sumber tanpa menentukan `versionId` permintaan. • `s3:GetObjectVersion` — Diperlukan jika Anda ingin menyalin versi objek tertentu dari bucket sumber dengan menentukan `versionId` dalam permintaan.
	(Diperlukan secara kondisional) `kms:Decrypt`	Diperlukan jika Anda ingin menyalin objek terenkripsi kunci terkelola AWS KMS pelanggan dari bucket sumber.
	Untuk bagian tujuan:	Untuk bagian tujuan:
	(Diperlukan) `s3:PutObject`	Diperlukan untuk mengunggah bagian unggahan multibagian ke bucket tujuan.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
	(Diperlukan secara kondisional) `kms:GenerateDataKey`	Diperlukan jika Anda ingin mengenkripsi bagian dengan kunci terkelola AWS KMS pelanggan saat Anda mengunggah bagian tersebut ke bucket tujuan.

Titik akses untuk operasi dan izin bucket tujuan umum

Operasi titik akses adalah operasi API S3 yang beroperasi pada jenis `accesspoint` sumber daya. Anda harus menentukan tindakan kebijakan S3 untuk operasi titik akses dalam kebijakan berbasis identitas IAM, bukan dalam kebijakan bucket atau kebijakan titik akses.

Dalam kebijakan, `Resource` elemennya haruslah `accesspoint` ARN. Untuk informasi selengkapnya tentang format `Resource` elemen dan kebijakan contoh, lihatTitik akses untuk operasi bucket tujuan umum.

> **Note**
>
> Jika Anda ingin menggunakan titik akses untuk mengontrol akses ke operasi bucket atau objek, perhatikan hal berikut:
>
> - Untuk menggunakan titik akses untuk mengontrol akses ke operasi bucket, lihatOperasi bucket dalam kebijakan untuk titik akses untuk bucket tujuan umum.
> - Untuk menggunakan titik akses untuk mengontrol akses ke operasi objek, lihatOperasi objek dalam kebijakan titik akses.
> - Untuk informasi selengkapnya tentang cara mengonfigurasi kebijakan titik akses, lihatMengonfigurasi kebijakan IAM untuk menggunakan titik akses.

Berikut ini adalah pemetaan operasi titik akses dan tindakan kebijakan yang diperlukan.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
CreateAccessPoint	(Diperlukan) `s3:CreateAccessPoint`	Diperlukan untuk membuat titik akses yang terkait dengan bucket S3.
DeleteAccessPoint	(Diperlukan) `s3:DeleteAccessPoint`	Diperlukan untuk menghapus titik akses.
DeleteAccessPointPolicy	(Diperlukan) `s3:DeleteAccessPointPolicy`	Diperlukan untuk menghapus kebijakan titik akses.
GetAccessPointPolicy	(Diperlukan) `s3:GetAccessPointPolicy`	Diperlukan untuk mengambil kebijakan jalur akses.
GetAccessPointPolicyStatus	(Diperlukan) `s3:GetAccessPointPolicyStatus`	Diperlukan untuk mengambil informasi apakah titik akses yang ditentukan saat ini memiliki kebijakan yang memungkinkan akses publik.
PutAccessPointPolicy	(Diperlukan) `s3:PutAccessPointPolicy`	Diperlukan untuk menempatkan kebijakan titik akses.

Operasi dan izin Objek Lambda Access Point

Operasi Object Lambda Access Point adalah operasi API S3 yang beroperasi pada jenis sumber daya. `objectlambdaaccesspoint` Untuk informasi selengkapnya tentang cara mengonfigurasi kebijakan untuk operasi Titik Akses Objek Lambda, lihat. Mengonfigurasi kebijakan IAM untuk Titik Akses Lambda Objek

Berikut ini adalah pemetaan operasi Object Lambda Access Point dan tindakan kebijakan yang diperlukan.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
CreateAccessPointForObjectLambda	(Diperlukan) `s3:CreateAccessPointForObjectLambda`	Diperlukan untuk membuat Object Lambda Access Point.
DeleteAccessPointForObjectLambda	(Diperlukan) `s3:DeleteAccessPointForObjectLambda`	Diperlukan untuk menghapus Titik Akses Objek Lambda tertentu.
DeleteAccessPointPolicyForObjectLambda	(Diperlukan) `s3:DeleteAccessPointPolicyForObjectLambda`	Diperlukan untuk menghapus kebijakan pada Titik Akses Lambda Objek tertentu.
GetAccessPointConfigurationForObjectLambda	(Diperlukan) `s3:GetAccessPointConfigurationForObjectLambda`	Diperlukan untuk mengambil konfigurasi Object Lambda Access Point.
GetAccessPointForObjectLambda	(Diperlukan) `s3:GetAccessPointForObjectLambda`	Diperlukan untuk mengambil informasi tentang Object Lambda Access Point.
GetAccessPointPolicyForObjectLambda	(Diperlukan) `s3:GetAccessPointPolicyForObjectLambda`	Diperlukan untuk mengembalikan kebijakan titik akses yang terkait dengan Titik Akses Objek Lambda yang ditentukan.
GetAccessPointPolicyStatusForObjectLambda	(Diperlukan) `s3:GetAccessPointPolicyStatusForObjectLambda`	Diperlukan untuk mengembalikan status kebijakan untuk kebijakan Titik Akses Objek Lambda tertentu.
PutAccessPointConfigurationForObjectLambda	(Diperlukan) `s3:PutAccessPointConfigurationForObjectLambda`	Diperlukan untuk mengatur konfigurasi Object Lambda Access Point.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
PutAccessPointPolicyForObjectLambda	(Diperlukan) `s3:PutAccessPointPolicyForObjectLambda`	Diperlukan untuk mengaitkan kebijakan akses dengan Titik Akses Objek Lambda tertentu.

Operasi dan izin Jalur Akses Multi-Wilayah

Operasi Titik Akses Multi-Wilayah adalah operasi API S3 yang beroperasi pada jenis `multiregionaccesspoint` sumber daya. Untuk informasi selengkapnya tentang cara mengonfigurasi kebijakan untuk operasi Titik Akses Multi-Wilayah, lihatContoh kebijakan Titik Akses Multi-Wilayah.

Berikut ini adalah pemetaan operasi Titik Akses Multi-Wilayah dan tindakan kebijakan yang diperlukan.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
CreateMultiRegionAccessPoint	(Diperlukan) `s3:CreateMultiRegionAccessPoint`	Diperlukan untuk membuat Titik Akses Multi-Wilayah dan mengaitkannya dengan bucket S3.
DeleteMultiRegionAccessPoint	(Diperlukan) `s3:DeleteMultiRegionAccessPoint`	Diperlukan untuk menghapus Titik Akses Multi-Wilayah.
DescribeMultiRegionAccessPointOperation	(Diperlukan) `s3:DescribeMultiRegionAccessPointOperation`	Diperlukan untuk mengambil status permintaan asinkron untuk mengelola Titik Akses Multi-Wilayah.
GetMultiRegionAccessPoint	(Diperlukan) `s3:GetMultiRegionAccessPoint`	Diperlukan untuk mengembalikan informasi konfigurasi tentang Titik Akses Multi-Wilayah yang ditentukan.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
GetMultiRegionAccessPointPolicy	(Diperlukan) `s3:GetMultiRegionAccessPointPolicy`	Diperlukan untuk mengembalikan kebijakan kontrol akses dari Titik Akses Multi-Wilayah yang ditentukan.
GetMultiRegionAccessPointPolicyStatus	(Diperlukan) `s3:GetMultiRegionAccessPointPolicyStatus`	Diperlukan untuk mengembalikan status kebijakan untuk Titik Akses Multi-Wilayah tertentu tentang apakah Titik Akses Multi-Wilayah yang ditentukan memiliki kebijakan kontrol akses yang memungkinkan akses publik.
GetMultiRegionAccessPointRoutes	(Diperlukan) `s3:GetMultiRegionAccessPointRoutes`	Diperlukan untuk mengembalikan konfigurasi routing untuk Titik Akses Multi-Region.
PutMultiRegionAccessPointPolicy	(Diperlukan) `s3:PutMultiRegionAccessPointPolicy`	Diperlukan untuk memperbarui kebijakan kontrol akses dari Titik Akses Multi-Wilayah yang ditentukan.
SubmitMultiRegionAccessPointRoutes	(Diperlukan) `s3:SubmitMultiRegionAccessPointRoutes`	Diperlukan untuk mengirimkan konfigurasi rute yang diperbarui untuk Titik Akses Multi-Wilayah.

Operasi dan izin pekerjaan Batch

Operasi pekerjaan (Operasi Batch) adalah operasi API S3 yang beroperasi pada jenis `job` sumber daya. Anda harus menentukan tindakan kebijakan S3 untuk operasi pekerjaan dalam kebijakan berbasis identitas IAM, bukan dalam kebijakan bucket.

Dalam kebijakan, `Resource` elemennya haruslah `job` ARN. Untuk informasi selengkapnya tentang format `Resource` elemen dan kebijakan contoh, lihatOperasi pekerjaan Batch.

Berikut ini adalah pemetaan operasi pekerjaan batch dan tindakan kebijakan yang diperlukan.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
DeleteJobTagging	(Diperlukan) `s3:DeleteJobTagging`	Diperlukan untuk menghapus tag dari pekerjaan Operasi Batch S3 yang ada.
DescribeJob	(Diperlukan) `s3:DescribeJob`	Diperlukan untuk mengambil parameter konfigurasi dan status untuk pekerjaan Operasi Batch.
GetJobTagging	(Diperlukan) `s3:GetJobTagging`	Diperlukan untuk mengembalikan kumpulan tag dari pekerjaan Operasi Batch S3 yang ada.
PutJobTagging	(Diperlukan) `s3:PutJobTagging`	Diperlukan untuk menempatkan atau mengganti tag pada pekerjaan Operasi Batch S3 yang ada.
UpdateJobPriority	(Diperlukan) `s3:UpdateJobPriority`	Diperlukan untuk memperbarui prioritas pekerjaan yang ada.
UpdateJobStatus	(Diperlukan) `s3:UpdateJobStatus`	Diperlukan untuk memperbarui status untuk pekerjaan yang ditentukan.

Operasi dan izin konfigurasi Lensa Penyimpanan S3

Operasi konfigurasi Lensa Penyimpanan S3 adalah operasi API S3 yang beroperasi pada jenis `storagelensconfiguration` sumber daya. Untuk informasi selengkapnya tentang cara mengonfigurasi operasi konfigurasi Lensa Penyimpanan S3, lihatMenyetel izin Lensa Penyimpanan Amazon S3.

Berikut ini adalah pemetaan operasi konfigurasi Lensa Penyimpanan S3 dan tindakan kebijakan yang diperlukan.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
DeleteStorageLensConfiguration	(Diperlukan) `s3:DeleteStorageLensConfiguration`	Diperlukan untuk menghapus konfigurasi Lensa Penyimpanan S3.
DeleteStorageLensConfigurationTagging	(Diperlukan) `s3:DeleteStorageLensConfigurationTagging`	Diperlukan untuk menghapus tag konfigurasi Lensa Penyimpanan S3.
GetStorageLensConfiguration	(Diperlukan) `s3:GetStorageLensConfiguration`	Diperlukan untuk mendapatkan konfigurasi Lensa Penyimpanan S3.
GetStorageLensConfigurationTagging	(Diperlukan) `s3:GetStorageLensConfigurationTagging`	Diperlukan untuk mendapatkan tag konfigurasi Lensa Penyimpanan S3.
PutStorageLensConfigurationTagging	(Diperlukan) `s3:PutStorageLensConfigurationTagging`	Diperlukan untuk meletakkan atau mengganti tag pada konfigurasi Lensa Penyimpanan S3 yang ada.

Lensa Penyimpanan S3 mengelompokkan operasi dan izin

Operasi grup Lensa Penyimpanan S3 adalah operasi API S3 yang beroperasi pada jenis `storagelensgroup` sumber daya. Untuk informasi selengkapnya tentang cara mengonfigurasi izin grup Lensa Penyimpanan S3, lihat. Izin grup Lensa Penyimpanan

Berikut ini adalah pemetaan operasi grup Lensa Penyimpanan S3 dan tindakan kebijakan yang diperlukan.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
DeleteStorageLensGroup	(Diperlukan) `s3:DeleteStorageLensGroup`	Diperlukan untuk menghapus grup Lensa Penyimpanan S3 yang ada.
GetStorageLensGroup	(Diperlukan) `s3:GetStorageLensGroup`	Diperlukan untuk mengambil detail konfigurasi grup Lensa Penyimpanan S3.
UpdateStorageLensGroup	(Diperlukan) `s3:UpdateStorageLensGroup`	Diperlukan untuk memperbarui grup Lensa Penyimpanan S3 yang ada.
CreateStorageLensGroup	(Diperlukan) `s3:CreateStorageLensGroup`	Diperlukan untuk membuat grup Lensa Penyimpanan baru.
CreateStorageLensGroup, TagResource	(Diperlukan)`s3:CreateStorageLensGroup` , `s3:TagResource`	Diperlukan untuk membuat grup Lensa Penyimpanan baru dengan tag.
ListStorageLensGroups	(Diperlukan) `s3:ListStorageLensGroups`	Diperlukan untuk mencantumkan semua grup Lensa Penyimpanan di Wilayah rumah Anda.
ListTagsForResource	(Diperlukan) `s3:ListTagsForResource`	Diperlukan untuk mencantumkan tag yang ditambahkan ke grup Lensa Penyimpanan Anda.
TagResource	(Diperlukan) `s3:TagResource`	Diperlukan untuk menambah atau memperbarui tag grup Lensa Penyimpanan untuk grup Lensa Penyimpanan yang ada.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
UntagResource	(Diperlukan) `s3:UntagResource`	Diperlukan untuk menghapus tag dari grup Lensa Penyimpanan.

S3 Access Memberikan operasi dan izin instans

Operasi instans S3 Access Grants adalah operasi API S3 yang beroperasi pada jenis sumber daya. `accessgrantsinstance` Instance S3 Access Grants adalah wadah logis untuk hibah akses Anda. Untuk informasi selengkapnya tentang bekerja dengan instans S3 Access Grants, lihat. Bekerja dengan instans S3 Access Grants

Berikut ini adalah pemetaan operasi konfigurasi instans S3 Access Grants dan tindakan kebijakan yang diperlukan.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
AssociateAccessGrantsIdentityCenter	(Diperlukan) `s3:AssociateAccessGrantsIdentityCenter`	Diperlukan untuk mengaitkan AWS IAM Identity Center instance dengan instans S3 Access Grants Anda, sehingga memungkinkan Anda membuat hibah akses untuk pengguna dan grup di direktori identitas perusahaan Anda. Anda juga harus memiliki izin berikut: `sso:CreateApplication` , `sso:PutApplicationGrant` , dan `sso:PutApplicationAuthenticationMethod` .

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
CreateAccessGrantsInstance	(Diperlukan) `s3:CreateAccessGrantsInstance`	Diperlukan untuk membuat instance S3 Access Grants (`accessgrantsinstance` resource) yang merupakan wadah untuk hibah akses individual Anda. Untuk mengaitkan AWS IAM Identity Center instance dengan instans S3 Access Grants, Anda juga harus memiliki`sso:DescribeInstance` ,, `sso:CreateApplication` `sso:PutApplicationGrant` , dan `sso:PutApplicationAuthenticationMethod` izin.
DeleteAccessGrantsInstance	(Diperlukan) `s3:DeleteAccessGrantsInstance`	Diperlukan untuk menghapus instance (`accessgrantsinstance` sumber daya) S3 Access Grants dari akun Wilayah AWS Anda.
DeleteAccessGrantsInstanceResourcePolicy	(Diperlukan) `s3:DeleteAccessGrantsInstanceResourcePolicy`	Diperlukan untuk menghapus kebijakan sumber daya untuk instance S3 Access Grants Anda.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
DissociateAccessGrantsIdentityCenter	(Diperlukan) `s3:DissociateAccessGrantsIdentityCenter`	Diperlukan untuk memisahkan AWS IAM Identity Center instance dari instance S3 Access Grants Anda. Anda juga harus memiliki izin berikut: `sso:DeleteApplication`
GetAccessGrantsInstance	(Diperlukan) `s3:GetAccessGrantsInstance`	Diperlukan untuk mengambil instance S3 Access Grants untuk akun Wilayah AWS Anda.
GetAccessGrantsInstanceForPrefix	(Diperlukan) `s3:GetAccessGrantsInstanceForPrefix`	Diperlukan untuk mengambil instance S3 Access Grants yang berisi awalan tertentu.
GetAccessGrantsInstanceResourcePolicy	(Diperlukan) `s3:GetAccessGrantsInstanceResourcePolicy`	Diperlukan untuk mengembalikan kebijakan sumber daya instans S3 Access Grants Anda.
ListAccessGrantsInstances	(Diperlukan) `s3:ListAccessGrantsInstances`	Diperlukan untuk mengembalikan daftar instans S3 Access Grants di akun Anda.
PutAccessGrantsInstanceResourcePolicy	(Diperlukan) `s3:PutAccessGrantsInstanceResourcePolicy`	Diperlukan untuk memperbarui kebijakan sumber daya instance S3 Access Grants.

Akses S3 Memberikan operasi dan izin lokasi

Operasi lokasi S3 Access Grants adalah operasi API S3 yang beroperasi pada jenis sumber daya. `accessgrantslocation` Untuk informasi selengkapnya tentang bekerja dengan lokasi S3 Access Grants, lihat. Bekerja dengan lokasi S3 Access Grants

Berikut ini adalah pemetaan operasi konfigurasi lokasi S3 Access Grants dan tindakan kebijakan yang diperlukan.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
CreateAccessGrantsLocation	(Diperlukan) `s3:CreateAccessGrantsLocation`	Diperlukan untuk mendaftarkan lokasi di instans S3 Access Grants Anda (membuat `accessgrantslocation` sumber daya). Anda juga harus memiliki izin berikut untuk peran IAM yang ditentukan: `iam:PassRole`
DeleteAccessGrantsLocation	(Diperlukan) `s3:DeleteAccessGrantsLocation`	Diperlukan untuk menghapus lokasi terdaftar dari instans S3 Access Grants Anda.
GetAccessGrantsLocation	(Diperlukan) `s3:GetAccessGrantsLocation`	Diperlukan untuk mengambil detail lokasi tertentu yang terdaftar di instans S3 Access Grants Anda.
ListAccessGrantsLocations	(Diperlukan) `s3:ListAccessGrantsLocations`	Diperlukan untuk mengembalikan daftar lokasi yang terdaftar di instans S3 Access Grants Anda.
UpdateAccessGrantsLocation	(Diperlukan) `s3:UpdateAccessGrantsLocation`	Diperlukan untuk memperbarui peran IAM dari lokasi terdaftar

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
		di instans S3 Access Grants Anda.

S3 Access Grants memberikan operasi dan izin

Operasi hibah S3 Access Grants adalah operasi API S3 yang beroperasi pada jenis sumber daya. `accessgrant` Untuk informasi lebih lanjut tentang bekerja dengan hibah individu menggunakan S3 Access Grants, lihat. Bekerja dengan hibah di S3 Access Grants

Berikut ini adalah pemetaan operasi konfigurasi hibah S3 Access Grants dan tindakan kebijakan yang diperlukan.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
CreateAccessGrant	(Diperlukan) `s3:CreateAccessGrant`	Diperlukan untuk membuat hibah individu (`accessgrant` sumber daya) untuk pengguna atau grup di instance S3 Access Grants Anda. Anda juga harus memiliki izin berikut: Untuk identitas direktori apa pun — `sso:DescribeInstance` dan `sso:DescribeApplication` Untuk pengguna direktori — `identitystore:DescribeUser`
DeleteAccessGrant	(Diperlukan) `s3:DeleteAccessGrant`	Diperlukan untuk menghapus hibah akses individual (`accessgrant` sumber

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
		daya) dari instance S3 Access Grants Anda.
GetAccessGrant	(Diperlukan) `s3:GetAccessGrant`	Diperlukan untuk mendapatkan detail tentang hibah akses individual di instans Hibah Akses S3 Anda.
ListAccessGrants	(Diperlukan) `s3:ListAccessGrants`	Diperlukan untuk mengembalikan daftar hibah akses individual di instance Hibah Akses S3 Anda.
ListCallerAccessGrants	(Diperlukan) `s3:ListCallerAccessGrants`	Diperlukan untuk mencantumkan hibah akses yang memberikan akses pemanggil ke data Amazon S3 melalui S3 Access Grants.

Operasi dan izin akun

Operasi akun adalah operasi API S3 yang beroperasi pada tingkat akun. Akun bukanlah jenis sumber daya yang ditentukan oleh Amazon S3. Anda harus menentukan tindakan kebijakan S3 untuk operasi akun dalam kebijakan berbasis identitas IAM, bukan dalam kebijakan bucket.

Dalam kebijakan, `Resource` elemennya harus`"*"`. Untuk informasi selengkapnya tentang kebijakan contoh, lihatOperasi akun.

Berikut ini adalah pemetaan operasi akun dan tindakan kebijakan yang diperlukan.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
CreateJob	(Diperlukan) `s3:CreateJob`	Diperlukan untuk membuat pekerjaan Operasi Batch S3 baru.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
CreateStorageLensGroup	(Diperlukan) `s3:CreateStorageLensGroup`	Diperlukan untuk membuat grup Lensa Penyimpanan S3 baru dan mengaitkannya dengan Akun AWS ID yang ditentukan.
	(Diperlukan secara kondisional) `s3:TagResource`	Diperlukan jika Anda ingin membuat grup Lensa Penyimpanan S3 dengan tag AWS sumber daya.
DeletePublicAccessBlock(Tingkat akun)	(Diperlukan) `s3:PutAccountPublicAccessBlock`	Diperlukan untuk menghapus konfigurasi blokir akses publik dari file Akun AWS.
GetAccessPoint	(Diperlukan) `s3:GetAccessPoint`	Diperlukan untuk mengambil informasi konfigurasi tentang titik akses yang ditentukan.
GetAccessPointPolicy(Tingkat akun)	(Diperlukan) `s3:GetAccountPublicAccessBlock`	Diperlukan untuk mengambil konfigurasi akses publik blok untuk file Akun AWS.
ListAccessPoints	(Diperlukan) `s3:ListAccessPoints`	Diperlukan untuk mencantumkan titik akses bucket S3 yang dimiliki oleh file Akun AWS.
ListAccessPointsForObjectLambda	(Diperlukan) `s3:ListAccessPointsForObjectLambda`	Diperlukan untuk mencantumkan Titik Akses Objek Lambda.
ListBuckets	(Diperlukan) `s3:ListAllMyBuckets`	Diperlukan untuk mengembalikan daftar semua bucket yang dimiliki oleh pengirim permintaan yang diautentikasi.

Operasi API	Tindakan kebijakan	Deskripsi tindakan kebijakan
ListJobs	(Diperlukan) `s3:ListJobs`	Diperlukan untuk membuat daftar pekerjaan dan pekerjaan saat ini yang telah berakhir baru-baru ini.
ListMultiRegionAccessPoints	(Diperlukan) `s3:ListMultiRegionAccessPoints`	Diperlukan untuk mengembalikan daftar Titik Akses Multi-Wilayah yang saat ini terkait dengan yang ditentukan Akun AWS.
ListStorageLensConfigurations	(Diperlukan) `s3:ListStorageLensConfigurations`	Diperlukan untuk mendapatkan daftar konfigurasi Lensa Penyimpanan S3 untuk file. Akun AWS
ListStorageLensGroups	(Diperlukan) `s3:ListStorageLensGroups`	Diperlukan untuk mencantumkan semua grup Lensa Penyimpanan S3 di rumah Wilayah AWS yang ditentukan.
PutPublicAccessBlock(Tingkat akun)	(Diperlukan) `s3:PutAccountPublicAccessBlock`	Diperlukan untuk membuat atau memodifikasi konfigurasi blokir akses publik untuk file Akun AWS.
PutStorageLensConfiguration	(Diperlukan) `s3:PutStorageLensConfiguration`	Diperlukan untuk menempatkan konfigurasi Lensa Penyimpanan S3.

Kebijakan dan izin di Amazon S3

Halaman ini memberikan ikhtisar kebijakan bucket dan pengguna di Amazon S3 dan menjelaskan elemen dasar kebijakan AWS Identity and Access Management (IAM). Setiap elemen yang

tercantum berkaitan dengan perincian lebih lanjut tentang elemen tersebut dan contoh-contoh cara penggunaannya.

Untuk daftar lengkap tindakan, sumber daya, dan ketentuan Amazon S3, lihat Tindakan, sumber daya, dan kunci kondisi untuk Amazon S3 di Referensi Otorisasi Layanan.

Untuk informasi selengkapnya tentang izin operasi S3 API menurut jenis sumber daya S3, lihat. Izin yang diperlukan untuk operasi API Amazon S3

Secara mendasar, sebuah kebijakan berisi elemen-elemen berikut ini:

- Resource- Bucket Amazon S3, objek, titik akses, atau pekerjaan yang berlaku untuk kebijakan tersebut. Gunakan Amazon Resource Name (ARN) dari bucket, objek, titik akses, atau job untuk mengidentifikasi sumber daya.

  Contoh untuk operasi tingkat bucket:

  `"Resource": "arn:aws:s3:::bucket_name"`

  Contoh operasi tingkat objek:

  - `"Resource": "arn:aws:s3:::bucket_name/*"`untuk semua benda di ember.
  - `"Resource": "arn:aws:s3:::bucket_name/prefix/*"`untuk objek di bawah awalan tertentu di ember.

  Untuk informasi selengkapnya, lihat Sumber daya kebijakan untuk Amazon S3.

- Actions— Untuk setiap sumber daya, Amazon S3 mendukung serangkaian operasi. Anda mengidentifikasi operasi sumber daya yang Anda izinkan (atau tolak) dengan menggunakan kata kunci tindakan.

  Misalnya, `s3:ListBucket` izin memungkinkan pengguna untuk menggunakan operasi Amazon S3 ListObjectsV2. (`s3:ListBucket`Izin adalah kasus di mana nama tindakan tidak dipetakan langsung ke nama operasi.) Untuk informasi selengkapnya tentang penggunaan tindakan Amazon S3, lihat Tindakan kebijakan untuk Amazon S3. Untuk daftar lengkap tindakan Amazon S3, lihat Tindakan di Referensi API Layanan Penyimpanan Sederhana Amazon.

- Effect— Apa efeknya ketika pengguna meminta tindakan tertentu—ini bisa berupa salah satu atau. `Allow` `Deny`

  Jika Anda tidak secara eksplisit memberikan akses ke (mengizinkan) sumber daya, akses akan ditolak secara implisit. Anda juga dapat secara eksplisit menolak akses ke sumber daya. Anda

dapat melakukan ini untuk memastikan bahwa pengguna tidak dapat mengakses sumber daya, bahkan jika kebijakan yang berbeda memberikan akses. Untuk informasi selengkapnya, lihat Elemen Kebijakan IAM JSON: Efek dalam Panduan Pengguna IAM.

- Principal— Akun atau pengguna yang diizinkan mengakses tindakan dan sumber daya dalam pernyataan. Dalam kebijakan bucket, pengguna utama adalah pengguna, akun, layanan, atau entitas lain yang merupakan penerima izin ini. Untuk informasi selengkapnya, lihat Prinsipal untuk kebijakan bucket.
- Condition— Ketentuan kapan kebijakan berlaku. Anda dapat menggunakan kunci AWS-wide dan kunci khusus Amazon S3 untuk menentukan kondisi dalam kebijakan akses Amazon S3. Untuk informasi selengkapnya, lihat Contoh kebijakan bucket menggunakan tombol kondisi.

Contoh kebijakan bucket berikut menunjukkan`Effect`,, `Principal`Action, dan `Resource` elemen. Kebijakan ini memungkinkan*Akua*, pengguna dalam akun*123456789012*, `s3:GetObjects3:GetBucketLocation`, dan izin `s3:ListBucket` Amazon S3 di bucket. *amzn-s3-demo-bucket1*

```
{
    "Version": "2012-10-17",
    "Id": "ExamplePolicy01",
    "Statement": [
        {
            "Sid": "ExampleStatement01",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/Akua"
            },
            "Action": [
                "s3:GetObject",
                "s3:GetBucketLocation",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket1/*",
                "arn:aws:s3:::amzn-s3-demo-bucket1"
            ]
        }
    ]
}
```

Untuk informasi bahasa kebijakan selengkapnya, lihat Kebijakan dan izin dalam referensi kebijakan IAM dan IAM JSON di Panduan Pengguna IAM.

Delegasi izin

Jika Akun AWS memiliki sumber daya, ia dapat memberikan izin tersebut kepada yang lain. Akun AWS Akun tersebut kemudian dapat mendelegasikan izin-izin tersebut, atau sebagian izin, kepada pengguna dalam akun. Ini disebut sebagai delegasi izin. Tetapi akun yang menerima izin dari akun lain tidak dapat mendelegasikan izin lintas akun ke akun lain. Akun AWS

bucket Amazon S3 dan kepemilikan objek

bucket dan objek adalah sumber daya Amazon S3. Secara bawaan, hanya pemilik sumber daya yang dapat mengakses sumber daya ini. Pemilik sumber daya mengacu pada Akun AWS yang menciptakan sumber daya. Misalnya:

- Akun AWS Yang Anda gunakan untuk membuat bucket dan mengunggah objek memiliki sumber daya tersebut.
- Jika Anda mengunggah objek menggunakan pengguna AWS Identity and Access Management (IAM) atau kredensi peran, pengguna atau peran Akun AWS tersebut memiliki objek tersebut.
- Pemilik bucket dapat memberikan izin lintas akun kepada Akun AWS lain (atau pengguna di akun lain) untuk unggah objek. Dalam hal ini, Akun AWS yang unggah objek tersebut adalah pemilik objek tersebut. Pemilik bucket tidak memiliki izin pada objek yang dimiliki akun lain, dengan pengecualian berikut:
  - Pemilik bucket membayar tagihan. Pemilik bucket dapat menolak akses ke objek apa pun, atau menghapus objek apa pun di dalam bucket, tanpa memandang siapa yang memilikinya.
  - Pemilik bucket dapat mengarsipkan objek apa pun atau memulihkan objek yang diarsipkan, tanpa memandang siapa yang memilikinya. Pengarsipan mengacu pada kelas penyimpanan yang digunakan untuk menyimpan objek. Untuk informasi selengkapnya, lihat Mengelola siklus hidup objek.

Autentikasi kepemilikan dan permintaan

Semua permintaan ke bucket akan diautentikasi atau tidak diautentikasi. Permintaan yang diautentikasi harus menyertakan nilai tanda tangan yang mengautentikasi pengirim permintaan, permintaan yang tidak diautentikasi tidak harus menyertakannya. Untuk informasi selengkapnya tentang autentikasi permintaan, lihat Membuat permintaan di Referensi API Amazon S3.

Pemilik bucket dapat mengizinkan permintaan yang tidak diauntentikasi. Misalnya, PutObjectpermintaan yang tidak diautentikasi diizinkan jika bucket memiliki kebijakan bucket publik, atau saat ACL bucket memberikan `WRITE` atau `FULL_CONTROL` mengakses `All Users` grup atau pengguna anonim secara khusus. Untuk informasi selengkapnya tentang kebijakan bucket publik dan daftar kontrol akses publik (ACLs), lihatArti "publik".

Semua permintaan yang tidak diautentikasi dibuat oleh pengguna anonim. Pengguna ini diwakili ACLs oleh ID pengguna kanonik tertentu. `65a011a29cdf8ec533ec3d1ccaae921c` Jika objek diunggah ke bucket melalui permintaan yang tidak terotentikasi, pengguna anonim memiliki objek. ACL objek bawaan memberikan `FULL_CONTROL` kepada pengguna anonim sebagai pemilik objek. Oleh karena itu, Amazon S3 mengizinkan permintaan yang tidak diautentikasi untuk mengambil objek atau memodifikasi ACL-nya.

Untuk mencegah objek dimodifikasi oleh pengguna anonim, sebaiknya Anda tidak menerapkan kebijakan bucket yang mengizinkan penulisan publik anonim ke bucket Anda atau penggunaan ACLs yang memungkinkan pengguna anonim menulis akses ke bucket Anda. Anda dapat menerapkan perilaku yang direkomendasikan ini dengan menggunakan Blokir Akses Publik Amazon S3.

Untuk informasi lebih lanjut tentang pemblokiran akses publik, lihat Melakukan blok akses publik ke penyimpanan Amazon S3 Anda. Untuk informasi selengkapnya tentang ACLs, lihat Gambaran umum daftar kontrol akses (ACL).

> **Important**
>
> Kami menyarankan agar Anda tidak menggunakan kredensi pengguna Akun AWS root untuk membuat permintaan yang diautentikasi. Sebagai gantinya, buat peran IAM dan berikan akses penuh kepada peran tersebut. Kami merujuk ke pengguna dengan peran ini sebagai pengguna administrator. Anda dapat menggunakan kredensil yang ditetapkan ke peran administrator, bukan kredensil pengguna Akun AWS root, untuk berinteraksi dengan AWS dan melakukan tugas, seperti membuat bucket, membuat pengguna, dan memberikan izin. Untuk informasi selengkapnya, lihat AWS kredensi keamanan dan praktik terbaik Keamanan di IAM di Panduan Pengguna IAM.

Kebijakan bucket untuk Amazon S3

Kebijakan bucket adalah kebijakan berbasis sumber daya yang dapat Anda gunakan untuk memberikan izin akses ke bucket Amazon S3 dan objek di dalamnya. Hanya pemilik bucket yang dapat mengaitkan kebijakan dengan bucket. Izin yang dipasang pada bucket berlaku untuk semua objek di bucket yang dimiliki oleh pemilik bucket. Izin ini tidak berlaku untuk objek yang dimiliki oleh orang lain Akun AWS.

S3 Object Ownership adalah setelan tingkat ember Amazon S3 yang dapat Anda gunakan untuk mengontrol kepemilikan objek yang diunggah ke bucket Anda dan untuk menonaktifkan atau mengaktifkan daftar kontrol akses (). ACLs Secara default, Kepemilikan Objek disetel ke setelan diberlakukan pemilik Bucket dan semuanya ACLs dinonaktifkan. Pemilik bucket memiliki semua objek di bucket dan mengelola akses ke data secara eksklusif menggunakan kebijakan.

Kebijakan bucket menggunakan bahasa kebijakan berbasis JSON AWS Identity and Access Management (IAM). Anda dapat menggunakan kebijakan bucket untuk menambah atau menolak izin objek dalam bucket. Kebijakan bucket dapat mengizinkan atau menolak permintaan berdasarkan elemen dalam kebijakan. Elemen-elemen ini termasuk pemohon, tindakan S3, sumber daya, dan aspek atau kondisi permintaan (seperti alamat IP yang digunakan untuk membuat permintaan).

Misalnya, Anda dapat membuat kebijakan bucket yang melakukan langkah berikut:

- Berikan izin lintas akun lainnya untuk mengunggah objek ke bucket S3 Anda
- Memastikan bahwa Anda, pemilik bucket, memiliki kendali penuh atas objek yang diunggah

Untuk informasi selengkapnya, lihat Contoh kebijakan bucket Amazon S3.

> **Important**
>
> Anda tidak dapat menggunakan kebijakan bucket untuk mencegah penghapusan atau transisi oleh aturan Siklus Hidup S3. Misalnya, meskipun kebijakan bucket Anda menolak semua tindakan untuk semua prinsipal, konfigurasi Siklus Hidup S3 Anda tetap berfungsi seperti biasa.

Topik di bagian ini memberikan contoh dan menunjukkan cara menambahkan kebijakan bucket di konsol S3. Untuk informasi tentang kebijakan berbasis identitas, lihat. Kebijakan berbasis identitas

untuk Amazon S3 Untuk informasi tentang bahasa kebijakan bucket, lihat Kebijakan dan izin di Amazon S3.

Untuk informasi selengkapnya tentang izin operasi S3 API menurut jenis sumber daya S3, lihat. Izin yang diperlukan untuk operasi API Amazon S3

Topik

- Menambahkan kebijakan bucket dengan menggunakan konsol Amazon S3
- Megendalikan akses dari titik akhir VPC dengan kebijakan bucket
- Contoh kebijakan bucket Amazon S3
- Contoh kebijakan bucket menggunakan tombol kondisi

Menambahkan kebijakan bucket dengan menggunakan konsol Amazon S3

Anda dapat menggunakan Pembuat Kebijakan AWS dan konsol Amazon S3 untuk menambahkan kebijakan bucket baru, atau mengedit kebijakan bucket yang sudah ada. Kebijakan bucket adalah kebijakan berbasis sumber daya AWS Identity and Access Management (IAM). Anda menambahkan kebijakan bucket ke bucket untuk memberikan izin akses kepada pengguna lain Akun AWS atau IAM untuk bucket dan objek di dalamnya. Izin objek hanya berlaku untuk objek yang dibuat oleh pemilik bucket. Untuk informasi lebih lanjut tentang kebijakan bucket, lihat Identity and Access Management untuk Amazon S3.

Pastikan Anda menyelesaikan peringatan keamanan, kesalahan, peringatan umum, dan saran dari AWS Identity and Access Management Access Analyzer sebelum Anda menyimpan kebijakan Anda. Penganalisis Akses IAM menjalankan pemeriksaan kebijakan untuk memvalidasi kebijakan Anda terhadap tata bahasa kebijakan IAM dan praktik terbaik. Pemeriksaan ini menghasilkan temuan dan memberikan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang fungsional dan sesuai dengan praktik terbaik keamanan. Untuk mempelajari validasi kebijakan menggunakan Penganalisis Akses IAM lebih lanjut, lihat Memvalidasi kebijakan Penganalisis Akses IAM di Panduan Pengguna IAM. Untuk melihat daftar peringatan, kesalahan, dan saran yang ditampilkan oleh Penganalisis Akses IAM, lihat referensi pemeriksaan kebijakan Penganalisis Akses IAM.

Untuk panduan tentang pemecahan masalah kesalahan dengan kebijakan, lihat. Memecahkan masalah kesalahan akses ditolak (403 Forbidden) di Amazon S3

Untuk membuat atau mengedit kebijakan bucket

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum atau ember Direktori.
3. Dalam daftar bucket, pilih nama bucket yang ingin Anda buat kebijakan bucket atau kebijakan bucket yang ingin Anda edit.
4. Pilih tab Izin.
5. Di Bawah Kebijakan bucket, pilih Edit. Halaman Edit kebijakan bucket akan muncul.
6. Di halaman Edit kebijakan bucket, lakukan salah satu hal berikut:

   - Untuk melihat contoh kebijakan bucket, pilih Contoh kebijakan. Atau lihat Contoh kebijakan bucket Amazon S3 di Panduan Pengguna Amazon S3.
   - Untuk membuat kebijakan secara otomatis, atau mengedit JSON di bagian Kebijakan, pilih Pembuat kebijakan.

   Jika Anda memilih Policy Generator, AWS Policy Generator akan terbuka di jendela baru.

   a. Pada halaman AWS Pembuat Kebijakan, untuk Pilih Jenis Kebijakan, pilih Kebijakan Bucket S3.

   b. Tambahkan pernyataan dengan memasukkan informasi di bidang yang disediakan, lalu pilih Tambah Pernyataan. Ulangi langkah ini sebanyak jumlah pernyataan yang ingin Anda tambahkan. Untuk informasi selengkapnya tentang persyaratan kebijakan, lihat Referensi kebijakan IAM JSON di Panduan Pengguna IAM.

      > **Note**
      >
      > Untuk kenyamanan Anda, halaman Edit kebijakan bucket menampilkan Bucket ARN (Nama Sumber Daya Amazon) dari bucket saat ini di atas bidang teks Kebijakan. Anda dapat menyalin ARN ini untuk digunakan dalam pernyataan di halaman Pembuat Kebijakan AWS .

   c. Setelah Anda selesai menambahkan pernyataan, pilih Buat Kebijakan.

   d. Salin teks kebijakan yang dihasilkan, pilih Tutup, dan kembali ke halaman Edit kebijakan bucket di konsol Amazon S3.

7. Di kotak Kebijakan, edit kebijakan yang ada atau tempel kebijakan bucket dari AWS Policy Generator. Pastikan peringatan keamanan, kesalahan, peringatan umum, dan saran telah ditangani sebelum menyimpan kebijakan.

   > **Note**
   >
   > Kebijakan bucket dibatasi hingga ukuran 20 KB.

8. (Opsional) Pilih Pratinjau akses eksternal di sudut kanan bawah untuk melihat bagaimana kebijakan baru memengaruhi publik dan akses lintas akun ke sumber daya Anda. Sebelum Anda menyimpan kebijakan Anda, Anda dapat memeriksa apakah itu memperkenalkan temuan Penganalisa Akses IAM baru atau menyelesaikan temuan yang ada. Jika Anda tidak melihat penganalisis aktif, pilih Buka Penganalisis Akses untuk membuat penganalisis akun di Penganalisis Akses IAM. Untuk informasi selengkapnya, lihat Pratinjau akses dalam Panduan Pengguna IAM.
9. Pilih Simpan perubahan, yang mengembalikan Anda ke tab Izin.

Megendalikan akses dari titik akhir VPC dengan kebijakan bucket

Anda dapat menggunakan kebijakan bucket Amazon S3 untuk mengontrol akses ke bucket dari titik akhir virtual private cloud (VPC) tertentu atau spesifik. VPCs Bagian ini berisi contoh kebijakan bucket yang dapat Anda gunakan untuk mengontrol akses bucket Amazon S3 dari titik akhir VPC. Untuk mempelajari cara mengatur titik akhir VPC, lihat Titik Akhir VPC dalam Panduan Pengguna VPC.

VPC memungkinkan Anda meluncurkan AWS sumber daya ke jaringan virtual yang Anda tentukan. Endpoint VPC memungkinkan Anda membuat koneksi pribadi antara VPC Anda dan VPC lainnya. Layanan AWS Koneksi pribadi ini tidak memerlukan akses melalui internet, melalui koneksi jaringan pribadi virtual (VPN), melalui instance NAT, atau melalui AWS Direct Connect.

Titik akhir VPC untuk Amazon S3 adalah entitas logika dalam VPC yang memungkinkan konektivitas hanya ke Amazon S3. Rute titik akhir VPC mengarahkan permintaan ke Amazon S3 dan mengarahkan respons kembali ke VPC. Titik akhir VPC hanya mengubah bagaimana permintaan diarahkan. Titik akhir publik Amazon S3 dan nama DNS akan terus bekerja dengan titik akhir VPC. Untuk informasi penting tentang penggunaan titik akhir VPC dengan Amazon S3, lihat Titik akhir Gateway dan titik akhir Gateway untuk Amazon S3 di Panduan Pengguna VPC.

Titik akhir VPC untuk Amazon S3 menyediakan dua cara untuk mengontrol akses ke data Amazon S3 Anda:

- Anda dapat mengontrol permintaan, pengguna, atau grup yang diizinkan melalui titik akhir VPC tertentu. Untuk informasi tentang jenis kontrol akses ini, lihat Mengontrol akses ke titik akhir VPC menggunakan kebijakan titik akhir di Panduan Pengguna VPC.
- Anda dapat mengontrol titik akhir VPCs atau VPC mana yang memiliki akses ke bucket Anda dengan menggunakan kebijakan bucket Amazon S3. Untuk contoh jenis kontrol akses kebijakan bucket seperti ini, lihat topik-topik yang membahas tentang pembatasan akses.

Topik

- Membatasi akses ke titik akhir VPC kustom
- Membatasi Akses ke VPC Tertentu

> **Important**
>
> Saat menerapkan kebijakan bucket Amazon S3 untuk titik akhir VPC yang dijelaskan di bagian ini, Anda dapat memblokir akses ke bucket secara tidak sengaja. Izin bucket yang dimaksudkan untuk secara kustom membatasi akses bucket ke koneksi yang berasal dari titik akhir VPC Anda dapat memblokir semua koneksi ke bucket tersebut. Untuk informasi tentang cara memperbaiki masalah ini, lihat Bagaimana cara memperbaiki kebijakan bucket saya jika memiliki ID titik akhir VPC atau VPC yang salah? di pusat AWS Dukungan pengetahuan.

Membatasi akses ke titik akhir VPC kustom

Berikut ini adalah contoh kebijakan bucket Amazon S3 yang membatasi akses ke bucket tertentu, `awsexamplebucket1`, hanya dari titik akhir VPC dengan ID `vpce-1a2b3c4d`. Jika titik akhir yang ditentukan tidak digunakan, kebijakan menolak semua akses ke bucket. `aws:SourceVpce`Kondisi menentukan titik akhir. `aws:SourceVpce`Kondisi ini tidak memerlukan Nama Sumber Daya Amazon (ARN) untuk sumber daya titik akhir VPC, hanya ID titik akhir VPC. Untuk informasi lebih lanjut tentang penggunaan kondisi dalam kebijakan, lihat Contoh kebijakan bucket menggunakan tombol kondisi.

> **Important**
>
> - Sebelum menggunakan kebijakan contoh berikut ini, ganti ID titik akhir VPC dengan nilai yang sesuai untuk kasus penggunaan Anda. Jika tidak, Anda tidak akan dapat mengakses bucket Anda.

- Kebijakan ini menonaktifkan akses konsol ke bucket yang ditentukan karena permintaan konsol tidak berasal dari titik akhir VPC yang ditentukan.

```
{
   "Version": "2012-10-17",
   "Id": "Policy1415115909152",
   "Statement": [
     {
       "Sid": "Access-to-specific-VPCE-only",
       "Principal": "*",
       "Action": "s3:*",
       "Effect": "Deny",
       "Resource": ["arn:aws:s3:::awsexamplebucket1",
                    "arn:aws:s3:::awsexamplebucket1/*"],
       "Condition": {
         "StringNotEquals": {
           "aws:SourceVpce": "vpce-1a2b3c4d"
         }
       }
     }
   ]
}
```

Membatasi Akses ke VPC Tertentu

Anda dapat membuat kebijakan bucket yang membatasi akses ke VPC tertentu dengan menggunakan kondisi `aws:SourceVpc` tersebut. Hal ini berguna jika Anda memiliki beberapa titik akhir VPC yang dikonfigurasi dalam VPC yang sama, dan Anda ingin mengelola akses ke bucket Amazon S3 Anda untuk semua titik akhir Anda. Berikut ini adalah contoh kebijakan yang menolak akses ke `awsexamplebucket1` dan objeknya dari siapa pun di luar VPC`vpc-111bbb22`. Jika VPC yang ditentukan tidak digunakan, kebijakan menolak semua akses ke bucket. Pernyataan ini tidak memberikan akses ke bucket. Untuk memberikan akses, Anda harus menambahkan `Allow` pernyataan terpisah. Kunci `vpc-111bbb22` kondisi tidak memerlukan ARN untuk sumber daya VPC, hanya ID VPC.

Important

- Sebelum menggunakan kebijakan contoh berikut, ganti ID VPC dengan nilai yang sesuai untuk kasus penggunaan Anda. Jika tidak, Anda tidak akan dapat mengakses bucket Anda.
- Kebijakan ini menonaktifkan akses konsol ke bucket yang ditentukan karena permintaan konsol tidak berasal dari VPC yang ditentukan.

```
{
   "Version": "2012-10-17",
   "Id": "Policy1415115909153",
   "Statement": [
     {
       "Sid": "Access-to-specific-VPC-only",
       "Principal": "*",
       "Action": "s3:*",
       "Effect": "Deny",
       "Resource": ["arn:aws:s3:::awsexamplebucket1",
                    "arn:aws:s3:::awsexamplebucket1/*"],
       "Condition": {
         "StringNotEquals": {
           "aws:SourceVpc": "vpc-111bbb22"
         }
       }
     }
   ]
}
```

Untuk informasi tentang cara membatasi akses ke bucket Anda berdasarkan spesifik IPs atau VPCs, lihat Bagaimana cara mengizinkan hanya titik akhir atau alamat IP VPC tertentu untuk mengakses bucket Amazon S3 saya? di pusat AWS re:Post pengetahuan.

Contoh kebijakan bucket Amazon S3

Dengan kebijakan bucket Amazon S3, Anda dapat mengamankan akses ke objek di bucket, sehingga hanya pengguna dengan izin yang sesuai yang dapat mengaksesnya. Anda bahkan dapat mencegah pengguna yang diautentikasi tanpa izin yang sesuai untuk mengakses sumber daya Amazon S3 Anda.

Bagian ini menyajikan contoh kasus penggunaan umum untuk kebijakan bucket. Kebijakan sampel ini digunakan *amzn-s3-demo-bucket* sebagai nilai sumber daya. Untuk menguji kebijakan ini, ganti *user input placeholders* dengan informasi Anda sendiri (seperti nama bucket Anda).

Untuk memberikan atau menolak izin ke sekumpulan objek, Anda dapat menggunakan karakter wildcard (*) di Amazon Resource Names (ARNs) dan nilai lainnya. Misalnya, Anda dapat mengontrol akses ke grup objek yang dimulai dengan awalan umum atau diakhiri dengan ekstensi tertentu, seperti `.html`.

Untuk informasi selengkapnya tentang bahasa kebijakan AWS Identity and Access Management (IAM), lihatKebijakan dan izin di Amazon S3.

Untuk informasi selengkapnya tentang izin operasi S3 API menurut jenis sumber daya S3, lihat. Izin yang diperlukan untuk operasi API Amazon S3

> **Note**
>
> Saat menguji izin dengan menggunakan konsol Amazon S3, Anda harus memberikan izin tambahan yang diperlukan konsol—`s3:ListAllMyBuckets`, `s3:GetBucketLocation`, dan. `s3:ListBucket`. Untuk panduan contoh yang memberikan izin kepada pengguna dan menguji izin tersebut dengan menggunakan konsol, lihat. Mengontrol akses ke bucket dengan kebijakan pengguna

Sumber daya tambahan untuk membuat kebijakan bucket meliputi:

- Untuk daftar tindakan kebijakan IAM, sumber daya, dan kunci kondisi yang dapat Anda gunakan saat membuat kebijakan bucket, lihat Kunci tindakan, sumber daya, dan kondisi untuk Amazon S3 di Referensi Otorisasi Layanan.
- Untuk informasi selengkapnya tentang izin operasi S3 API menurut jenis sumber daya S3, lihat. Izin yang diperlukan untuk operasi API Amazon S3
- Untuk panduan tentang membuat kebijakan S3 Anda, lihat Menambahkan kebijakan bucket dengan menggunakan konsol Amazon S3.
- Untuk memecahkan masalah kesalahan dengan kebijakan, lihat Memecahkan masalah kesalahan akses ditolak (403 Forbidden) di Amazon S3.

Jika Anda mengalami masalah saat menambahkan atau memperbarui kebijakan, lihat Mengapa saya mendapatkan kesalahan “Prinsipal tidak valid dalam kebijakan” saat saya mencoba memperbarui kebijakan bucket Amazon S3 saya? di pusat AWS re:Post pengetahuan.

Topik

- Memberikan izin baca-saja kepada pengguna anonim publik
- Membutuhkan enkripsi
- Mengelola ember menggunakan kaleng ACLs
- Mengelola akses objek dengan penandaan objek
- Mengelola akses objek dengan menggunakan kunci kondisi global
- Mengelola akses sesuai permintaan HTTP atau HTTPS
- Mengelola akses pengguna ke folder tertentu
- Mengelola akses log akses
- Mengelola akses ke Amazon CloudFront OAI
- Mengelola akses untuk Lensa Penyimpanan Amazon S3
- Mengelola izin untuk laporan Inventaris S3, S3 analytics, dan Inventaris S3
- Membutuhkan MFA
- Mencegah pengguna menghapus objek

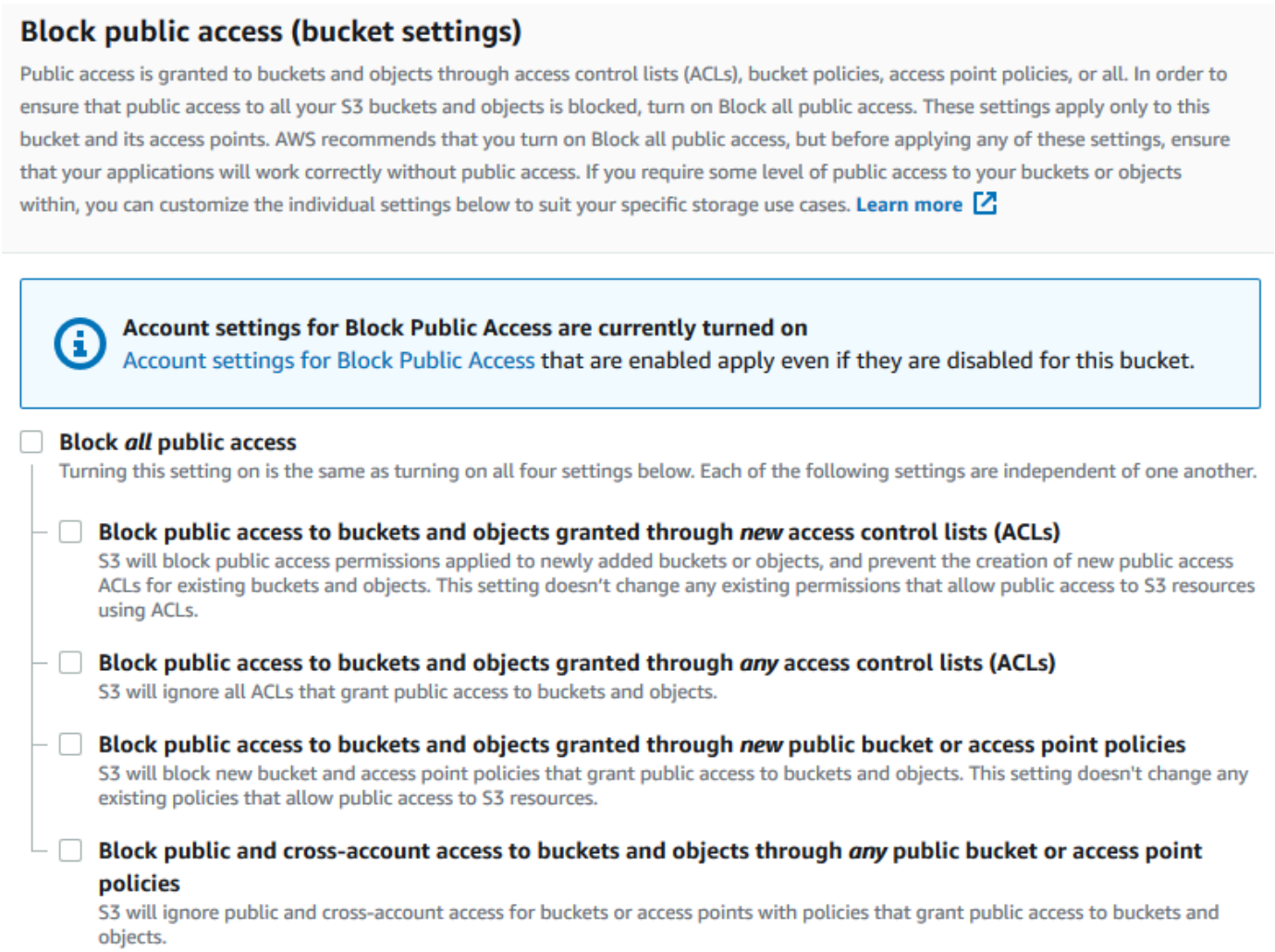
Memberikan izin baca-saja kepada pengguna anonim publik

Anda dapat menggunakan setelan kebijakan untuk memberikan akses ke pengguna anonim publik, yang berguna jika Anda mengonfigurasi bucket sebagai situs web statis. Memberikan akses ke pengguna anonim publik mengharuskan Anda menonaktifkan pengaturan Blokir Akses Publik untuk bucket Anda. Untuk informasi selengkapnya tentang cara melakukannya, dan kebijakan yang diperlukan, lihatMengatur izin untuk akses situs web. Untuk mempelajari cara menyiapkan kebijakan yang lebih ketat untuk tujuan yang sama, lihat Bagaimana cara memberikan akses baca publik ke beberapa objek di bucket Amazon S3 saya? di pusat AWS pengetahuan.

Secara default, Amazon S3 memblokir akses publik ke akun dan bucket Anda. Jika Anda ingin menggunakan bucket untuk menghosting situs web statis, Anda dapat menggunakan langkah-langkah ini untuk mengedit pengaturan blokir akses publik Anda.

> **Warning**
>
> Sebelum Anda menyelesaikan langkah-langkah ini, tinjau Melakukan blok akses publik ke penyimpanan Amazon S3 Anda untuk memastikan bahwa Anda memahami dan menerima risiko yang terkait dengan mengizinkan akses publik. Saat Anda mematikan pengaturan blokir akses publik untuk membuat bucket Anda menjadi publik, siapa pun di internet dapat mengakses bucket Anda. Kami sarankan agar Anda memblokir semua akses publik ke bucket Anda.

1. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Pilih nama bucket yang telah Anda konfigurasikan sebagai situs web statis.
3. Pilih Izin.
4. Di bagian bawah Blokir akses publik (pengaturan bucket), pilih Edit.
5. Kosongkan Blokir semua akses publik, lalu pilih Simpan perubahan.



Amazon S3 mematikan pengaturan Blokir Akses Publik untuk bucket Anda. Untuk membuat situs web statis publik, Anda mungkin juga harus mengedit pengaturan Blokir Akses Publik untuk akun Anda sebelum menambahkan kebijakan bucket. Jika pengaturan Blokir Akses Publik untuk akun Anda saat ini diaktifkan, Anda akan melihat catatan di bawah Blokir akses publik (pengaturan bucket).

Membutuhkan enkripsi

Anda dapat memerlukan enkripsi sisi server dengan AWS Key Management Service (AWS KMS) kunci (SSE-KMS), seperti yang ditunjukkan pada contoh berikut.

Membutuhkan SSE-KMS untuk semua objek yang ditulis ke bucket

Contoh kebijakan berikut mengharuskan setiap objek yang ditulis ke bucket dienkripsi dengan enkripsi sisi server menggunakan AWS Key Management Service (AWS KMS) keys (SSE-KMS). Jika objek tidak dienkripsi dengan SSE-KMS, permintaan ditolak.

```
{
"Version": "2012-10-17",
"Id": "PutObjPolicy",
"Statement": [{
  "Sid": "DenyObjectsThatAreNotSSEKMS",
  "Principal": "*",
  "Effect": "Deny",
  "Action": "s3:PutObject",
  "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
  "Condition": {
    "Null": {
      "s3:x-amz-server-side-encryption-aws-kms-key-id": "true"
    }
  }
}]
}
```

Memerlukan SSE-KMS dengan spesifik AWS KMS key untuk semua objek yang ditulis ke bucket

Contoh kebijakan berikut menyangkal objek apa pun ditulis ke bucket jika tidak dienkripsi dengan SSE-KMS dengan menggunakan ID kunci KMS tertentu. Bahkan jika objek dienkripsi dengan SSE-KMS dengan menggunakan header per permintaan atau enkripsi default bucket, objek tidak dapat ditulis ke bucket jika belum dienkripsi dengan kunci KMS yang ditentukan. Pastikan untuk mengganti ARN kunci KMS yang digunakan dalam contoh ini dengan ARN kunci KMS Anda sendiri.

```
{
"Version": "2012-10-17",
"Id": "PutObjPolicy",
"Statement": [{
  "Sid": "DenyObjectsThatAreNotSSEKMSWithSpecificKey",
  "Principal": "*",
  "Effect": "Deny",
  "Action": "s3:PutObject",
  "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
  "Condition": {
    "ArnNotEqualsIfExists": {
```

```
      "s3:x-amz-server-side-encryption-aws-kms-key-id": "arn:aws:kms:us-
east-2:111122223333:key/01234567-89ab-cdef-0123-456789abcdef"
    }
  }
}]
}
```

Mengelola ember menggunakan kaleng ACLs

Memberikan izin ke beberapa akun untuk mengunggah objek atau mengatur objek ACLs untuk akses publik

Contoh kebijakan berikut memberikan `s3:PutObjectAcl` izin `s3:PutObject` dan ke beberapa. Akun AWS Selain itu, kebijakan contoh mengharuskan setiap permintaan untuk operasi ini harus menyertakan daftar kontrol akses `public-read` kaleng (ACL). Untuk informasi selengkapnya, lihat Tindakan kebijakan untuk Amazon S3 dan Kunci kondisi kebijakan untuk Amazon S3.

> **Warning**
>
> ACL `public-read` terekam memungkinkan siapa pun di dunia untuk melihat objek di bucket Anda. Berhati-hatilah saat memberikan akses anonim ke bucket Amazon S3 Anda atau menonaktifkan pengaturan blokir akses publik. Saat Anda memberikan akses anonim, siapa pun di dunia dapat mengakses bucket Anda. Kami menyarankan Anda untuk tidak pernah memberikan akses anonim ke bucket Amazon S3 Anda kecuali jika Anda secara kustom memerlukannya, seperti hosting situs web statis. Jika Anda ingin mengaktifkan blokir pengaturan akses publik untuk hosting situs web statis, lihat Tutorial: Mengonfigurasi situs web statis di Amazon S3.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AddPublicReadCannedAcl",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:root",
                    "arn:aws:iam::444455556666:root"
                ]
            },
```

```
            "Action": [
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": [
                        "public-read"
                    ]
                }
            }
        }
    ]
}
```

Berikan izin lintas akun untuk unggah objek sekaligus memastikan bahwa pemilik bucket memiliki kendali penuh

Contoh berikut menunjukkan cara mengizinkan orang lain Akun AWS mengunggah objek ke bucket sambil memastikan bahwa Anda memiliki kendali penuh atas objek yang diunggah. Kebijakan ini memberikan kemampuan spesifik Akun AWS (*111122223333*) untuk mengunggah objek hanya jika akun tersebut menyertakan ACL yang `bucket-owner-full-control` dikalengkan saat diunggah. `StringEquals`Kondisi dalam kebijakan tersebut menyebutkan kunci `s3:x-amz-acl` kondisi untuk mengekspresikan persyaratan ACL terekam. Untuk informasi selengkapnya, lihat Kunci kondisi kebijakan untuk Amazon S3.

```
{
   "Version":"2012-10-17",
   "Statement":[
     {
       "Sid":"PolicyForAllowUploadWithACL",
       "Effect":"Allow",
       "Principal":{"AWS":"111122223333"},
       "Action":"s3:PutObject",
       "Resource":"arn:aws:s3:::amzn-s3-demo-bucket/*",
       "Condition": {
         "StringEquals": {"s3:x-amz-acl":"bucket-owner-full-control"}
       }
     }
   ]
}
```

Mengelola akses objek dengan penandaan objek

Mengizinkan pengguna untuk membaca objek yang memiliki kunci dan nilai tag spesifik

Kebijakan izin berikut membatasi pengguna agar hanya membaca objek yang memiliki kunci dan nilai tag `environment: production`. Kebijakan ini menggunakan kunci kondisi `s3:ExistingObjectTag` untuk menentukan kunci tag dan nilai.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Principal":{
            "AWS":"arn:aws:iam::111122223333:role/JohnDoe"
         },
         "Effect":"Allow",
         "Action":[
            "s3:GetObject",
            "s3:GetObjectVersion"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket/*",
         "Condition":{
            "StringEquals":{
               "s3:ExistingObjectTag/environment":"production"
            }
         }
      }
   ]
}
```

Batasi kunci tag objek mana yang dapat ditambahkan pengguna

Kebijakan contoh berikut memberikan izin pengguna untuk melakukan `s3:PutObjectTagging` tindakan, yang memungkinkan pengguna menambahkan tag ke objek yang sudah ada. Kondisi ini menggunakan kunci kondisi `s3:RequestObjectTagKeys` untuk menentukan kunci tag yang diizinkan, seperti `Owner` atau `CreationDate`. Untuk informasi selengkapnya, lihat Membuat kondisi yang menguji beberapa nilai kunci dalam Panduan Pengguna IAM.

Kebijakan ini memastikan agar setiap kunci tag yang ditentukan dalam permintaan adalah kunci tag yang diotorisasi. Kualifikasi `ForAnyValue` dalam kondisi tersebut memastikan bahwa setidaknya ada satu kunci yang ditentukan dalam permintaan.

```
{
   "Version": "2012-10-17",
  "Statement": [
    {"Principal":{"AWS":[
            "arn:aws:iam::111122223333:role/JohnDoe"
         ]
       },
 "Effect": "Allow",
      "Action": [
        "s3:PutObjectTagging"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ],
      "Condition": {"ForAnyValue:StringEquals": {"s3:RequestObjectTagKeys": [
            "Owner",
            "CreationDate"
          ]
        }
      }
    }
  ]
}
```

Memerlukan kunci dan nilai tag spesifik saat mengizinkan pengguna untuk menambahkan tag objek

Kebijakan contoh berikut memberikan izin pengguna untuk melakukan `s3:PutObjectTagging` tindakan, yang memungkinkan pengguna menambahkan tag ke objek yang sudah ada. Kondisi ini mengharuskan pengguna untuk menyertakan kunci tag tertentu (seperti *`Project`*) dengan set nilai ke *X*.

```
{
   "Version": "2012-10-17",
  "Statement": [
    {"Principal":{"AWS":[
       "arn:aws:iam::111122223333:user/JohnDoe"
         ]
       },
      "Effect": "Allow",
      "Action": [
        "s3:PutObjectTagging"
      ],
```

```
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ],
      "Condition": {"StringEquals": {"s3:RequestObjectTag/Project": "X"
        }
      }
    }
  ]
}
```

Mengizinkan pengguna untuk hanya menambahkan objek dengan kunci dan nilai tag objek tertentu

Kebijakan contoh berikut memberikan izin pengguna untuk melakukan `s3:PutObject` tindakan, sehingga pengguna dapat menambahkan tag ke objek yang sudah ada. Namun, `Condition` pernyataan membatasi kunci tag dan nilai yang diizinkan pada objek yang diunggah. Dalam contoh ini, pengguna hanya dapat menambahkan objek yang memiliki kunci tag tertentu (*`Department`*) dengan nilai yang disetel *`Finance`* ke bucket.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Principal":{
            "AWS":[
                 "arn:aws:iam::111122223333:user/JohnDoe"
         ]
        },
        "Effect": "Allow",
        "Action": [
            "s3:PutObject"
        ],
        "Resource": [
            "arn:aws:s3:::amzn-s3-demo-bucket/*"
        ],
        "Condition": {
            "StringEquals": {
                "s3:RequestObjectTag/Department": "Finance"
            }
        }
    }]
}
```

Mengelola akses objek dengan menggunakan kunci kondisi global

Kunci kondisi global adalah kunci konteks kondisi dengan `aws` awalan. Layanan AWS dapat mendukung kunci kondisi global atau kunci khusus layanan yang menyertakan awalan layanan. Anda dapat menggunakan `Condition` elemen kebijakan JSON untuk membandingkan kunci dalam permintaan dengan nilai kunci yang Anda tentukan dalam kebijakan Anda.

Batasi akses hanya ke pengiriman log akses server Amazon S3

Dalam contoh kebijakan bucket berikut, kunci kondisi aws:SourceArnglobal digunakan untuk membandingkan Amazon Resource Name (ARN) sumber daya, membuat service-to-service permintaan dengan ARN yang ditentukan dalam kebijakan. Kunci kondisi `aws:SourceArn` global digunakan untuk mencegah layanan Amazon S3 digunakan sebagai wakil yang bingung selama transaksi antar layanan. Hanya layanan Amazon S3 yang diizinkan untuk menambahkan objek ke bucket Amazon S3.

Contoh kebijakan bucket ini hanya memberikan `s3:PutObject` izin kepada pengguna utama layanan logging (`logging.s3.amazonaws.com`).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutObjectS3ServerAccessLogsPolicy",
            "Principal": {
                "Service": "logging.s3.amazonaws.com"
            },
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket-logs/*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111111111111"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:::EXAMPLE-SOURCE-BUCKET"
                }
            }
        },
        {
            "Sid": "RestrictToS3ServerAccessLogs",
            "Effect": "Deny",
```

```
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket-logs/*",
            "Condition": {
                "ForAllValues:StringNotEquals": {
                    "aws:PrincipalServiceNamesList": "logging.s3.amazonaws.com"
                }
            }
        }
    ]
}
```

Izinkan akses hanya ke organisasi Anda

Jika Anda ingin meminta semua kepala sekolah IAM yang mengakses sumber daya berasal dari organisasi Anda (termasuk akun AWS Organizations manajemen), Anda dapat menggunakan kunci kondisi global. Akun AWS `aws:PrincipalOrgID`

Untuk memberikan atau membatasi jenis akses ini, tentukan `aws:PrincipalOrgID` kondisi dan tetapkan nilainya ke ID organisasi Anda dalam kebijakan bucket. ID organisasi digunakan untuk mengontrol akses ke bucket. Saat Anda menggunakan `aws:PrincipalOrgID` kondisi ini, izin dari kebijakan bucket juga diterapkan ke semua akun baru yang ditambahkan ke organisasi.

Berikut adalah contoh kebijakan bucket berbasis sumber daya yang dapat Anda gunakan untuk memberikan pengguna utama IAM tertentu di organisasi Anda akses langsung ke bucket Anda. Dengan menambahkan kunci kondisi `aws:PrincipalOrgID` global ke kebijakan bucket Anda, pengguna utama sekarang harus berada di organisasi Anda untuk mendapatkan akses ke sumber daya. Bahkan jika Anda secara tidak sengaja menentukan akun yang salah saat memberikan akses, kunci kondisi aws:PrincipalOrgID global bertindak sebagai perlindungan tambahan. Ketika kunci global ini digunakan dalam kebijakan, ini mencegah semua pengguna utama dari luar organisasi tertentu mengakses bucket S3. Hanya pengguna utama dari akun di organisasi yang terdaftar yang dapat memperoleh akses ke sumber daya.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "AllowGetObject",
        "Principal": {
            "AWS": "*"
        },
        "Effect": "Allow",
```

```
        "Action": "s3:GetObject",
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
        "Condition": {
            "StringEquals": {
                "aws:PrincipalOrgID": ["o-aa111bb222"]
            }
        }
    }]
}
```

Mengelola akses sesuai permintaan HTTP atau HTTPS

Batasi Akses Hanya ke Permintaan HTTPS

Jika Anda ingin mencegah penyerang potensial memanipulasi lalu lintas jaringan, Anda dapat menggunakan HTTPS (TLS) untuk hanya mengizinkan koneksi terenkripsi sambil membatasi permintaan HTTP mengakses bucket Anda. Untuk menentukan apakah permintaan tersebut HTTP atau HTTPS, gunakan kunci kondisi aws:SecureTransportglobal dalam kebijakan bucket S3 Anda. Kunci `aws:SecureTransport` kondisi memeriksa apakah permintaan dikirim dengan menggunakan HTTP.

Jika permintaan kembali`true`, maka permintaan dikirim melalui HTTPS. Jika permintaan kembali`false`, maka permintaan dikirim melalui HTTP. Anda kemudian dapat mengizinkan atau menolak akses ke bucket Anda berdasarkan skema permintaan yang diinginkan.

Dalam contoh berikut, kebijakan bucket secara eksplisit menolak permintaan HTTP.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "RestrictToTLSRequestsOnly",
        "Action": "s3:*",
        "Effect": "Deny",
        "Resource": [
            "arn:aws:s3:::amzn-s3-demo-bucket",
            "arn:aws:s3:::amzn-s3-demo-bucket/*"
        ],
        "Condition": {
            "Bool": {
                "aws:SecureTransport": "false"
            }
        },
        "Principal": "*"
```

```
    }]
}
```

Membatasi akses ke suatu Wilayah tertentu

Misalkan Anda memiliki situs web dengan nama domain *www.example.com* atau *example.com* dengan tautan ke foto dan video yang disimpan di bucket Anda bernama *amzn-s3-demo-bucket*. Secara default, semua sumber daya Amazon S3 bersifat pribadi, jadi hanya Akun AWS yang membuat sumber daya yang dapat mengaksesnya.

Untuk mengizinkan akses baca ke objek ini dari situs web Anda, Anda dapat menambahkan kebijakan bucket yang mengizinkan `s3:GetObject` izin dengan syarat `GET` permintaan harus berasal dari halaman web tertentu. Kebijakan berikut membatasi permintaan dengan menggunakan `StringLike` kondisi dengan kunci `aws:Referer` kondisi.

```
{
  "Version":"2012-10-17",
  "Id":"HTTP referer policy example",
  "Statement":[
    {
      "Sid":"Allow only GET requests originating from www.example.com and
 example.com.",
      "Effect":"Allow",
      "Principal":"*",
      "Action":["s3:GetObject","s3:GetObjectVersion"],
      "Resource":"arn:aws:s3:::amzn-s3-demo-bucket/*",
      "Condition":{
        "StringLike":{"aws:Referer":["http://www.example.com/*","http://example.com/
*"]}
      }
    }
  ]
}
```

Pastikan browser yang Anda gunakan menyertakan `referer` header HTTP dalam permintaan.

> **Warning**
>
> Sebaiknya Anda berhati-hati saat menggunakan tombol `aws:Referer` kondisi. Menyertakan nilai header perujuk HTTP yang diketahui publik sangat berbahaya. Pihak yang tidak berwenang dapat menggunakan browser yang diubah atau disesuaikan untuk menyediakan

nilai `aws:Referer` yang mereka pilih. Oleh karena itu, jangan gunakan `aws:Referer` untuk mencegah pihak yang tidak berwenang membuat AWS permintaan langsung.
Kunci `aws:Referer` kondisi ditawarkan hanya untuk memungkinkan pelanggan melindungi konten digital mereka, seperti konten yang disimpan di Amazon S3, agar tidak dirujuk di situs pihak ketiga yang tidak sah. Untuk informasi lebih lanjut, lihat aws:Referer dalam Panduan Pengguna IAM.

Mengelola akses pengguna ke folder tertentu

memberikan pengguna akses ke folder tertentu

Misalkan Anda mencoba memberikan pengguna akses ke folder tertentu. Jika pengguna IAM dan bucket S3 milik yang sama Akun AWS, maka Anda dapat menggunakan kebijakan IAM untuk memberikan akses pengguna ke folder bucket tertentu. Dengan pendekatan ini, Anda tidak perlu memperbarui kebijakan bucket untuk memberikan akses. Anda dapat menambahkan kebijakan IAM ke peran IAM yang dapat dialihkan oleh beberapa pengguna.

Jika identitas IAM dan bucket S3 milik yang berbeda Akun AWS, Anda harus memberikan akses lintas akun baik dalam kebijakan IAM maupun kebijakan bucket. Untuk informasi selengkapnya tentang pemberian akses lintas akun, lihat Pemilik bucket yang memberikan izin bucket lintas akun.

Contoh kebijakan bucket berikut memberikan akses konsol *JohnDoe* penuh hanya ke foldernya (home/*JohnDoe*/). Dengan membuat home folder dan memberikan izin yang sesuai kepada pengguna, Anda dapat meminta beberapa pengguna berbagi satu bucket. Kebijakan ini terdiri dari tiga `Allow` pernyataan:

- *AllowRootAndHomeListingOfCompanyBucket*: Memungkinkan user (*JohnDoe*) untuk daftar objek di tingkat root *amzn-s3-demo-bucket* bucket dan di home folder. Pernyataan ini juga memungkinkan pengguna untuk mencari pada prefiks home/ dengan menggunakan konsol.
- *AllowListingOfUserFolder*: Memungkinkan user (*JohnDoe*) untuk daftar semua objek dalam home/*JohnDoe*/ folder dan subfolder apapun.
- *AllowAllS3ActionsInUserFolder*: Memungkinkan pengguna untuk melakukan semua tindakan Amazon S3 dengan memberikan `Read`, `Write`, dan izin `Delete`. Izin terbatas pada folder beranda pemilik bucket.

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Sid": "AllowRootAndHomeListingOfCompanyBucket",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:user/JohnDoe"
                ]
            },
            "Effect": "Allow",
            "Action": ["s3:ListBucket"],
            "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket"],
            "Condition": {
                "StringEquals": {
                    "s3:prefix": ["", "home/", "home/JohnDoe"],
                    "s3:delimiter": ["/"]
                }
            }
        },
        {
            "Sid": "AllowListingOfUserFolder",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:user/JohnDoe"
                ]
            },
            "Action": ["s3:ListBucket"],
            "Effect": "Allow",
            "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket"],
            "Condition": {
                "StringLike": {
                    "s3:prefix": ["home/JohnDoe/*"]
                }
            }
        },
        {
            "Sid": "AllowAllS3ActionsInUserFolder",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:user/JohnDoe"
                ]
            },
            "Action": ["s3:*"],
            "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket/home/JohnDoe/*"]
```

```
        }
    ]
}
```

Mengelola akses log akses

Berikan akses ke Penyeimbang Beban Aplikasi untuk mengaktifkan log akses

Saat Anda mengaktifkan pencatatan akses untuk Penyeimbang Beban Aplikasi, Anda harus menentukan nama bucket S3 tempat penyeimbang beban akan menyimpan log. Bucket harus memiliki kebijakan terlampir yang memberikan izin Elastic Load Balancing untuk menulis ke bucket.

Pada contoh berikut, kebijakan bucket memberikan izin Elastic Load Balancing (ELB) untuk menulis log akses ke bucket:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Principal": {
        "AWS": "arn:aws:iam::elb-account-id:root"
      },
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/AWSLogs/111122223333/*"
    }
  ]
}
```

> **Note**
>
> Pastikan untuk mengganti *elb-account-id* dengan Akun AWS ID untuk Elastic Load Balancing untuk Anda. Wilayah AWS Untuk daftar Wilayah Elastic Load Balancing, lihat Melampirkan kebijakan ke bucket Amazon S3 Anda di Panduan Pengguna Elastic Load Balancing.

Jika Anda Wilayah AWS tidak muncul di daftar Wilayah Elastic Load Balancing yang didukung, gunakan kebijakan berikut, yang memberikan izin ke layanan pengiriman log yang ditentukan.

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
       "Principal": {
         "Service": "logdelivery.elasticloadbalancing.amazonaws.com"
          },
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/AWSLogs/111122223333/*"
    }
  ]
}
```

Kemudian, pastikan untuk mengonfigurasi log akses Elastic Load Balancing Anda dengan mengaktifkannya. Anda dapat memverifikasi izin bucket dengan membuat file pengujian.

Mengelola akses ke Amazon CloudFront OAI

Berikan izin ke Amazon CloudFront OAI

Contoh kebijakan bucket berikut memberikan izin identitas akses CloudFront asal (OAI) untuk mendapatkan (membaca) semua objek di bucket S3 Anda. Anda dapat menggunakan CloudFront OAI untuk memungkinkan pengguna mengakses objek di bucket Anda CloudFront tetapi tidak secara langsung melalui Amazon S3. Untuk informasi selengkapnya, lihat Membatasi akses ke konten Amazon S3 menggunakan Identitas Akses Asal di Panduan Pengembang CloudFront Amazon.

Kebijakan berikut menggunakan ID OAI sebagai `Principal`. Untuk informasi selengkapnya tentang menggunakan kebijakan bucket S3 guna memberikan akses ke CloudFront OAI, lihat Memigrasi dari identitas akses asal (OAI) ke kontrol akses asal (OAC) di Panduan Pengembang Amazon. CloudFront

Untuk menggunakan contoh ini:

- Ganti *EH1HDMB1FH2TC* dengan ID OAI. Untuk menemukan ID OAI, lihat halaman Origin Access Identity di CloudFront konsol, atau gunakan ListCloudFrontOriginAccessIdentitiesdi CloudFront API.
- Ganti *amzn-s3-demo-bucket* dengan nama bucket Anda.

```
{
    "Version": "2012-10-17",
```

```
    "Id": "PolicyForCloudFrontPrivateContent",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access
 Identity EH1HDMB1FH2TC"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
        }
    ]
}
```

Mengelola akses untuk Lensa Penyimpanan Amazon S3

Berikan Izin untuk Lensa Penyimpanan Amazon S3

Lensa Penyimpanan S3 menggabungkan metrik Anda dan menampilkan informasi di bagian Snapshot akun di halaman Bucket konsol Amazon S3. Lensa Penyimpanan S3 juga menyediakan dasbor interaktif yang dapat Anda gunakan untuk memvisualisasikan wawasan dan tren, menandai outlier, serta menerima rekomendasi untuk mengoptimalkan biaya penyimpanan dan menerapkan praktik terbaik perlindungan data. Dasbor Anda memiliki opsi perincian untuk menghasilkan dan memvisualisasikan wawasan di tingkat organisasi, akun, Wilayah AWS, kelas penyimpanan, bucket, prefiks, atau grup Lensa Penyimpanan. Anda juga dapat mengirimkan ekspor metrik harian dalam format CSV atau Parquet ke bucket S3.

Lensa Penyimpanan S3 dapat mengekspor metrik penggunaan penyimpanan gabungan Anda ke bucket Amazon S3 untuk analisis lebih lanjut. Bucket tempat Lensa Penyimpanan S3 menempatkan ekspor metrik nya dikenal sebagai bucket tujuan. Saat mengatur ekspor metrik Lensa Penyimpanan S3, Anda harus memiliki kebijakan bucket untuk bucket tujuan. Untuk informasi selengkapnya, lihat Menilai aktivitas penyimpanan dan penggunaan dengan Lensa Penyimpanan Amazon S3.

Contoh kebijakan bucket berikut memberikan kepada Amazon S3 izin untuk tulis objek PUT (permintaan) ke bucket tujuan. Anda menggunakan kebijakan bucket seperti ini pada bucket tujuan saat membuat pengaturan ekspor metrik Lensa Penyimpanan S3.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Sid": "S3StorageLensExamplePolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "storage-lens.s3.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-destination-bucket/destination-prefix/
StorageLens/111122223333/*"
            ],
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceAccount": "111122223333",
                    "aws:SourceArn": "arn:aws:s3:region-code:111122223333:storage-
lens/storage-lens-dashboard-configuration-id"
                }
            }
        }
    ]
}
```

Saat Anda menyiapkan ekspor metrik tingkat organisasi Lensa Penyimpanan S3, gunakan modifikasi berikut pada pernyataan `Resource` kebijakan bucket sebelumnya.

```
"Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket/destination-prefix/
StorageLens/your-organization-id/*",
```

Mengelola izin untuk laporan Inventaris S3, S3 analytics, dan Inventaris S3

Berikan izin untuk Inventaris S3 dan analitik S3

S3 Inventaris membuat daftar objek dalam bucket, dan ekspor Analisis Kelas Penyimpanan analitik S3 membuat file output dari data yang digunakan dalam analisis. Bucket tempat inventaris mencantumkan objek disebut bucket sumber. Bucket tempat file inventaris atau file ekspor analitik ditulis disebut bucket tujuan. Saat membuat inventaris atau ekspor analitik, Anda harus membuat kebijakan bucket untuk bucket tujuan. Untuk informasi selengkapnya, lihat Membuat katalog dan menganalisis data Anda dengan S3 Inventory dan Analitik Amazon S3–Analisis Kelas Penyimpanan.

Contoh kebijakan bucket berikut memberikan kepada Amazon S3 izin untuk menulis objek PUT (permintaan) dari akun untuk bucket sumber ke bucket tujuan. Anda menggunakan kebijakan bucket seperti ini pada bucket tujuan saat menyiapkan S3 Inventory, dan ekspor analitik S3.

```
{
      "Version": "2012-10-17",
      "Statement": [
        {
            "Sid": "InventoryAndAnalyticsExamplePolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-DESTINATION-BUCKET/*"
            ],
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET"
                },
                "StringEquals": {
                    "aws:SourceAccount": "111122223333",
                    "s3:x-amz-acl": "bucket-owner-full-control"
                }
            }
        }
    ]
}
```

Kontrol pembuatan konfigurasi laporan Inventaris S3

Membuat katalog dan menganalisis data Anda dengan S3 Inventorymembuat daftar objek dalam bucket S3 dan metadata untuk setiap objek. `s3:PutInventoryConfiguration`Izin ini memungkinkan pengguna untuk membuat konfigurasi inventaris yang mencakup semua bidang metadata objek yang tersedia secara default dan menentukan bucket tujuan untuk menyimpan inventaris. Pengguna dengan akses baca ke objek di bucket tujuan dapat mengakses semua bidang metadata objek yang tersedia dalam laporan inventaris. Untuk informasi selengkapnya tentang bidang metadata yang tersedia di Inventaris S3, lihat. Daftar Inventaris Amazon S3

Untuk membatasi pengguna mengonfigurasi laporan Inventaris S3, hapus `s3:PutInventoryConfiguration` izin dari pengguna.

Beberapa bidang metadata objek dalam konfigurasi laporan Inventaris S3 bersifat opsional, artinya mereka tersedia secara default tetapi dapat dibatasi saat Anda memberikan izin kepada pengguna. `s3:PutInventoryConfiguration` Anda dapat mengontrol apakah pengguna dapat menyertakan bidang metadata opsional ini dalam laporan mereka dengan menggunakan kunci `s3:InventoryAccessibleOptionalFields` kondisi. Untuk daftar bidang metadata opsional yang tersedia di S3 Inventory, lihat OptionalFieldsdi Referensi API Amazon Simple Storage Service.

Untuk memberikan izin kepada pengguna untuk membuat konfigurasi inventaris dengan bidang metadata opsional tertentu, gunakan kunci `s3:InventoryAccessibleOptionalFields` kondisi untuk menyempurnakan kondisi dalam kebijakan bucket Anda.

Contoh kebijakan berikut memberikan izin kepada pengguna (*Ana*) untuk membuat konfigurasi inventaris secara kondisional. `ForAllValues:StringEquals`Kondisi dalam kebijakan menggunakan kunci `s3:InventoryAccessibleOptionalFields` kondisi untuk menentukan dua bidang metadata opsional yang diizinkan, yaitu `Size` dan. `StorageClass` Jadi, saat *Ana* membuat konfigurasi inventaris, satu-satunya bidang metadata opsional yang dapat dia sertakan adalah `Size` dan. `StorageClass`

```
{
 "Id": "InventoryConfigPolicy",
 "Version": "2012-10-17",
 "Statement": [{
   "Sid": "AllowInventoryCreationConditionally",
   "Effect": "Allow",
   "Principal": {
    "AWS": "arn:aws:iam::111122223333:user/Ana"
   },
   "Action":
    "s3:PutInventoryConfiguration",
   "Resource":
    "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET",
   "Condition": {
    "ForAllValues:StringEquals": {
     "s3:InventoryAccessibleOptionalFields": [
        "Size",
        "StorageClass"
        ]
      }
    }
   }
 ]
```

```
}
```

Untuk membatasi pengguna mengonfigurasi laporan Inventaris S3 yang menyertakan bidang metadata opsional tertentu, tambahkan `Deny` pernyataan eksplisit ke kebijakan bucket untuk bucket sumber. Contoh kebijakan bucket berikut menyangkal pengguna *Ana* membuat konfigurasi inventaris di bucket sumber *DOC-EXAMPLE-SOURCE-BUCKET* yang menyertakan bidang opsional `ObjectAccessControlList` atau `ObjectOwner` metadata. Pengguna masih *Ana* dapat membuat konfigurasi inventaris dengan bidang metadata opsional lainnya.

```
{
 "Id": "InventoryConfigSomeFields",
 "Version": "2012-10-17",
 "Statement": [{
   "Sid": "AllowInventoryCreation",
   "Effect": "Allow",
   "Principal": {
    "AWS": "arn:aws:iam::111122223333:user/Ana"
   },
   "Action": "s3:PutInventoryConfiguration",
   "Resource":
    "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET",

  },
  {
   "Sid": "DenyCertainInventoryFieldCreation",
   "Effect": "Deny",
   "Principal": {
    "AWS": "arn:aws:iam::111122223333:user/Ana"
   },
   "Action": "s3:PutInventoryConfiguration",
   "Resource":
     "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET",
   "Condition": {
    "ForAnyValue:StringEquals": {
     "s3:InventoryAccessibleOptionalFields": [
        "ObjectOwner",
        "ObjectAccessControlList"
        ]
      }
    }
   }
 ]
```

```
}
```

> **Note**
>
> Penggunaan kunci `s3:InventoryAccessibleOptionalFields` kondisi dalam kebijakan bucket tidak memengaruhi pengiriman laporan inventaris berdasarkan konfigurasi inventaris yang ada.

> **Important**
>
> Kami menyarankan Anda menggunakan `ForAllValues` dengan `Allow` efek atau `ForAnyValue` dengan `Deny` efek, seperti yang ditunjukkan pada contoh sebelumnya. Jangan gunakan `ForAllValues` dengan `Deny` efek atau `ForAnyValue` `Allow` efek, karena kombinasi ini bisa terlalu membatasi dan memblokir penghapusan konfigurasi inventaris. Untuk mempelajari selengkapnya tentang operator set `ForAllValues` dan `ForAnyValue` kondisi, lihat Kunci konteks Multivalued di Panduan Pengguna IAM.

Membutuhkan MFA

Amazon S3 mendukung akses API yang dilindungi MFA, sebuah fitur yang dapat memberlakukan autentikasi multi-faktor (MFA) untuk mengakses sumber daya Amazon S3. Otentikasi multi-faktor memberikan tingkat keamanan ekstra yang dapat Anda terapkan ke lingkungan Anda AWS . MFA adalah fitur keamanan yang mengharuskan pengguna membuktikan kepemilikan fisik perangkat MFA dengan menyediakan kode MFA yang valid. Untuk informasi selengkapnya, lihat Autentikasi Multi-Faktor AWS. Anda dapat mewajibkan MFA untuk setiap permintaan akses sumber daya Amazon S3.

Untuk memberlakukan persyaratan MFA, gunakan kunci kondisi `aws:MultiFactorAuthAge` dalam kebijakan bucket. Pengguna IAM dapat mengakses sumber daya Amazon S3 dengan menggunakan kredenal sementara yang dikeluarkan oleh (). AWS Security Token Service AWS STS Anda memberikan kode MFA pada saat permintaan AWS STS .

Saat Amazon S3 menerima permintaan dengan autentikasi multi-faktor, kunci `aws:MultiFactorAuthAge` kondisi memberikan nilai numerik yang menunjukkan berapa lama (dalam detik) kredensial sementara dibuat. Jika kredensial sementara yang diberikan dalam permintaan tidak dibuat menggunakan perangkat MFA, maka nilai kunci ini akan kosong (tidak ada).

Dalam kebijakan bucket, Anda dapat menambahkan sebuah kondisi untuk memeriksa nilai ini, seperti yang ditunjukkan dalam contoh berikut.

Contoh kebijakan ini menolak operasi Amazon S3 pada */taxdocuments*folder dalam bucket jika permintaan tidak *amzn-s3-demo-bucket* diautentikasi dengan menggunakan MFA. Untuk pelajari selengkapnya tentang otentikasi multifaktor (MFA), lihat Menggunakan Autentikasi Multi-Faktor (MFA) di AWS dalam Panduan Pengguna IAM.

```
{
    "Version": "2012-10-17",
    "Id": "123",
    "Statement": [
      {
        "Sid": "",
        "Effect": "Deny",
        "Principal": "*",
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/taxdocuments/*",
        "Condition": { "Null": { "aws:MultiFactorAuthAge": true }}
      }
    ]
 }
```

`Null`Kondisi di `Condition` blok mengevaluasi `true` apakah nilai kunci `aws:MultiFactorAuthAge` kondisi adalah nol, yang mana hal ini menunjukkan bahwa kredensial keamanan sementara dalam permintaan dibuat tanpa perangkat MFA.

Kebijakan bucket berikut merupakan perpanjangan dari kebijakan bucket sebelumnya. Kebijakan berikut mencakup dua pernyataan kebijakan. Satu pernyataan memberikan izin `s3:GetObject` pada bucket (*amzn-s3-demo-bucket*) kepada semua orang. Pernyataan lain membatasi lebih lanjut akses ke folder *amzn-s3-demo-bucket/taxdocuments* yang ada dalam bucket dengan mewajibkan MFA.

```
{
    "Version": "2012-10-17",
    "Id": "123",
    "Statement": [
      {
        "Sid": "",
        "Effect": "Deny",
        "Principal": "*",
```

```
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/taxdocuments/*",
        "Condition": { "Null": { "aws:MultiFactorAuthAge": true } }
      },
      {
        "Sid": "",
        "Effect": "Allow",
        "Principal": "*",
        "Action": ["s3:GetObject"],
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
      }
    ]
 }
```

Anda dapat menggunakan kondisi numerik secara opsional untuk membatasi durasi berlakunya `aws:MultiFactorAuthAge` kunci. Durasi yang Anda tentukan dengan `aws:MultiFactorAuthAge` kunci tidak bergantung pada usia pakai kredensial keamanan sementara yang digunakan untuk mengautentikasi permintaan.

Sebagai contoh, kebijakan bucket berikut, selain mewajibkan autentikasi MFA, juga memeriksa berapa lama lagi sesi sementara dibuat. Kebijakan tersebut menolak setiap operasi jika nilai kunci `aws:MultiFactorAuthAge` menunjukkan bahwa sesi sementara dibuat lebih dari satu jam yang lalu (3.600 detik).

```
{
    "Version": "2012-10-17",
    "Id": "123",
    "Statement": [
      {
        "Sid": "",
        "Effect": "Deny",
        "Principal": "*",
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/taxdocuments/*",
        "Condition": {"Null": {"aws:MultiFactorAuthAge": true }}
      },
      {
        "Sid": "",
        "Effect": "Deny",
        "Principal": "*",
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/taxdocuments/*",
```

```
        "Condition": {"NumericGreaterThan": {"aws:MultiFactorAuthAge": 3600 }}
      },
      {
        "Sid": "",
        "Effect": "Allow",
        "Principal": "*",
        "Action": ["s3:GetObject"],
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
      }
    ]
}
```

Mencegah pengguna menghapus objek

Secara bawaan, pengguna tidak memiliki izin. Namun saat membuat kebijakan, Anda mungkin memberikan izin kepada pengguna yang tidak ingin Anda berikan. Untuk menghindari celah izin tersebut, Anda dapat menulis kebijakan akses yang lebih ketat dengan menambahkan penolakan eksplisit.

Untuk secara eksplisit memblokir pengguna atau akun agar tidak menghapus objek, Anda harus menambahkan tindakan berikut ke kebijakan bucket:`s3:DeleteObject`,`s3:DeleteObjectVersion`, dan izin. `s3:PutLifecycleConfiguration` Ketiga tindakan tersebut diperlukan karena Anda dapat menghapus objek baik dengan memanggil operasi `DELETE Object` API secara eksplisit atau dengan mengonfigurasi siklus hidupnya (lihat) Mengelola siklus hidup objek sehingga Amazon S3 dapat menghapus objek saat masa pakainya berakhir.

Dalam contoh kebijakan berikut, Anda secara eksplisit menolak `DELETE Object` izin kepada pengguna. *MaryMajor* DenyPernyataan eksplisit selalu menggantikan izin lain yang diberikan.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "statement1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/MaryMajor"
      },
      "Action": [
        "s3:GetObjectVersion",
```

```
        "s3:GetBucketAcl"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket1",
   "arn:aws:s3:::amzn-s3-demo-bucket1/*"
      ]
    },
    {
      "Sid": "statement2",
      "Effect": "Deny",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/MaryMajor"
      },
      "Action": [
        "s3:DeleteObject",
        "s3:DeleteObjectVersion",
        "s3:PutLifecycleConfiguration"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket1",
     "arn:aws:s3:::amzn-s3-demo-bucket1/*"
      ]
    }
  ]
}
```

Contoh kebijakan bucket menggunakan tombol kondisi

Anda dapat menggunakan bahasa kebijakan akses untuk menentukan syarat-syarat saat memberikan izin. Anda dapat menggunakan `Condition` elemen opsional, atau `Condition` memblokir, untuk menentukan kondisi kapan kebijakan diberlakukan.

Untuk kebijakan yang menggunakan kunci syarat Amazon S3 untuk operasi objek dan bucket, lihat contoh berikut. Untuk informasi lebih lanjut tentang kunci syarat ini, lihat Kunci kondisi kebijakan untuk Amazon S3. Untuk daftar lengkap tindakan Amazon S3, kunci kondisi, dan sumber daya yang dapat Anda tentukan dalam kebijakan, lihat Tindakan, sumber daya, dan kunci kondisi untuk Amazon S3 di Referensi Otorisasi Layanan.

Untuk informasi selengkapnya tentang izin operasi S3 API menurut jenis sumber daya S3, lihat. Izin yang diperlukan untuk operasi API Amazon S3

Contoh: Kunci kondisi Amazon S3 untuk operasi objek

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan kunci kondisi khusus Amazon S3 untuk operasi objek. Untuk daftar lengkap tindakan Amazon S3, kunci kondisi, dan sumber daya yang dapat Anda tentukan dalam kebijakan, lihat Tindakan, sumber daya, dan kunci kondisi untuk Amazon S3 di Referensi Otorisasi Layanan.

Untuk informasi selengkapnya tentang izin operasi S3 API menurut jenis sumber daya S3, lihat. Izin yang diperlukan untuk operasi API Amazon S3

Beberapa contoh kebijakan menunjukkan bagaimana Anda dapat menggunakan kunci kondisi operasi PUT Objek. Operasi PUT Objek memungkinkan header spesifik daftar kontrol akses (ACL) yang dapat Anda gunakan untuk memberikan izin berbasis ACL. Dengan menggunakan tombol kondisi ini, Anda dapat mengatur kondisi untuk memerlukan izin akses tertentu saat pengguna mengunggah objek. Anda juga dapat memberikan izin berbasis ACL dengan operasi tersebut. PutObjectAcl Untuk informasi lebih lanjut, lihat PutObjectAcl dalam Referensi API Layanan Penyimpanan Sederhana Amazon S3 Amazon. Untuk informasi lebih lanjut tentang ACLs, lihatGambaran umum daftar kontrol akses (ACL).

Topik

- Contoh 1: Memberikan s3:PutObject izin yang mengharuskan objek disimpan menggunakan enkripsi sisi server
- Contoh 2: Memberikan s3:PutObject izin untuk menyalin objek dengan batasan pada sumber salinan
- Contoh 3: Memberikan akses ke versi objek tertentu
- Contoh 4: Memberikan izin berdasarkan tag objek
- Contoh 5: Membatasi akses oleh Akun AWS ID pemilik bucket
- Contoh 6: Membutuhkan versi TLS minimum
- Contoh 7: Mengecualikan prinsipal tertentu dari pernyataan Deny
- Contoh 8: Menegakkan klien untuk mengunggah objek secara kondisional berdasarkan nama kunci objek atau ETags

Contoh 1: Memberikan **`s3:PutObject`** izin yang mengharuskan objek disimpan menggunakan enkripsi sisi server

Misalkan ketika Akun A memiliki bucket. Administrator akun ingin memberi Jane, pengguna di Akun A, izin untuk mengunggah objek dengan syarat Jane selalu meminta enkripsi sisi server

dengan kunci terkelola Amazon S3 (SSE-S3). Administrator Akun A dapat menentukan persyaratan ini dengan menggunakan kunci `s3:x-amz-server-side-encryption` kondisi seperti yang ditunjukkan. Pasangan kunci-nilai di `Condition` blok berikut menentukan kunci `s3:x-amz-server-side-encryption` kondisi dan SSE-S3 () `AES256` sebagai jenis enkripsi:

```
"Condition": {
     "StringNotEquals": {
         "s3:x-amz-server-side-encryption": "AES256"
     }}
```

Saat menguji izin ini dengan menggunakan AWS CLI, Anda harus menambahkan enkripsi yang diperlukan dengan menggunakan `--server-side-encryption` parameter, seperti yang ditunjukkan pada contoh berikut. Untuk menggunakan perintah contoh ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key HappyFace.jpg --body c:
\HappyFace.jpg --server-side-encryption "AES256" --profile AccountAadmin
```

Contoh 2: Memberikan **`s3:PutObject`** izin untuk menyalin objek dengan batasan pada sumber salinan

Dalam permintaan PUT objek, ketika Anda menentukan objek sumber, permintaan adalah operasi salinan (lihat CopyObject). Oleh karena itu, pemilik bucket dapat memberikan izin kepada pengguna untuk menyalin objek dengan pembatasan pada sumber, misalnya:

- Izinkan menyalin objek hanya dari bucket sumber yang ditentukan (misalnya,*`amzn-s3-demo-source-bucket`*).
- Izinkan menyalin objek dari bucket sumber yang ditentukan dan hanya objek yang awalan nama kuncinya dimulai dengan awalan tertentu, seperti *`public/`* (misalnya,). *`amzn-s3-demo-source-bucket/public/*`*
- Izinkan menyalin hanya objek tertentu dari bucket sumber (misalnya,*`amzn-s3-demo-source-bucket/example.jpg`*).

Kebijakan bucket berikut memberikan `s3:PutObject` izin kepada pengguna (*`Dave`*). Kebijakan ini memungkinkan dia untuk menyalin objek hanya dengan syarat permintaan menyertakan `s3:x-amz-copy-source` header dan bahwa nilai header menentukan awalan nama */`amzn-s3-demo-source-bucket/public/*`* kunci. Untuk menggunakan kebijakan contoh ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
{
    "Version": "2012-10-17",
    "Statement": [
       {
            "Sid": "cross-account permission to user in your own account",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/Dave"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
        },
        {
            "Sid": "Deny your user permission to upload object if copy source is not /
bucket/prefix",
            "Effect": "Deny",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/Dave"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-source-bucket/*",
            "Condition": {
                "StringNotLike": {
                    "s3:x-amz-copy-source": "amzn-s3-demo-source-bucket/public/*"
                }
            }
        }
    ]
}
```

Uji kebijakan dengan AWS CLI

Anda dapat menguji izin menggunakan AWS CLI `copy-object` perintah. Anda menentukan sumber dengan menambahkan parameter `--copy-source`; prefiks nama kunci harus cocok dengan prefiks yang diperbolehkan dalam kebijakan. Anda perlu memberikan kredensial Dave kepada pengguna dengan menggunakan parameter `--profile`. Untuk informasi selengkapnya tentang pengaturan AWS CLI, lihat Mengembangkan dengan Amazon S3 menggunakan AWS CLI di Referensi API Amazon S3.

```
aws s3api copy-object --bucket amzn-s3-demo-source-bucket --key HappyFace.jpg
```

```
--copy-source amzn-s3-demo-source-bucket/public/PublicHappyFace1.jpg --
profile AccountADave
```

Berikan izin untuk menyalin objek tertentu

Kebijakan sebelumnya menggunakan kondisi `StringNotLike`. Untuk memberikan izin untuk menyalin hanya objek tertentu, Anda harus mengubah kondisi dari `StringNotLike` ke `StringNotEquals` dan kemudian menentukan kunci objek yang tepat, seperti yang ditunjukkan pada contoh berikut. Untuk menggunakan perintah contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
"Condition": {
       "StringNotEquals": {
           "s3:x-amz-copy-source": "amzn-s3-demo-source-bucket/public/
PublicHappyFace1.jpg"
       }
}
```

Contoh 3: Memberikan akses ke versi objek tertentu

Misalkan Akun A memiliki bucket berkemampuan versi. Bucket tersebut memiliki beberapa versi objek *HappyFace.jpg*. Administrator Akun A sekarang ingin memberikan *Dave* izin kepada pengguna untuk mendapatkan hanya versi objek tertentu. Administrator akun dapat melakukannya dengan memberikan `s3:GetObjectVersion` izin kepada pengguna *Dave* secara kondisional, seperti yang ditunjukkan pada contoh berikut. Pasangan nilai kunci dalam pemblokiran `Condition` menentukan kunci kondisi `s3:VersionId`. Dalam hal ini, untuk mengambil objek dari bucket berkemampuan versi tertentu, *Dave* perlu mengetahui ID versi objek yang tepat. Untuk menggunakan kebijakan contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

Untuk informasi selengkapnya, lihat GetObject dalam Referensi API Amazon Simple Storage Service.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "statement1",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/Dave"
```

```
            },
            "Action": "s3:GetObjectVersion",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/HappyFace.jpg"
        },
        {
            "Sid": "statement2",
            "Effect": "Deny",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/Dave"
            },
            "Action": "s3:GetObjectVersion",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/HappyFace.jpg",
            "Condition": {
                "StringNotEquals": {
                    "s3:VersionId": "AaaHbAQitwiL_h47_44lRO2DDfLlBO5e"
                }
            }
        }
    ]
}
```

Uji kebijakan dengan AWS CLI

Anda dapat menguji izin dalam kebijakan ini dengan menggunakan AWS CLI `get-object` perintah dengan `--version-id` parameter untuk mengidentifikasi versi objek tertentu yang akan diambil. Perintah mengambil versi objek yang ditentukan dan menyimpannya ke *`OutputFile.jpg`* file.

```
aws s3api get-object --bucket amzn-s3-demo-bucket --key HappyFace.jpg OutputFile.jpg --
version-id AaaHbAQitwiL_h47_44lRO2DDfLlBO5e --profile AccountADave
```

Contoh 4: Memberikan izin berdasarkan tag objek

Untuk contoh cara menggunakan kunci kondisi penandaan objek dengan operasi Amazon S3, lihat. Kebijakan pemberian tag dan kontrol akses

Contoh 5: Membatasi akses oleh Akun AWS ID pemilik bucket

Anda dapat menggunakan kunci `aws:ResourceAccount` atau `s3:ResourceAccount` kondisi untuk menulis kebijakan titik akhir IAM atau virtual private cloud (VPC) yang membatasi akses pengguna, peran, atau aplikasi ke bucket Amazon S3 yang dimiliki oleh ID tertentu. Akun AWS Anda dapat menggunakan kunci kondisi ini untuk membatasi klien dalam VPC Anda agar tidak mengakses bucket yang tidak Anda miliki.

Namun, ketahuilah bahwa beberapa AWS layanan bergantung pada akses ke bucket yang AWS dikelola. Oleh karena itu, menggunakan kunci `aws:ResourceAccount` atau `s3:ResourceAccount` dalam kebijakan IAM Anda juga dapat memengaruhi akses ke sumber daya ini. Untuk informasi selengkapnya, lihat sumber daya berikut:

- Batasi akses ke bucket dalam Akun AWS yang ditentukan dalam Panduan AWS PrivateLink
- Batasi akses ke bucket yang digunakan Amazon ECR dalam Panduan Amazon ECR
- Menyediakan akses yang diperlukan ke Systems Manager untuk bucket Amazon S3 AWS terkelola dalam Panduan AWS Systems Manager

Untuk informasi selengkapnya tentang kunci *`s3:ResourceAccount`* kondisi *`aws:ResourceAccount`* dan contoh yang menunjukkan cara menggunakannya, lihat Membatasi akses ke bucket Amazon S3 yang dimiliki oleh spesifik Akun AWS di Blog Penyimpanan.AWS

Contoh 6: Membutuhkan versi TLS minimum

Anda dapat menggunakan tombol `s3:TlsVersion` kondisi untuk menulis IAM, virtual private cloud endpoint (VPCE), atau kebijakan bucket yang membatasi akses pengguna atau aplikasi ke bucket Amazon S3 berdasarkan versi TLS yang digunakan oleh klien. Anda dapat menggunakan kunci syarat ini untuk tulis kebijakan yang memerlukan versi TLS minimum.

> **Note**
>
> Ketika AWS layanan melakukan panggilan ke AWS layanan lain atas nama Anda (service-to-service panggilan), konteks otorisasi khusus jaringan tertentu dihapus, termasuk`s3:TlsVersion`,,, dan. `aws:SecureTransport` `aws:SourceIp` `aws:VpcSourceIp` Jika kebijakan Anda menggunakan kunci kondisi ini dengan `Deny` pernyataan, prinsip AWS layanan mungkin diblokir secara tidak sengaja. Untuk memungkinkan AWS layanan berfungsi dengan baik sambil mempertahankan persyaratan keamanan Anda, kecualikan prinsip layanan dari `Deny` pernyataan Anda dengan menambahkan kunci `aws:PrincipalIsAWSService` kondisi dengan nilai. `false` Misalnya:
>
> ```
> {
>   "Effect": "Deny",
>   "Action": "s3:*",
>   "Resource": "*",
>   "Condition": {
> ```

```
    "Bool": {
      "aws:SecureTransport": "false",
      "aws:PrincipalIsAWSService": "false"
    }
  }
}
```

Kebijakan ini menolak akses ke operasi S3 ketika HTTPS tidak `aws:SecureTransport` digunakan (salah), tetapi hanya untuk prinsipal AWS non-layanan. Ini memastikan pembatasan bersyarat Anda berlaku untuk semua kepala sekolah kecuali kepala sekolah layanan. AWS

Example

Contoh kebijakan bucket berikut menolak *`PutObject`* permintaan klien yang memiliki versi TLS lebih awal dari 1.2, misalnya 1.1 atau 1.0. Untuk menggunakan kebijakan contoh ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket1",
                "arn:aws:s3:::amzn-s3-demo-bucket1/*"
            ],
            "Condition": {
                "NumericLessThan": {
                    "s3:TlsVersion": 1.2
                }
            }
        }
    ]
}
```

Example

Contoh kebijakan bucket berikut memungkinkan `PutObject` permintaan oleh klien yang memiliki versi TLS lebih lambat 1.1, misalnya, 1.2, 1.3, atau yang lebih baru:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket1",
                "arn:aws:s3:::amzn-s3-demo-bucket1/*"
            ],
            "Condition": {
                "NumericGreaterThan": {
                    "s3:TlsVersion": 1.1
                }
            }
        }
    ]
}
```

Contoh 7: Mengecualikan prinsipal tertentu dari pernyataan **`Deny`**

Kebijakan bucket berikut menolak `s3:GetObject` akses ke*amzn-s3-demo-bucket*, kecuali ke kepala sekolah dengan nomor rekening. *123456789012* Untuk menggunakan kebijakan contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyAccessFromPrincipalNotInSpecificAccount",
      "Principal": {
        "AWS": "*"
      },
      "Action": "s3:GetObject",
```

```
      "Effect": "Deny",
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ],
      "Condition": {
        "StringNotEquals": {
          "aws:PrincipalAccount": [
            "123456789012"
          ]
        }
      }
    }
  ]
}
```

Contoh 8: Menegakkan klien untuk mengunggah objek secara kondisional berdasarkan nama kunci objek atau ETags

Dengan penulisan bersyarat, Anda dapat menambahkan header tambahan ke `WRITE` permintaan Anda untuk menentukan prasyarat untuk operasi S3 Anda. Header ini menentukan kondisi yang, jika tidak terpenuhi, akan mengakibatkan operasi S3 gagal. Misalnya, Anda dapat mencegah penimpaan data yang ada dengan memvalidasi tidak ada objek dengan nama kunci yang sama yang sudah ada di bucket Anda selama pengunggahan objek. Anda juga dapat memeriksa tag entitas objek (ETag) di Amazon S3 sebelum menulis objek.

Untuk contoh kebijakan bucket yang menggunakan kondisi dalam kebijakan bucket guna menerapkan penulisan bersyarat, lihat. the section called “Menegakkan penulisan bersyarat”

Contoh: Kunci kondisi Amazon S3 untuk operasi bucket

Contoh kebijakan berikut menunjukkan cara Anda dapat menggunakan kunci kondisi khusus Amazon S3 untuk operasi bucket.

Topik

- Contoh 1: Memberikan s3:GetObject izin dengan syarat pada alamat IP
- Contoh 2: Mendapatkan daftar objek di bucket dengan prefiks tertentu
- Contoh 3: Mengatur jumlah kunci maksimum

Contoh 1: Memberikan **`s3:GetObject`** izin dengan syarat pada alamat IP

Anda dapat memberikan izin kepada pengguna yang diautentikasi untuk menggunakan `s3:GetObject` tindakan jika permintaan berasal dari rentang alamat IP tertentu (misalnya,*`192.0.2.*`*), kecuali alamat IP adalah salah satu yang ingin Anda kecualikan (misalnya,*`192.0.2.188`*). Di `Condition` blok, `IpAddress` dan `NotIpAddress` kondisi, dan setiap kondisi disediakan pasangan kunci-nilai untuk evaluasi. Kedua pasangan kunci-nilai dalam contoh ini menggunakan kunci `aws:SourceIp` AWS lebar. Untuk menggunakan kebijakan contoh ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

> **Note**
>
> Nilai `IPAddress` dan `NotIpAddress` kunci yang ditentukan dalam `Condition` blok menggunakan notasi CIDR, seperti yang dijelaskan dalam RFC 4632. Untuk informasi lebih lanjut, lihat http://www.rfc-editor. org/rfc/rfc4632.txt.

```
{
    "Version": "2012-10-17",
    "Id": "S3PolicyId1",
    "Statement": [
        {
            "Sid": "statement1",
            "Effect": "Allow",
            "Principal": "*",
            "Action":"s3:GetObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Condition" : {
                "IpAddress" : {
                    "aws:SourceIp": "192.0.2.0/24"
                },
                "NotIpAddress" : {
                    "aws:SourceIp": "192.0.2.188/32"
                }
            }
        }
    ]
}
```

Anda juga dapat menggunakan kunci kondisi AWS-wide lainnya dalam kebijakan Amazon S3. Misalnya, Anda dapat menentukan kunci kondisi `aws:SourceVpce` dan `aws:SourceVpc` dalam

kebijakan bucket untuk titik akhir VPC. Untuk contoh spesifik, lihatMegendalikan akses dari titik akhir VPC dengan kebijakan bucket.

> **Note**
>
> Untuk beberapa kunci kondisi AWS global, hanya jenis sumber daya tertentu yang didukung. Oleh karena itu, periksa apakah Amazon S3 mendukung kunci kondisi global dan jenis sumber daya yang ingin Anda gunakan, atau apakah Anda perlu menggunakan kunci kondisi khusus Amazon S3. Untuk daftar lengkap jenis sumber daya dan kunci kondisi yang didukung untuk Amazon S3, lihat Tindakan, sumber daya, dan kunci kondisi untuk Amazon S3 di Referensi Otorisasi Layanan.
> Untuk informasi selengkapnya tentang izin operasi S3 API menurut jenis sumber daya S3, lihat. Izin yang diperlukan untuk operasi API Amazon S3

Contoh 2: Mendapatkan daftar objek di bucket dengan prefiks tertentu

Anda dapat menggunakan kunci `s3:prefix` kondisi untuk membatasi respons operasi ListObjectsV2API ke nama kunci dengan awalan tertentu. Jika Anda adalah pemilik bucket, Anda dapat menggunakan kunci kondisi ini untuk membatasi pengguna agar mencantumkan konten awalan tertentu di bucket. Kunci `s3:prefix` kondisi berguna jika objek dalam bucket diatur oleh awalan nama kunci.

Konsol Amazon S3 menggunakan prefiks nama kunci untuk menampilkan konsep folder. Hanya konsol tersebut yang mendukung konsep folder; Amazon S3 API hanya mendukung bucket dan objek. Misalnya, jika Anda memiliki dua objek dengan nama kunci *`public/object1.jpg`* dan*`public/object2.jpg`*, konsol menunjukkan objek di bawah *`public`* folder. Pada Amazon S3 API, objek-objek ini adalah objek dengan prefiks, bukan objek yang ada dalam folder. Untuk informasi lebih lanjut tentang bagaimana menggunakan prefiks dan pembatas untuk memfilter izin akses, lihat Mengontrol akses ke bucket dengan kebijakan pengguna.

Dalam skenario berikut, pemilik bucket dan akun induk yang dimiliki pengguna adalah sama. Jadi pemilik bucket dapat menggunakan kebijakan bucket atau kebijakan pengguna untuk memberikan akses. Untuk informasi selengkapnya tentang kunci kondisi lain yang dapat Anda gunakan dengan operasi `ListObjectsV2` API, lihat ListObjectsV2.

> **Note**
>
> Jika bucket diaktifkan versi, untuk mencantumkan objek di bucket, Anda harus memberikan `s3:ListBucketVersions` izin dalam kebijakan berikut, bukan izin. `s3:ListBucket` `s3:ListBucketVersions`Izin juga mendukung kunci `s3:prefix` kondisi.

Kebijakan pengguna

Kebijakan pengguna berikut memberikan `s3:ListBucket` izin (lihat ListObjectsV2) dengan `Condition` pernyataan yang mengharuskan pengguna untuk menentukan awalan dalam permintaan dengan nilai. *`projects`* Untuk menggunakan kebijakan contoh ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"statement1",
         "Effect":"Allow",
         "Action": "s3:ListBucket",
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket",
         "Condition" : {
             "StringEquals" : {
                 "s3:prefix": "projects"
             }
          }
       },
      {
         "Sid":"statement2",
         "Effect":"Deny",
         "Action": "s3:ListBucket",
         "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
         "Condition" : {
             "StringNotEquals" : {
                 "s3:prefix": "projects"
             }
          }
       }
    ]
}
```

`Condition`Pernyataan membatasi pengguna untuk mencantumkan hanya kunci objek yang memiliki *`projects`* awalan. `Deny`Pernyataan eksplisit yang ditambahkan menyangkal pengguna dari mencantumkan kunci dengan awalan lain, apa pun izin lain yang mungkin dimiliki pengguna. Misalnya, pengguna dapat memperoleh izin untuk membuat daftar kunci objek tanpa batasan apa pun, baik melalui pembaruan pada kebijakan pengguna sebelumnya atau melalui kebijakan bucket. Karena `Deny` pernyataan eksplisit selalu mengganti `Allow` pernyataan, jika pengguna mencoba membuat daftar kunci selain yang memiliki *`projects`* awalan, permintaan ditolak.

Kebijakan bucket

Jika Anda menambahkan `Principal` elemen ke kebijakan pengguna di atas, mengidentifikasi pengguna, Anda sekarang memiliki kebijakan bucket, seperti yang ditunjukkan pada contoh berikut. Untuk menggunakan kebijakan contoh ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"statement1",
         "Effect":"Allow",
         "Principal": {
            "AWS": "arn:aws:iam::123456789012:user/bucket-owner"
         },
         "Action":  "s3:ListBucket",
         "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
         "Condition" : {
             "StringEquals" : {
                 "s3:prefix": "projects"
             }
          }
       },
      {
         "Sid":"statement2",
         "Effect":"Deny",
         "Principal": {
            "AWS": "arn:aws:iam::123456789012:user/bucket-owner"
         },
         "Action": "s3:ListBucket",
         "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
         "Condition" : {
             "StringNotEquals" : {
```

```
                "s3:prefix": "projects"
            }
        }
    }
  ]
}
```

Uji kebijakan dengan AWS CLI

Anda dapat menguji kebijakan menggunakan `list-object` AWS CLI perintah berikut. Dalam perintah ini, Anda memberikan kredensial pengguna dengan menggunakan parameter `--profile`. Untuk informasi selengkapnya tentang pengaturan dan penggunaan AWS CLI, lihat Mengembangkan dengan Amazon S3 menggunakan AWS CLI di Referensi API Amazon S3.

```
aws s3api list-objects --bucket amzn-s3-demo-bucket --prefix projects --
profile AccountA
```

Contoh 3: Mengatur jumlah kunci maksimum

Anda dapat menggunakan tombol `s3:max-keys` kondisi untuk mengatur jumlah maksimum kunci yang dapat dikembalikan oleh pemohon dalam ListObjectVersionspermintaan ListObjectsV2atau permintaan. Secara default, operasi API ini mengembalikan hingga 1.000 kunci. Untuk daftar operator kondisi numerik yang dapat Anda gunakan dengan `s3:max-keys` dan contoh yang menyertainya, lihat Operator kondisi Numerik dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas untuk Amazon S3

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi sumber daya Amazon S3. Mereka juga tidak dapat melakukan tugas dengan menggunakan AWS Management Console, AWS Command Line Interface (AWS CLI), atau AWS API. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM dengan menggunakan contoh dokumen kebijakan JSON ini, lihat Membuat kebijakan IAM (konsol) di Panduan Pengguna IAM.

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh Amazon S3, termasuk format ARNs untuk setiap jenis sumber daya, lihat Kunci tindakan, sumber daya, dan kondisi untuk Amazon S3 di Referensi Otorisasi Layanan.

Untuk informasi selengkapnya tentang izin operasi S3 API menurut jenis sumber daya S3, lihat. Izin yang diperlukan untuk operasi API Amazon S3

Topik

- Praktik terbaik kebijakan
- Mengontrol akses ke bucket dengan kebijakan pengguna
- Contoh kebijakan berbasis identitas untuk Amazon S3

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya Amazon S3 di akun Anda. Tindakan ini membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat Kebijakan yang dikelola AWS atau Kebijakan yang dikelola AWS untuk fungsi tugas dalam Panduan Pengguna IAM.
- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk mengajukan izin, lihat Kebijakan dan izin dalam IAM dalam Panduan Pengguna IAM.
- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti AWS CloudFormation. Untuk informasi selengkapnya, lihat Elemen kebijakan JSON IAM: Kondisi dalam Panduan Pengguna IAM.

- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat Validasi kebijakan dengan IAM Access Analyzer dalam Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat Amankan akses API dengan MFA dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat Praktik terbaik keamanan di IAM dalam Panduan Pengguna IAM.

Mengontrol akses ke bucket dengan kebijakan pengguna

Panduan ini menjelaskan cara kerja izin pengguna dengan Amazon S3. Dalam contoh ini, Anda membuat bucket dengan folder. Anda kemudian membuat pengguna AWS Identity and Access Management IAM di Anda Akun AWS dan memberikan izin tambahan kepada pengguna tersebut di bucket Amazon S3 Anda dan folder di dalamnya.

Topik

- Basic bucket dan folder
- Ringkasan panduan
- Bersiap untuk panduan
- Langkah 1: Buat bucket
- Langkah 2: Buat pengguna dan grup IAM
- Langkah 3: Verifikasi bahwa pengguna IAM tidak memiliki izin
- Langkah 4: Berikan izin tingkat grup
- Langkah 5: Berikan izin tertentu kepada Alice pengguna IAM
- Langkah 6: Berikan izin tertentu kepada Bob pengguna IAM
- Langkah 7: Amankan folder privat

- Langkah 8: Membersihkan
- Sumber daya terkait

Basic bucket dan folder

Model data Amazon S3 bersifat struktur yang datar: Anda membuat bucket, dan bucket menyimpan objek tersebut. Tidak ada hierarki sub-bucket atau sub-folder, tetapi Anda dapat berusaha menyamai hierarki folder. Alat-alat seperti konsol Amazon S3 dapat menyajikan tampilan folder dan sub-folder logika ini di dalam bucket.

Konsol tersebut menunjukkan bahwa bucket bernama `companybucket` memiliki tiga folder, `Private`, `Development`, dan `Finance`, dan sebuah objek, `s3-dg.pdf`. Konsol tersebut menggunakan nama objek (kunci) untuk membuat hierarki logis dengan folder dan subfolder. Pertimbangkan contoh berikut:

- Saat Anda membuat folder `Development`, konsol tersebut membuat objek dengan kunci `Development/`. Perhatikan jejak garis miring (`/`) pembatas.
- Saat Anda mengunggah objek yang diberi nama `Projects1.xls` dalam folder `Development`, konsol mengunggah objek dan memberikannya kunci `Development/Projects1.xls`.

  Dalam kunci tersebut, `Development` adalah prefiks dan `/` adalah pembatas. Amazon S3 API mendukung prefiks dan pembatas dalam operasinya. Sebagai contoh, Anda dapat memperoleh daftar semua objek dari bucket dengan prefiks dan pembatas tertentu. Pada konsol, saat Anda membuka folder `Development`, konsol akan mencantumkan objek yang ada di folder tersebut. Pada contoh berikut, folder `Development` berisi satu objek.

  Saat konsol mencantumkan folder `Development` di dalam bucket `companybucket`, konsol tersebut mengirimkan permintaan ke Amazon S3 yang menyebutkan prefiks dari `Development` dan pembatas dari `/` dalam permintaan. Respons konsol terlihat seperti daftar folder di sistem file komputer Anda. Contoh sebelumnya menunjukkan bahwa bucket `companybucket` memiliki sebuah objek dengan kunci `Development/Projects1.xls`.

Konsol menggunakan kunci objek untuk membuat hierarki logika. Amazon S3 tidak memiliki hierarki fisik. Amazon S3 hanya memiliki bucket yang berisi objek dalam struktur file datar. Saat Anda membuat objek menggunakan Amazon S3 API, Anda dapat menggunakan kunci objek yang menyiratkan hierarki logika. Saat Anda membuat hierarki logika dari objek-objek tersebut, Anda dapat mengelola akses ke setiap folder, seperti yang ditunjukkan oleh panduan ini.

Sebelum memulai, pastikan bahwa Anda sudah terbiasa dengan konsep konten bucket tingkat root. Misalkan bucket companybucket memiliki objek-objek berikut:

- Private/privDoc1.txt
- Private/privDoc2.zip
- Development/project1.xls
- Development/project2.xls
- Finance/Tax2011/document1.pdf
- Finance/Tax2011/document2.pdf
- s3-dg.pdf

Kunci-kunci objek ini membuat hierarki logika dengan Private, Development, dan sebagai folder tingkat root Finance dan sebagai objek tingkat root s3-dg.pdf. Saat Anda memilih nama bucket di konsol Amazon S3, item tingkat root muncul. Konsol menunjukkan prefiks tingkat atas (Private/, Development/, dan Finance/) sebagai folder tingkat root. Kunci objek s3-dg.pdf tidak memiliki prefiks, sehingga muncul sebagai item tingkat root.

Ringkasan panduan

Dalam panduan ini, Anda membuat sebuah bucket dengan tiga folder (Private, Development, dan Finance) di dalamnya.

Anda memiliki dua pengguna, Alice dan Bob. Anda ingin Alice hanya mengakses folder Development, dan Anda ingin Bob hanya mengakses folder Finance. Anda ingin membuat konten folder Private tetap privat. Dalam panduan, Anda mengelola akses dengan membuat pengguna IAM (contoh menggunakan nama pengguna Alice dan Bob) dan memberi mereka izin yang diperlukan.

IAM juga mendukung pembuatan grup pengguna dan pemberian izin tingkat grup yang berlaku untuk semua pengguna yang ada dalam grup. Hal ini membantu Anda untuk mengelola izin dengan lebih baik. Untuk latihan ini, baik Alice maupun Bob memerlukan beberapa izin yang sama. Jadi, Anda juga membuat kelompok bernama Consultants, lalu menambahkan Alice dan Bob ke dalam grup tersebut. Anda terlebih dahulu harus memberikan izin dengan melampirkan kebijakan grup ke grup tersebut. Kemudian Anda menambahkan izin kustom pengguna dengan melampirkan kebijakan kepada pengguna tertentu.

> **Note**
>
> Panduan menggunakan `companybucket` sebagai nama bucket, Alice dan Bob sebagai pengguna IAM, dan `Consultants` sebagai nama grup. Karena Amazon S3 mewajibkan bahwa nama bucket harus bersifat unik secara global, maka Anda harus mengganti nama bucket dengan nama yang Anda buat.

Bersiap untuk panduan

Dalam contoh ini, Anda menggunakan Akun AWS kredensil Anda untuk membuat pengguna IAM. Pada awalnya, pengguna ini tidak memiliki izin. Anda kemudian memberikan pengguna tersebut izin tambahan untuk melakukan tindakan Amazon S3 tertentu. Untuk menguji izin ini, Anda masuk ke konsol dengan menggunakan kredensial masing-masing pengguna. Saat Anda secara bertahap memberikan izin sebagai Akun AWS pemilik dan menguji izin sebagai pengguna IAM, Anda harus masuk dan keluar, setiap kali menggunakan kredensil yang berbeda. Anda dapat melakukan pengujian ini dengan satu browser, tetapi prosesnya akan berjalan lebih cepat jika Anda dapat menggunakan dua browser yang berbeda. Gunakan satu browser untuk terhubung AWS Management Console dengan Akun AWS kredensyal Anda dan browser lain untuk terhubung dengan kredensyal pengguna IAM.

Untuk masuk ke AWS Management Console dengan Akun AWS kredensi Anda, buka/. https://console.aws.amazon.com Pengguna IAM tidak dapat masuk menggunakan tautan yang sama. Pengguna IAM harus menggunakan halaman masuk dengan IAM yang diaktifkan. Sebagai pemilik akun, Anda dapat memberikan tautan ini kepada pengguna Anda.

Untuk informasi lebih lanjut tentang IAM, lihat Halaman Masuk AWS Management Console dalam Panduan Pengguna IAM.

Untuk menyediakan tautan masuk bagi pengguna IAM

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Pada jendela Navigasi, pilih Dasbor IAM .
3. Perhatikan URL dalam Tautan masuk pengguna IAM:. Anda akan memberikan tautan ini ke pengguna IAM untuk masuk ke konsol dengan nama pengguna dan kata sandi IAM mereka.

Langkah 1: Buat bucket

Pada langkah ini, Anda masuk ke konsol Amazon S3 dengan Akun AWS kredensialnya, membuat bucket, menambahkan folder ke bucket, dan mengunggah satu atau dua contoh dokumen di setiap folder.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Buat bucket.

   Untuk step-by-step instruksi, lihatMembuat ember tujuan umum.
3. Unggah satu dokumen ke bucket.

   Latihan ini mengasumsikan bahwa Anda memiliki dokumen `s3-dg.pdf` pada tingkat root bucket ini. Jika Anda mengunggah dokumen lain, ganti nama file untuk `s3-dg.pdf`.
4. Tambahkan tiga folder yang diberi nama `Private`, `Finance`, dan `Development` ke dalam bucket.

   Untuk step-by-step petunjuk cara membuat folder, lihat Mengatur objek di konsol Amazon S3 dengan menggunakan folder > di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.
5. Unggah satu atau dua dokumen ke setiap folder.

   Untuk latihan ini, asumsikan bahwa Anda telah mengunggah beberapa dokumen di setiap folder, sehingga bucket tersebut memiliki objek dengan kunci-kunci berikut:

   - `Private/privDoc1.txt`
   - `Private/privDoc2.zip`
   - `Development/project1.xls`
   - `Development/project2.xls`
   - `Finance/Tax2011/document1.pdf`
   - `Finance/Tax2011/document2.pdf`
   - `s3-dg.pdf`

   Untuk step-by-step instruksi, lihatMengunggah Objek.

Langkah 2: Buat pengguna dan grup IAM

Sekarang gunakan Konsol IAM untuk menambahkan dua pengguna IAM, Alice dan Bob, ke pengguna Anda. Akun AWS Untuk step-by-step petunjuk, lihat Membuat pengguna IAM di Panduan Pengguna IAM Anda Akun AWS.

Juga buat grup administratif bernama`Consultants`. Kemudian tambahkan kedua pengguna ke grup. Untuk step-by-step petunjuknya, lihat Membuat grup pengguna IAM.

> **Warning**
>
> Saat Anda menambahkan pengguna dan grup, jangan melampirkan kebijakan apa pun yang memberikan izin kepada pengguna-pengguna ini. Awalnya, para pengguna ini tidak memiliki izin. Dalam bagian berikut ini, Anda memberikan izin sebagai tambahan. Pertama, Anda harus memastikan bahwa Anda telah menetapkan kata sandi untuk para pengguna IAM ini. Anda menggunakan kredensial para pengguna ini untuk menguji tindakan Amazon S3 dan memverifikasi bahwa izin berfungsi sesuai harapan.

Untuk step-by-step petunjuk cara membuat pengguna IAM baru, lihat Membuat pengguna IAM di Panduan Pengguna IAM Anda Akun AWS. Saat Anda membuat pengguna untuk panduan ini, pilih akses AWS Management Console dan hapus Akses terprogram.

Untuk step-by-step petunjuk cara membuat grup administratif, lihat Membuat Pengguna dan Grup Admin IAM Pertama Anda di Panduan Pengguna IAM.

Langkah 3: Verifikasi bahwa pengguna IAM tidak memiliki izin

Jika Anda menggunakan dua browser, sekarang Anda dapat menggunakan browser kedua untuk masuk ke konsol menggunakan salah satu kredensial pengguna IAM.

1. Menggunakan tautan masuk pengguna IAM (lihat Untuk menyediakan tautan masuk bagi pengguna IAM), masuklah ke AWS Management Console menggunakan salah satu kredensial pengguna IAM.
2. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

   Verifikasi pesan konsol yang memberi tahu Anda bahwa akses ditolak.

Sekarang, Anda dapat mulai memberikan izin tambahan kepada pengguna. Pertama, Anda harus melampirkan kebijakan grup yang memberikan izin yang harus dimiliki kedua pengguna.

Langkah 4: Berikan izin tingkat grup

Anda ingin pengguna dapat melakukan hal berikut ini:

- Cantumkan semua bucket yang dimiliki oleh akun induk. Untuk melakukannya, Bob dan Alice harus memiliki izin untuk tindakan `s3:ListAllMyBuckets`.
- Cantumkan item tingkat root, folder, dan objek dalam bucket `companybucket`. Untuk melakukannya, Bob dan Alice harus memiliki izin untuk tindakan `s3:ListBucket` pada bucket `companybucket`.

Pertama, Anda harus membuat kebijakan yang memberikan izin-izin tersebut, lalu Anda harus melampirkannya ke grup `Consultants`.

Langkah 4.1: Berikan izin untuk mencantumkan semua bucket

Pada langkah ini, Anda membuat kebijakan terkelola yang memberikan izin minimum kepada para pengguna untuk memungkinkan mereka mencantumkan daftar semua bucket yang dimiliki oleh akun induk. Kemudian Anda melampirkan kebijakan tersebut ke grup `Consultants`. Saat Anda melampirkan kebijakan terkelola ke pengguna atau grup, Anda memberikan izin kepada pengguna atau grup untuk mendapatkan daftar bucket yang dimiliki oleh Akun AWS induk.

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.

   > **Note**
   >
   > Karena Anda memberikan izin kepada pengguna, masuklah dengan menggunakan kredensial Akun AWS Anda, bukan sebagai pengguna IAM.

2. Buat kebijakan terkelola.

   a. Pada panel navigasi yang ada di sebelah kiri, pilih Kebijakan, lalu pilih Buat Kebijakan.

   b. Pilih tab JSON.

   c. Salin kebijakan akses berikut dan tempelkan ke dalam kolom teks kebijakan.

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowGroupToSeeBucketListInTheConsole",
      "Action": ["s3:ListAllMyBuckets"],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::*"]
    }
  ]
}
```

Kebijakan adalah dokumen JSON. Dalam dokumen, `Statement` adalah inline objek, yang masing-masing menjelaskan izin dengan menggunakan kumpulan pasangan nilai-nama. Kebijakan sebelumnya menjelaskan satu izin tertentu. `Action` menyebutkan jenis akses. Dalam kebijakan, `s3:ListAllMyBuckets` adalah tindakan Amazon S3 yang telah ditentukan sebelumnya. Tindakan ini mencakup operasi Amazon S3 GET Service, yang mengembalikan daftar semua bucket yang dimiliki oleh pengirim yang diautentikasi. Nilai elemen `Effect` menentukan apakah izin tertentu diperbolehkan atau ditolak.

d. Pilih Tinjau Kebijakan. Pada halaman berikutnya, masukkan `AllowGroupToSeeBucketListInTheConsole` pada kolom Nama, dan kemudian pilih Buat kebijakan.

> **Note**
>
> Entri Ringkasan menampilkan pesan yang menyatakan bahwa kebijakan tidak memberikan izin apa pun. Untuk panduan ini, Anda dapat mengabaikan pesan ini dengan aman.

3. Lampirkan kebijakan terkelola `AllowGroupToSeeBucketListInTheConsole` yang Anda buat ke grup `Consultants`.

   Untuk step-by-step petunjuk untuk melampirkan kebijakan terkelola, lihat Menambahkan dan menghapus izin identitas IAM di Panduan Pengguna IAM.

   Anda melampirkan dokumen kebijakan ke pengguna dan grup IAM di konsol IAM. Karena Anda ingin kedua pengguna dapat mencantumkan bucket, Anda harus melampirkan kebijakan ke grup.

4. Uji izin.

a. Menggunakan tautan masuk pengguna IAM (lihat Untuk menyediakan tautan masuk bagi pengguna IAM), masuklah ke konsol dengan menggunakan salah satu kredensial pengguna IAM.

b. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

   Konsol tersebut harus mencantumkan semua bucket tetapi bukan objek yang ada dalam bucket mana pun.

Langkah 4.2: Mengaktifkan pengguna untuk mencantumkan konten tingkat root dalam bucket

Selanjutnya, Anda mengizinkan semua pengguna di grup `Consultants` untuk mencantumkan item bucket `companybucket` tingkat root. Ketika pengguna memilih bucket perusahaan pada konsol Amazon S3, pengguna dapat melihat item tingkat root yang ada dalam bucket.

> **Note**
>
> Contoh ini menggunakan `companybucket` sebagai ilustrasi. Anda harus menggunakan nama bucket yang Anda buat.

Untuk memahami permintaan yang dikirim konsol ke Amazon S3 saat Anda memilih nama bucket, respons yang dikembalikan Amazon S3, dan bagaimana konsol menafsirkan respons, periksa alirannya sedikit lebih dekat.

Saat Anda memilih nama bucket, konsol mengirimkan permintaan GET Bucket (List Objects) ke Amazon S3. Permintaan ini mencakup parameter-parameter berikut ini:

- Parameter `prefix` dengan string kosong sebagai nilainya.
- Parameter `delimiter` dengan `/` sebagai nilainya.

Berikut ini adalah contoh permintaan.

```
GET ?prefix=&delimiter=/ HTTP/1.1
Host: companybucket.s3.amazonaws.com
Date: Wed, 01 Aug  2012 12:00:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:xQE0diMbLRepdf3YB+FIEXAMPLE=
```

Amazon S3 mengembalikan respons yang menyertakan elemen <`ListBucketResult/`> berikut ini.

```
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>companybucket</Name>
  <Prefix></Prefix>
  <Delimiter>/</Delimiter>
   ...
  <Contents>
    <Key>s3-dg.pdf</Key>
    ...
  </Contents>
  <CommonPrefixes>
    <Prefix>Development/</Prefix>
  </CommonPrefixes>
  <CommonPrefixes>
    <Prefix>Finance/</Prefix>
  </CommonPrefixes>
  <CommonPrefixes>
    <Prefix>Private/</Prefix>
  </CommonPrefixes>
</ListBucketResult>
```

Kuncinya `s3-dg.pdf` objek tidak berisi pembatas garis miring (/), dan Amazon S3 mengembalikan kunci dalam elemen `<Contents>`. Namun, semua kunci lain dalam contoh bucket berisi pembatas /. Amazon S3 mengelompokkan kunci-kunci ini dan mengembalikan elemen `<CommonPrefixes>` untuk setiap nilai prefiks yang berbeda `Development/`, `Finance/`, dan `Private/` yakni substring dari awal kunci-kunci tersebut hingga peristiwa pertama pembatas / yang ditentukan.

Konsol menafsirkan hasil ini dan menampilkan item tingkat root sebagai tiga folder dan satu kunci objek.

Jika Bob atau Alice membuka folder Pengembangan, konsol akan mengirim permintaan GET Bucket (List Objects) ke Amazon S3 dengan parameter `prefix` dan `delimiter` yang ditetapkan dengan nilai-nilai berikut:

- Parameter `prefix` dengan nilai `Development/`.
- Parameter `delimiter` dengan nilai "/".

Sebagai respons, Amazon S3 mengembalikan kunci-kunci objek yang dimulai dengan prefiks yang ditentukan.

```
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
```

```
  <Name>companybucket</Name>
  <Prefix>Development</Prefix>
  <Delimiter>/</Delimiter>
   ...
  <Contents>
    <Key>Project1.xls</Key>
    ...
  </Contents>
  <Contents>
    <Key>Project2.xls</Key>
    ...
  </Contents>
</ListBucketResult>
```

Konsol menampilkan kunci-kunci objek.

Sekarang, kembali ke pemberian izin pengguna untuk mencantumkan item bucket tingkat root. Untuk mencantumkan konten bucket, pengguna memerlukan izin untuk memanggil tindakan `s3:ListBucket`, sebagaimana yang diperlihatkan dalam pernyataan kebijakan berikut. Untuk memastikan bahwa mereka hanya melihat konten tingkat root, Anda harus menambahkan kondisi yang harus ditetapkan pengguna sebagai `prefix` kosong dalam permintaan—yaitu, mereka tidak diizinkan melakukan klik dua kali pada folder tingkat root. Terakhir, Anda menambahkan kondisi untuk mengharuskan akses gaya folder dengan mengharuskan permintaan pengguna untuk menyertakan parameter `delimiter` dengan nilai "/".

```
{
  "Sid": "AllowRootLevelListingOfCompanyBucket",
  "Action": ["s3:ListBucket"],
  "Effect": "Allow",
  "Resource": ["arn:aws:s3:::companybucket"],
  "Condition":{
         "StringEquals":{
             "s3:prefix":[""], "s3:delimiter":["/"]
                       }
              }
}
```

Saat Anda memilih bucket di konsol Amazon S3, konsol akan mengirimkan permintaan lokasi GET Bucket terlebih dahulu untuk menemukan Wilayah AWS tempat bucket digunakan. Kemudian konsol tersebut akan menggunakan titik akhir spesifik Wilayah untuk bucket untuk mengirimkan permintaan GET Bucket (List Objects). Akibatnya, jika pengguna akan menggunakan konsol, maka

Anda harus memberikan izin untuk tindakan `s3:GetBucketLocation` seperti yang ditunjukkan dalam pernyataan kebijakan berikut.

```
{
   "Sid": "RequiredByS3Console",
   "Action": ["s3:GetBucketLocation"],
   "Effect": "Allow",
   "Resource": ["arn:aws:s3:::*"]
}
```

Untuk memungkinkan pengguna mencantumkan konten bucket tingkat root

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.

   Gunakan Akun AWS kredensil Anda, bukan kredensil pengguna IAM, untuk masuk ke konsol.

2. Ganti kebijakan terkelola `AllowGroupToSeeBucketListInTheConsole` yang sudah ada yang terlampir pada grup `Consultants` dengan kebijakan berikut, yang juga memungkinkan tindakan `s3:ListBucket`. Ingatlah untuk mengganti *`companybucket`* dalam kebijakan `Resource` dengan nama ember Anda.

   Untuk step-by-step petunjuk, lihat Mengedit kebijakan IAM di Panduan Pengguna IAM. Saat mengikuti step-by-step instruksi, pastikan untuk mengikuti langkah-langkah untuk menerapkan perubahan Anda ke semua entitas utama yang dilampirkan kebijakan tersebut.

   ```
   {
     "Version": "2012-10-17",
     "Statement": [
        {
          "Sid":
    "AllowGroupToSeeBucketListAndAlsoAllowGetBucketLocationRequiredForListBucket",
          "Action": [ "s3:ListAllMyBuckets", "s3:GetBucketLocation" ],
          "Effect": "Allow",
          "Resource": [ "arn:aws:s3:::*"  ]
        },
        {
          "Sid": "AllowRootLevelListingOfCompanyBucket",
          "Action": ["s3:ListBucket"],
          "Effect": "Allow",
          "Resource": ["arn:aws:s3:::companybucket"],
          "Condition":{
   ```

```
            "StringEquals":{
                   "s3:prefix":[""], "s3:delimiter":["/"]
                          }
                  }
     }
  ]
}
```

3. Uji izin yang diperbarui.

   a. Menggunakan tautan masuk pengguna IAM (lihat Untuk menyediakan tautan masuk bagi pengguna IAM), masuklah ke AWS Management Console.

      Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

   b. Pilih bucket yang Anda buat, dan konsol akan menunjukkan item bucket tingkat root. Jika Anda memilih folder apa pun yang ada dalam bucket, maka Anda tidak akan dapat melihat konten folder tersebut karena Anda belum memberikan izin.

Pengujian ini berhasil ketika pengguna menggunakan konsol Amazon S3. Saat Anda memilih bucket pada konsol tersebut, implementasi konsol mengirim permintaan yang menyertakan parameter `prefix` dengan string kosong sebagai nilainya dan parameter `delimiter` dengan "`/`" sebagai nilainya.

Langkah 4.3: Ringkasan kebijakan grup

Hasil akhir dari kebijakan grup yang Anda tambahkan adalah memberikan kepada pengguna IAM Alice dan Bob paling tidak izin-izin berikut ini:

- Cantumkan semua bucket yang dimiliki oleh akun induk.
- Lihat item tingkat root di bucket `companybucket`.

Namun, pengguna masih tidak bisa melakukan banyak hal. Selanjutnya, Anda memberikan izin spesifik pengguna, sebagai berikut:

- Memungkinkan Alice untuk mendapatkan dan meletakkan objek di folder `Development`.
- Memungkinkan Bob untuk mendapatkan dan meletakkan objek di folder `Finance`.

Untuk izin-izin spesifik pengguna, Anda harus melampirkan kebijakan kepada pengguna tertentu, bukan ke grup. Pada bagian berikut, Anda memberikan izin kepada Alice untuk bekerja di folder

`Development`. Anda dapat mengulangi langkah untuk memberikan izin serupa kepada Bob untuk bekerja di folder `Finance`.

Langkah 5: Berikan izin tertentu kepada Alice pengguna IAM

Sekarang Anda memberikan izin tambahan kepada Alice sehingga dia dapat melihat konten folder `Development` dan mengambil dan meletakkan objek di folder itu.

Langkah 5.1: Berikan izin kepada Alice pengguna IAM untuk mencantumkan konten folder pengembangan

Agar Alice dapat mencantumkan konten `Development` folder, Anda harus menerapkan kebijakan kepada pengguna Alice yang memberikan izin untuk `s3:ListBucket` tindakan di `companybucket` bucket, asalkan permintaan tersebut menyertakan awalan. `Development/` Anda ingin kebijakan ini diterapkan hanya untuk pengguna Alice, sehingga Anda menggunakan kebijakan inline. Untuk informasi selengkapnya tentang kebijakan sebaris, lihat Kebijakan terkelola dan kebijakan sebaris di Panduan Pengguna IAM.

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.

   Gunakan Akun AWS kredensil Anda, bukan kredensil pengguna IAM, untuk masuk ke konsol.
2. Buat kebijakan selaras untuk memberikan izin kepada pengguna Alice untuk mencantumkan konten folder `Development`.

   a. Pada panel navigasi di sebelah kiri, pilih Pengguna.

   b. Pilih nama pengguna Alice.

   c. Pada halaman rincian pengguna, pilih tab Izin dan kemudian pilih Tambahkan kebijakan selaras.

   d. Pilih tab JSON.

   e. Salin kebijakan berikut, dan tempelkan ke bidang teks kebijakan.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
      "Sid": "AllowListBucketIfSpecificPrefixIsIncludedInRequest",
      "Action": ["s3:ListBucket"],
      "Effect": "Allow",
```

```
      "Resource": ["arn:aws:s3:::companybucket"],
      "Condition":{  "StringLike":{"s3:prefix":["Development/*"] }
       }
    }
  ]
}
```

f. Pilih Tinjau Kebijakan. Pada halaman berikutnya, masukkan nama dalam kolom Nama, dan kemudian pilih Buat kebijakan.

3. Uji perubahan izin Alice:

   a. Menggunakan tautan masuk pengguna IAM (lihat Untuk menyediakan tautan masuk bagi pengguna IAM), masuklah ke AWS Management Console.

   b. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

   c. Pada konsol Amazon S3, verifikasi bahwa Alice dapat melihat daftar objek di folder `Development/` di dalam bucket.

      Ketika pengguna memilih folder `/Development` untuk melihat daftar objek di dalamnya, konsol Amazon S3 mengirimkan permintaan `ListObjects` ke Amazon S3 dengan prefiks `/Development`. Karena pengguna diberikan izin untuk melihat daftar objek dengan prefiks `Development` dan pembatas `/`, Amazon S3 menampilkan daftar objek dengan key prefiks kunci `Development/`, dan konsol tersebut menampilkan daftar.

Langkah 5.2: Berikan izin kepada Alice pengguna IAM untuk mendapatkan dan menempatkan objek dalam folder pengembangan

Agar Alice mendapatkan dan meletakkan objek di folder `Development`, ia memerlukan izin untuk memanggil tindakan `s3:GetObject` dan `s3:PutObject`. Pernyataan kebijakan berikut memberikan izin-izin ini, dengan kondisi bahwa permintaan menyertakan parameter `prefix` dengan nilai `Development/`.

```
{
    "Sid":"AllowUserToReadWriteObjectData",
    "Action":["s3:GetObject", "s3:PutObject"],
    "Effect":"Allow",
    "Resource":["arn:aws:s3:::companybucket/Development/*"]
 }
```

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

   Gunakan Akun AWS kredensil Anda, bukan kredensil pengguna IAM, untuk masuk ke konsol.

2. Edit kebijakan selaras yang Anda buat pada langkah sebelumnya.

   a. Pada panel navigasi di sebelah kiri, pilih Pengguna.

   b. Pilih nama pengguna Alice.

   c. Pada halaman rincian pengguna, pilih tab Izin dan memperluas bagian Kebijakan Selaras.

   d. Di sebelah nama kebijakan yang Anda buat di langkah sebelumnya, pilih Edit Kebijakan.

   e. Salin kebijakan berikut ini, dan tempelkan ke dalam kolom teks kebijakan, yang akan menggantikan kebijakan yang sudah ada.

```
{
     "Version": "2012-10-17",
     "Statement":[
      {
         "Sid":"AllowListBucketIfSpecificPrefixIsIncludedInRequest",
         "Action":["s3:ListBucket"],
         "Effect":"Allow",
         "Resource":["arn:aws:s3:::companybucket"],
         "Condition":{
            "StringLike":{"s3:prefix":["Development/*"]
            }
         }
      },
      {
        "Sid":"AllowUserToReadWriteObjectDataInDevelopmentFolder",
        "Action":["s3:GetObject", "s3:PutObject"],
        "Effect":"Allow",
        "Resource":["arn:aws:s3:::companybucket/Development/*"]
      }
   ]
}
```

3. Uji kebijakan yang diperbarui:

   a. Menggunakan tautan masuk pengguna IAM (lihat Untuk menyediakan tautan masuk bagi pengguna IAM), masuklah ke AWS Management Console.

   b. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

c. Pada konsol Amazon S3, verifikasi bahwa Alice kini dapat menambahkan objek dan mengunduh objek di folder `Development`.

Langkah 5.3: Tolak secara eksplisit izin Alice pengguna IAM ke folder lain yang ada dalam bucket

Pengguna Alice kini dapat mencantumkan konten tingkat root di bucket `companybucket`. Dia juga dapat mengambil dan memasukkan objek di folder `Development`. Jika Anda benar-benar ingin mengencangkan izin akses, Anda dapat secara jelas menolak akses ke folder lain yang ada dalam bucket. Jika ada kebijakan lain (kebijakan bucket atau ACL) yang memberikan Alice akses ke folder lain dalam bucket, penolakan secara jelas ini menolak izin tersebut.

Anda dapat menambahkan pernyataan berikut ke kebijakan pengguna Alice yang mengharuskan semua permintaan yang dikirimkan Alice ke Amazon S3 harus menyertakan parameter `prefix`, yang nilainya dapat berupa `Development/*` atau string kosong.

```
{
   "Sid": "ExplicitlyDenyAnyRequestsForAllOtherFoldersExceptDevelopment",
   "Action": ["s3:ListBucket"],
   "Effect": "Deny",
   "Resource": ["arn:aws:s3:::companybucket"],
   "Condition":{  "StringNotLike": {"s3:prefix":["Development/*",""] },
                  "Null"         : {"s3:prefix":false }
    }
}
```

Ada dua ekspresi bersyarat di pemblokiran `Condition`. Hasil dari ekspresi kondisional ini dikombinasikan dengan menggunakan logis `AND`. Jika kedua syarat betul, hasil dari kondisi gabungan adalah BETUL. Karena `Effect` dalam kebijakan ini adalah `Deny`, saat `Condition` dievaluasi bernilai benar, maka pengguna tidak dapat menjalankan `Action` yang ditentukan.

- Ekspresi bersyarat `Null` memastikan bahwa permintaan dari Alice menyertakan parameter `prefix`.

  Parameter `prefix` memerlukan akses seperti folder. Jika Anda mengirim permintaan tanpa parameter `prefix`, Amazon S3 mengembalikan semua kunci objek.

  Jika permintaan menyertakan parameter `prefix` dengan nilai nol, maka ekspresi dievalusi bernilai benar, dan seluruh `Condition` dievaluasi bernilai benar. Anda harus mengizinkan string kosong sebagai nilai dari parameter `prefix`. Dari pembahasan sebelumnya, ingatlah

bahwa dengan memungkinkan string nol akan memungkinkan Alice mengambil item bucket tingkat root sebagaimana yang konsol lakukan dalam pembahasan sebelumnya. Untuk informasi selengkapnya, lihat Langkah 4.2: Mengaktifkan pengguna untuk mencantumkan konten tingkat root dalam bucket.

- Ekspresi kondisional `StringNotLike` memastikan bahwa jika nilai parameter `prefix` ditentukan dan bukan `Development/*`, maka permintaan gagal.

Ikuti langkah-langkah yang diuraikan pada bagian sebelumnya dan sekali lagi perbarui kebijakan selaras yang Anda buat untuk pengguna Alice.

Salin kebijakan berikut ini, dan tempelkan ke dalam kolom teks kebijakan, yang akan menggantikan kebijakan yang sudah ada.

```
{
   "Version": "2012-10-17",
   "Statement":[
      {
         "Sid":"AllowListBucketIfSpecificPrefixIsIncludedInRequest",
         "Action":["s3:ListBucket"],
         "Effect":"Allow",
         "Resource":["arn:aws:s3:::companybucket"],
         "Condition":{
            "StringLike":{"s3:prefix":["Development/*"]
            }
         }
      },
      {
        "Sid":"AllowUserToReadWriteObjectDataInDevelopmentFolder",
        "Action":["s3:GetObject", "s3:PutObject"],
        "Effect":"Allow",
        "Resource":["arn:aws:s3:::companybucket/Development/*"]
      },
      {
         "Sid": "ExplicitlyDenyAnyRequestsForAllOtherFoldersExceptDevelopment",
         "Action": ["s3:ListBucket"],
         "Effect": "Deny",
         "Resource": ["arn:aws:s3:::companybucket"],
         "Condition":{  "StringNotLike": {"s3:prefix":["Development/*",""] },
                        "Null"         : {"s3:prefix":false }
          }
      }
```

```
    ]
}
```

Langkah 6: Berikan izin tertentu kepada Bob pengguna IAM

Sekarang Anda ingin memberikan Bob izin ke folder `Finance`. Ikuti langkah-langkah yang telah Anda gunakan sebelumnya untuk memberikan izin kepada Alice, tetapi ganti folder `Development` dengan folder `Finance`. Untuk step-by-step instruksi, lihatLangkah 5: Berikan izin tertentu kepada Alice pengguna IAM.

Langkah 7: Amankan folder privat

Dalam contoh ini, Anda hanya memiliki dua pengguna. Anda memberikan semua izin minimum yang diperlukan di tingkat grup dan memberikan izin tingkat pengguna hanya ketika Anda benar-benar perlu untuk memberikan izin pada tingkat pengguna individu. Pendekatan ini membantu meminimalkan upaya untuk mengelola izin. Saat jumlah pengguna meningkat, mengelola izin dapat menjadi rumit. Misalnya, Anda tidak ingin siapa pun dari pengguna yang ada dalam contoh ini mengakses konten folder `Private`. Bagaimana Anda memastikan bahwa Anda tidak secara tidak sengaja memberikan izin pengguna ke `Private` folder? Anda menambahkan kebijakan yang secara jelas menolak akses ke folder. Penolakan secara jelas menolak izin lainnya.

Untuk memastikan bahwa folder `Private` tetap privat, Anda dapat menambahkan dua pernyataan penolakan berikut ke kebijakan grup:

- Tambahkan pernyataan berikut untuk secara jelas menolak tindakan apa pun terhadap sumber daya dalam folder `Private` (`companybucket/Private/*`).

  ```
  {
    "Sid": "ExplictDenyAccessToPrivateFolderToEveryoneInTheGroup",
    "Action": ["s3:*"],
    "Effect": "Deny",
    "Resource":["arn:aws:s3:::companybucket/Private/*"]
  }
  ```

- Anda juga menolak izin untuk tindakan cantumkan objek ketika permintaan menyebutkan prefiks `Private/`. Pada konsol, jika Bob atau Alice membuka folder `Private`, maka kebijakan ini akan menyebabkan Amazon S3 mengembalikan respons kesalahan.

  ```
  {
    "Sid": "DenyListBucketOnPrivateFolder",
  ```

```
  "Action": ["s3:ListBucket"],
  "Effect": "Deny",
  "Resource": ["arn:aws:s3:::*"],
  "Condition":{
      "StringLike":{"s3:prefix":["Private/"]}
   }
}
```

Mengganti kebijakan grup `Consultants` dengan kebijakan yang diperbarui yang menyertakan pernyataan penolakan sebelumnya. Setelah kebijakan yang diperbarui diterapkan, tidak ada pengguna dalam grup yang dapat mengakses folder `Private` di bucket Anda.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

   Gunakan Akun AWS kredensil Anda, bukan kredensil pengguna IAM, untuk masuk ke konsol.

2. Ganti kebijakan terkelola `AllowGroupToSeeBucketListInTheConsole` yang sudah ada yang terlampir pada grup `Consultants` dengan kebijakan berikut. Ingatlah untuk mengganti *`companybucket`* dalam kebijakan dengan nama bucket Anda.

   Untuk petunjuk, lihat Mengedit kebijakan yang dikelola pelanggan di Panduan Pengguna IAM. Ketika mengikuti petunjuk, pastikan untuk mengikuti petunjuk untuk menerapkan perubahan Anda ke semua entitas pengguna utama yang kepadanya kebijakan tersebut dilampirkan.

   ```
   {
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid":
    "AllowGroupToSeeBucketListAndAlsoAllowGetBucketLocationRequiredForListBucket",
         "Action": ["s3:ListAllMyBuckets", "s3:GetBucketLocation"],
         "Effect": "Allow",
         "Resource": ["arn:aws:s3:::*"]
       },
       {
         "Sid": "AllowRootLevelListingOfCompanyBucket",
         "Action": ["s3:ListBucket"],
         "Effect": "Allow",
         "Resource": ["arn:aws:s3:::companybucket"],
         "Condition":{
   ```

```
            "StringEquals":{"s3:prefix":[""]}
          }
      },
      {
        "Sid": "RequireFolderStyleList",
        "Action": ["s3:ListBucket"],
        "Effect": "Deny",
        "Resource": ["arn:aws:s3:::*"],
        "Condition":{
            "StringNotEquals":{"s3:delimiter":"/"}
          }
       },
      {
        "Sid": "ExplictDenyAccessToPrivateFolderToEveryoneInTheGroup",
        "Action": ["s3:*"],
        "Effect": "Deny",
        "Resource":["arn:aws:s3:::companybucket/Private/*"]
      },
      {
        "Sid": "DenyListBucketOnPrivateFolder",
        "Action": ["s3:ListBucket"],
        "Effect": "Deny",
        "Resource": ["arn:aws:s3:::*"],
        "Condition":{
            "StringLike":{"s3:prefix":["Private/"]}
          }
      }
    ]
}
```

Langkah 8: Membersihkan

Untuk membersihkan, buka Konsol IAM dan hapus pengguna Alice dan Bob. Untuk step-by-step petunjuk, lihat Menghapus pengguna IAM di Panduan Pengguna IAM.

Untuk memastikan bahwa Anda tidak lagi dikenakan biaya untuk penyimpanan, Anda juga harus menghapus objek dan bucket yang Anda buat untuk latihan ini.

Sumber daya terkait

- Mengelola kebijakan IAM dalam Panduan Pengguna IAM

Contoh kebijakan berbasis identitas untuk Amazon S3

Bagian ini menunjukkan beberapa contoh kebijakan berbasis identitas AWS Identity and Access Management (IAM) untuk mengontrol akses ke Amazon S3. Misalnya kebijakan bucket (kebijakan berbasis sumber daya), lihat. Kebijakan bucket untuk Amazon S3 Untuk informasi lebih lanjut tentang bahasa kebijakan IAM, lihatKebijakan dan izin di Amazon S3.

Contoh kebijakan berikut akan berfungsi jika Anda menggunakannya secara terprogram. Tetapi, untuk menggunakannya dengan konsol Amazon S3, Anda harus memberikan izin tambahan yang diperlukan oleh konsol. Untuk informasi tentang menggunakan kebijakan seperti ini dengan konsol Amazon S3, lihat Mengontrol akses ke bucket dengan kebijakan pengguna.

Untuk informasi selengkapnya tentang izin operasi S3 API menurut jenis sumber daya S3, lihat. Izin yang diperlukan untuk operasi API Amazon S3

Topik

- Mengizinkan akses pengguna IAM ke salah satu bucket Anda
- Mengizinkan setiap akses pengguna IAM ke folder dalam bucket
- Mengizinkan grup memiliki folder bersama di Amazon S3
- Mengizinkan semua pengguna Anda membaca objek di sebagian bucket
- Mengizinkan mitra untuk menempatkan file ke dalam bagian tertentu bucket
- Membatasi akses ke bucket Amazon S3 dalam bucket tertentu Akun AWS
- Membatasi akses ke bucket Amazon S3 dalam unit organisasi Anda
- Membatasi akses ke bucket Amazon S3 di organisasi Anda
- Memberikan izin untuk mengambil PublicAccessBlock konfigurasi untuk Akun AWS
- Membatasi pembuatan bucket ke satu Wilayah

Mengizinkan akses pengguna IAM ke salah satu bucket Anda

Dalam contoh ini, Anda ingin memberikan pengguna IAM dalam Akun AWS akses ke salah satu bucket Anda*amzn-s3-demo-bucket1*, dan memungkinkan pengguna untuk menambahkan, memperbarui, dan menghapus objek.

Selain memberikan izin `s3:PutObject`, `s3:GetObject`, dan `s3:DeleteObject` bagi pengguna, kebijakan tersebut juga memberikan izin `s3:ListAllMyBuckets`, `s3:GetBucketLocation`, dan `s3:ListBucket`. Izin-izin tersebut adalah izin tambahan yang diperlukan oleh konsol tersebut. Selain itu, tindakan `s3:PutObjectAcl` dan `s3:GetObjectAcl` diperlukan untuk dapat menyalin,

memotong, dan menempel objek di konsol. Untuk panduan contoh yang memberikan izin kepada pengguna dan mengujinya dengan menggunakan konsol, lihat Mengontrol akses ke bucket dengan kebijakan pengguna.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action": "s3:ListAllMyBuckets",
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":["s3:ListBucket","s3:GetBucketLocation"],
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1"
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:PutObjectAcl",
            "s3:GetObject",
            "s3:GetObjectAcl",
            "s3:DeleteObject"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/*"
      }
   ]
}
```

Mengizinkan setiap akses pengguna IAM ke folder dalam bucket

Dalam contoh ini, Anda ingin dua pengguna IAM, Mary dan Carlos, memiliki akses ke bucket Anda*amzn-s3-demo-bucket1*, sehingga mereka dapat menambahkan, memperbarui, dan menghapus objek. Namun, Anda ingin membatasi setiap akses pengguna ke satu prefiks (folder) dalam bucket. Anda dapat membuat folder dengan nama yang cocok dengan nama pengguna mereka.

```
amzn-s3-demo-bucket1
   Mary/
   Carlos/
```

Untuk memberikan setiap pengguna akses hanya ke foldernya saja, Anda dapat menulis kebijakan untuk setiap pengguna dan melampirkannya secara sendiri-sendiri. Misalnya, Anda dapat melampirkan kebijakan berikut ke pengguna Mary untuk mengizinkan izin Amazon S3 tertentu pada folder *amzn-s3-demo-bucket1*/*Mary*.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/Mary/*"
      }
   ]
}
```

Anda kemudian dapat melampirkan kebijakan serupa ke pengguna Carlos, menentukan folder *Carlos* dalam nilai`Resource`.

Alih-alih melampirkan kebijakan kepada masing-masing pengguna, Anda dapat menulis satu kebijakan yang menggunakan variabel kebijakan dan melampirkan kebijakan tersebut ke grup. Pertama, Anda harus membuat grup dan menambahkan Mary dan Carlos ke grup. Kebijakan contoh berikut memungkinkan serangkaian izin Amazon S3 pada folder *amzn-s3-demo-bucket1*/`${aws:username}`. Ketika kebijakan dievaluasi, variabel `${aws:username}` kebijakan diganti dengan nama pengguna pemohon. Misalnya, jika Mary mengirim permintaan untuk menempatkan objek, maka operasi hanya diperbolehkan jika Mary mengunggah objek tersebut ke *amzn-s3-demo-bucket1*/`Mary` folder.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
```

```
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/${aws:username}/*"
      }
   ]
}
```

> **Note**
>
> Saat menggunakan variabel kebijakan, Anda harus secara jelas menyebutkan versi `2012-10-17` dalam kebijakan tersebut. Versi bawaan bahasa kebijakan IAM, yaitu 2008-10-17, tidak mendukung variabel kebijakan ini.

Jika Anda ingin menguji kebijakan sebelumnya di konsol Amazon S3, maka konsol tersebut memerlukan izin tambahan, seperti yang ditunjukkan dalam kebijakan berikut. Untuk informasi tentang bagaimana konsol menggunakan izin ini, lihat Mengontrol akses ke bucket dengan kebijakan pengguna.

```
{
 "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AllowGroupToSeeBucketListInTheConsole",
      "Action": [
       "s3:ListAllMyBuckets",
       "s3:GetBucketLocation"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::*"
    },
    {
      "Sid": "AllowRootLevelListingOfTheBucket",
      "Action": "s3:ListBucket",
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1",
      "Condition":{
            "StringEquals":{
                   "s3:prefix":[""], "s3:delimiter":["/"]
```

```
                        }
                }
    },
    {
      "Sid": "AllowListBucketOfASpecificUserPrefix",
      "Action": "s3:ListBucket",
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1",
      "Condition":{  "StringLike":{"s3:prefix":["${aws:username}/*"] }
       }
    },
      {
     "Sid": "AllowUserSpecificActionsOnlyInTheSpecificUserPrefix",
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/${aws:username}/*"
      }
  ]
}
```

> **Note**
>
> Dalam versi kebijakan 2012-10-17, variabel kebijakan dimulai dengan $. Perubahan sintaks ini berpotensi membuat konflik jika kunci objek Anda (nama objek) termasuk $.
> Untuk menghindari konflik ini, tentukan $ karakter dengan menggunakan`${$}`. Misalnya, untuk menyertakan kunci objek `my$file` dalam kebijakan, tentukan sebagai`my${$}file`.

Meskipun nama pengguna IAM adalah pengidentifikasi yang dapat dibaca manusia dan mudah dikenali, tetapi nama-nama tersebut tidak harus bersifat unik secara global. Misalnya, jika pengguna Carlos meninggalkan organisasi dan Carlos lainnya bergabung, maka Carlos baru dapat mengakses informasi Carlos lama.

Alih-alih menggunakan nama pengguna, Anda bisa membuat folder berdasarkan pengguna IAM. IDs Setiap ID pengguna IAM adalah unik. Dalam hal ini, Anda harus mengubah kebijakan sebelumnya

untuk menggunakan variabel kebijakan`${aws:userid}`. Untuk informasi lebih lanjut tentang pengidentifikasi pengguna, lihat Pengidentifikasi IAM dalam Panduan Pengguna IAM.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/home/${aws:userid}/*"
      }
   ]
}
```

Mengizinkan pengguna non-IAM (pengguna aplikasi seluler) mengakses folder di dalam bucket

Misalkan Anda ingin mengembangkan sebuah aplikasi seluler, sebuah permainan yang menyimpan data pengguna dalam bucket S3. Untuk setiap pengguna aplikasi, Anda ingin membuat folder dalam bucket Anda. Anda juga ingin membatasi akses setiap pengguna ke folder mereka sendiri. Tetapi Anda tidak dapat membuat folder sebelum seseorang mengunduh aplikasi Anda dan mulai memainkan game, karena Anda tidak memiliki ID pengguna mereka.

Dalam hal ini, Anda dapat meminta pengguna untuk masuk ke aplikasi Anda dengan menggunakan penyedia identitas publik seperti Login dengan Amazon, Facebook, atau Google. Setelah pengguna masuk ke aplikasi Anda melalui salah satu penyedia layanan ini, mereka memiliki ID pengguna yang dapat Anda gunakan untuk membuat folder kustom pengguna saat runtime.

Anda kemudian dapat menggunakan federasi identitas web AWS Security Token Service untuk mengintegrasikan informasi dari penyedia identitas dengan aplikasi Anda dan untuk mendapatkan kredensi keamanan sementara untuk setiap pengguna. Anda kemudian dapat membuat kebijakan IAM yang memungkinkan aplikasi mengakses bucket Anda dan melakukan operasi seperti membuat folder kustom pengguna dan mengunggah data. Untuk informasi lebih lanjut tentang federasi identitas web, lihat Tentang Federasi Identitas Web dalam Panduan Pengguna IAM.

Mengizinkan grup memiliki folder bersama di Amazon S3

Melampirkan kebijakan berikut ke grup akan memberikan kepada semua orang di grup akses ke folder berikut di Amazon S3: *amzn-s3-demo-bucket1*/share/marketing. Anggota grup diizinkan untuk hanya mengakses izin Amazon S3 tertentu saja yang ditampilkan dalam kebijakan dan hanya untuk objek di folder tertentu saja.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/share/marketing/*"
      }
   ]
}
```

Mengizinkan semua pengguna Anda membaca objek di sebagian bucket

Dalam contoh ini, Anda membuat grup dengan nama *AllUsers*, yang berisi semua pengguna IAM yang dimiliki oleh Akun AWS. Anda kemudian melampirkan kebijakan yang memberikan grup akses ke GetObject dan GetObjectVersion, tetapi hanya untuk objek yang ada di folder *amzn-s3-demo-bucket1*/readonly saja.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObject",
            "s3:GetObjectVersion"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/readonly/*"
      }
```

```
    ]
}
```

Mengizinkan mitra untuk menempatkan file ke dalam bagian tertentu bucket

Dalam contoh ini, Anda membuat grup yang disebut *AnyCompany* yang mewakili perusahaan mitra. Anda membuat pengguna IAM untuk orang atau aplikasi tertentu di perusahaan mitra yang memerlukan akses, lalu Anda menempatkan pengguna tersebut dalam grup.

Anda kemudian melampirkan kebijakan yang memberikan `PutObject` akses kepada grup ke folder berikut dalam bucket:

amzn-s3-demo-bucket1/uploads/anycompany

Anda ingin mencegah *AnyCompany* grup melakukan hal lain dengan bucket, sehingga Anda menambahkan pernyataan yang secara jelas menolak izin untuk tindakan Amazon S3 kecuali pada sumber daya `PutObject` Amazon S3 di. Akun AWS

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"s3:PutObject",
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/uploads/anycompany/*"
      },
      {
         "Effect":"Deny",
         "Action":"s3:*",
         "NotResource":"arn:aws:s3:::amzn-s3-demo-bucket1/uploads/anycompany/*"
      }
   ]
}
```

Membatasi akses ke bucket Amazon S3 dalam bucket tertentu Akun AWS

Jika Anda ingin memastikan bahwa kepala sekolah Amazon S3 Anda hanya mengakses sumber daya yang ada di dalam tepercaya Akun AWS, Anda dapat membatasi akses. Misalnya, kebijakan IAM berbasis identitas ini menggunakan `Deny` efek untuk memblokir akses ke tindakan Amazon S3, kecuali sumber daya Amazon S3 yang sedang diakses ada di akun *222222222222*. Untuk mencegah prinsipal IAM dalam mengakses objek Amazon S3 di luar akun, lampirkan kebijakan IAM berikut: Akun AWS

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyS3AccessOutsideMyBoundary",
      "Effect": "Deny",
      "Action": [
        "s3:*"
      ],
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:ResourceAccount": [
            "222222222222"
          ]
        }
      }
    }
  ]
}
```

> **Note**
>
> Kebijakan ini tidak menggantikan kontrol akses IAM Anda yang ada, karena tidak memberikan akses apa pun. Sebagai gantinya, kebijakan ini bertindak sebagai pagar pembatas tambahan untuk izin IAM Anda yang lain, terlepas dari izin yang diberikan melalui kebijakan IAM lainnya.

Pastikan untuk mengganti ID akun *222222222222* di polis dengan ID Anda sendiri Akun AWS. Untuk menerapkan kebijakan ke beberapa akun sambil tetap mempertahankan batasan ini, ganti ID akun dengan kunci `aws:PrincipalAccount` kondisi. Kondisi ini mensyaratkan bahwa pengguna utama dan sumber daya harus berada dalam akun yang sama.

Membatasi akses ke bucket Amazon S3 dalam unit organisasi Anda

Jika Anda memiliki unit organisasi (OU) yang disiapkan AWS Organizations, Anda mungkin ingin membatasi akses bucket Amazon S3 ke bagian tertentu dari organisasi Anda. Dalam contoh ini, kami akan menggunakan `aws:ResourceOrgPaths` kunci untuk membatasi akses bucket Amazon S3

ke OU di organisasi Anda. Untuk contoh ini, ID OU adalah*ou-acroot-exampleou*. Pastikan untuk mengganti nilai ini dalam kebijakan Anda sendiri dengan OU Anda sendiri IDs.

```
{
   "Version": "2012-10-17",
   "Statement": [
     {
       "Sid": "AllowS3AccessOutsideMyBoundary",
       "Effect": "Allow",
       "Action": [
         "s3:*"
       ],
       "Resource": "*",
       "Condition": {
         "ForAllValues:StringNotLike": {
           "aws:ResourceOrgPaths": [
             "o-acorg/r-acroot/ou-acroot-exampleou/"
           ]
         }
       }
     }
   ]
 }
```

> **Note**
>
> Kebijakan ini tidak memberikan akses apa pun. Sebagai gantinya, kebijakan ini bertindak sebagai penghalang untuk izin IAM Anda yang lain, mencegah pengguna utama mengakses objek Amazon S3 di luar batas yang ditentukan OU.

Kebijakan ini menolak akses ke tindakan Amazon S3 kecuali objek Amazon S3 yang sedang diakses ada di OU di *ou-acroot-exampleou* organisasi Anda. Kondisi kebijakan IAM mengharuskan`aws:ResourceOrgPaths`, kunci kondisi multivalued, untuk memuat salah satu jalur OU yang terdaftar. Kebijakan menggunakan `ForAllValues:StringNotLike` operator untuk membandingkan nilai yang tercantum OUs tanpa `aws:ResourceOrgPaths` pencocokan peka huruf besar/kecil.

Membatasi akses ke bucket Amazon S3 di organisasi Anda

Untuk membatasi akses ke objek Amazon S3 dalam organisasi Anda, lampirkan kebijakan IAM ke root organisasi, terapkan ke semua akun di organisasi Anda. Untuk mewajibkan pengguna utama IAM Anda mengikuti aturan ini, gunakan kebijakan kontrol layanan (SCP). Jika Anda memilih untuk menggunakan SCP, pastikan untuk menguji SCP secara menyeluruh sebelum melampirkan kebijakan ke root organisasi.

Dalam contoh kebijakan berikut, akses ditolak ke tindakan Amazon S3 kecuali objek Amazon S3 yang diakses berada di organisasi yang sama dengan pengguna utama IAM yang mengaksesnya:

```
{
   "Version": "2012-10-17",
   "Statement": [
     {
       "Sid": "DenyS3AccessOutsideMyBoundary",
       "Effect": "Deny",
       "Action": [
         "s3:*"
       ],
       "Resource": "arn:aws:s3:::*/*",
       "Condition": {
         "StringNotEquals": {
           "aws:ResourceOrgID": "${aws:PrincipalOrgID}"
         }
       }
     }
   ]
 }
```

> **Note**
>
> Kebijakan ini tidak memberikan akses apa pun. Sebagai gantinya, kebijakan ini bertindak sebagai penghalang untuk izin IAM Anda yang lain, mencegah pengguna utama mengakses objek Amazon S3 apa pun di luar organisasi Anda. Kebijakan ini juga berlaku untuk sumber daya Amazon S3 yang dibuat setelah kebijakan diberlakukan.

Kondisi kebijakan IAM dalam contoh ini mengharuskan `aws:ResourceOrgID` dan `aws:PrincipalOrgID` agar sama satu dengan yang lainnya. Dengan persyaratan ini, pengguna

utama yang membuat permintaan dan sumber daya yang diakses harus berada di organisasi yang sama.

Memberikan izin untuk mengambil PublicAccessBlock konfigurasi untuk Akun AWS

Contoh kebijakan berbasis identitas berikut memberikan `s3:GetAccountPublicAccessBlock` izin kepada pengguna. Untuk izin ini, Anda mengatur nilai `Resource` ke `"*"`. Untuk informasi tentang sumber daya ARNs, lihatSumber daya kebijakan untuk Amazon S3.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"statement1",
         "Effect":"Allow",
         "Action":[
            "s3:GetAccountPublicAccessBlock"
         ],
         "Resource":[
            "*"
         ]
       }
    ]
}
```

Membatasi pembuatan bucket ke satu Wilayah

Misalkan Akun AWS administrator ingin memberikan izin kepada penggunanya (Dave) untuk membuat ember di Wilayah Amerika Selatan (São Paulo) saja. Administrator akun dapat melampirkan kebijakan pengguna berikut ini yang memberikan izin `s3:CreateBucket` dengan syarat seperti yang ditunjukkan. Pasangan nilai kunci dalam pemblokiran `Condition` menentukan kunci `s3:LocationConstraint` dan Wilayah `sa-east-1` sebagai nilainya.

> **Note**
>
> Dalam contoh ini, pemilik bucket memberikan izin kepada salah satu penggunanya, sehingga kebijakan bucket atau kebijakan pengguna dapat digunakan. Contoh ini menunjukkan kebijakan pengguna.

Untuk daftar Wilayah Amazon S3, lihat Wilayah dan Titik Akhir di Referensi Umum AWS.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"statement1",
         "Effect":"Allow",
         "Action": "s3:CreateBucket",
         "Resource": "arn:aws:s3:::*",
         "Condition": {
             "StringLike": {
                 "s3:LocationConstraint": "sa-east-1"
             }
         }
       }
    ]
}
```

Menambahkan penolakan eksplisit

Kebijakan sebelumnya membatasi pengguna untuk membuat bucket di Wilayah lain kecuali `sa-east-1`. Namun, beberapa kebijakan lain mungkin memberikan izin kepada pengguna ini untuk membuat bucket di Wilayah lain. Misalnya, jika pengguna merupakan bagian dari sebuah kelompok, dan kelompok tersebut mungkin memiliki kebijakan yang melekat padanya yang memberikan izin kepada semua pengguna dalam kelompok tersebut untuk membuat bucket di Wilayah lain. Untuk memastikan bahwa pengguna tidak mendapatkan izin untuk membuat bucket di Wilayah lain, Anda dapat menambahkan pernyataan penolakan eksplisit dalam kebijakan di atas.

Pernyataan `Deny` menggunakan kondisi `StringNotLike`. Artinya, permintaan buat bucket ditolak jika kendala lokasi tidak `sa-east-1`. Penolakan eksplisit tidak memungkinkan pengguna membuat bucket di Wilayah lain mana pun, apa pun izin lain yang didapat pengguna. Kebijakan berikut mencakup pernyataan penolakan eksplisit.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"statement1",
         "Effect":"Allow",
         "Action": "s3:CreateBucket",
         "Resource": "arn:aws:s3:::*",
         "Condition": {
```

```
            "StringLike": {
                "s3:LocationConstraint": "sa-east-1"
            }
         }
       },
      {
         "Sid":"statement2",
         "Effect":"Deny",
         "Action": "s3:CreateBucket",
         "Resource": "arn:aws:s3:::*",
         "Condition": {
            "StringNotLike": {
                "s3:LocationConstraint": "sa-east-1"
            }
         }
       }
    ]
}
```

Uji kebijakan dengan AWS CLI

Anda dapat menguji kebijakan menggunakan `create-bucket` AWS CLI perintah berikut. Contoh ini menggunakan file `bucketconfig.txt` untuk menentukan batasan lokasi. Perhatikan jalur Windows file. Anda perlu memperbarui nama bucket dan alurnya sesuai kebutuhan. Anda harus memberikan kredensial pengguna dengan menggunakan parameter `--profile`. Untuk informasi selengkapnya tentang pengaturan dan penggunaan AWS CLI, lihat Mengembangkan dengan Amazon S3 menggunakan AWS CLI di Referensi API Amazon S3.

```
aws s3api create-bucket --bucket examplebucket --profile AccountADave --create-bucket-
configuration file://c:/Users/someUser/bucketconfig.txt
```

File `bucketconfig.txt` menentukan konfigurasi sebagai berikut.

```
{"LocationConstraint": "sa-east-1"}
```

Panduan yang menggunakan kebijakan untuk mengelola akses ke sumber daya Amazon S3

Topik ini menyediakan contoh penelusuran pendahuluan berikut ini untuk memberikan akses ke sumber daya Amazon S3. Contoh-contoh ini menggunakan AWS Management Console untuk membuat sumber daya (bucket, objek, pengguna) dan memberi mereka izin. Contoh-contoh ini kemudian menunjukkan kepada Anda cara memverifikasi izin menggunakan alat baris perintah, sehingga Anda tidak perlu menulis kode apa pun. Kami menyediakan perintah menggunakan kedua AWS Command Line Interface (AWS CLI) dan AWS Tools for Windows PowerShell.

- Contoh 1: Pemilik bucket yang memberikan izin bucket kepada penggunanya

  Secara bawaan, pengguna IAM yang Anda buat di akun Anda tidak memiliki izin. Dalam latihan ini, Anda memberikan izin kepada pengguna untuk melakukan operasi bucket dan objek.

- Contoh 2: Pemilik bucket yang memberikan izin bucket lintas akun

  Dalam latihan ini, pemilik bucket, Akun A, memberikan izin lintas akun ke Akun AWS lain, Akun B. Akun B kemudian mendelegasikan izin tersebut kepada pengguna dalam akunnya.

- Mengelola izin objek saat pemilik objek dan bucket tidak sama

  Skenario contoh dalam kasus ini adalah tentang pemilik bucket yang memberikan izin objek kepada orang lain, tetapi tidak semua objek dalam bucket dimiliki oleh pemilik bucket. Apa izin yang diperlukan pemilik bucket, dan bagaimana izin tersebut dapat didelegasikan?

  Akun AWS Yang membuat ember disebut pemilik ember. Pemilik dapat memberikan Akun AWS izin lain untuk mengunggah objek, dan Akun AWS yang membuat objek memilikinya. Pemilik bucket tidak memiliki izin pada objek yang dibuat oleh Akun AWS lain. Jika pemilik bucket menulis kebijakan bucket yang memberikan akses ke objek, kebijakan tersebut tidak berlaku untuk objek yang dimiliki oleh akun lain.

  Dalam hal ini, pemilik objek harus terlebih dahulu memberikan izin kepada pemilik bucket dengan menggunakan ACL objek. Pemilik bucket kemudian dapat mendelegasikan izin objek tersebut kepada orang lain, kepada pengguna di akunnya sendiri, atau ke akun lain Akun AWS, seperti yang diilustrasikan oleh contoh berikut.

  - Contoh 3: Pemilik bucket yang memberikan izin kepada penggunanya ke objek yang bukan miliknya

Dalam latihan ini, pemilik bucket terlebih dahulu mendapatkan izin dari pemilik objek. Pemilik bucket kemudian mendelegasikan izin tersebut kepada pengguna dalam akunnya sendiri.

- Contoh 4 - Pemilik bucket memberikan izin lintas akun ke objek yang tidak dimilikinya

  Setelah menerima izin dari pemilik objek, pemilik bucket tidak dapat mendelegasikan izin ke yang lain Akun AWS karena delegasi lintas akun tidak didukung (lihat). Delegasi izin Sebagai gantinya, pemilik bucket dapat membuat peran IAM dengan izin untuk melakukan operasi tertentu (seperti get object) dan mengizinkan orang lain Akun AWS untuk mengambil peran tersebut. Siapa pun yang akan mengambil peran tersebut kemudian dapat mengakses objeknya. Contoh ini menunjukkan bagaimana pemilik bucket dapat menggunakan peran IAM untuk memungkinkan delegasi lintas akun ini.

Sebelum Anda mencoba contoh penelusuran

Contoh-contoh ini menggunakan AWS Management Console untuk membuat sumber daya dan memberikan izin. Untuk menguji izin, contoh menggunakan alat baris perintah, dan AWS CLI AWS Tools for Windows PowerShell, jadi Anda tidak perlu menulis kode apa pun. Untuk menguji izin, Anda harus menyiapkan salah satu alat ini. Untuk informasi selengkapnya, lihat Menyiapkan alat untuk penelusuran.

Selain itu, saat membuat sumber daya, contoh-contoh ini tidak menggunakan kredenal pengguna root dari file. Akun AWS Sebaliknya, Anda membuat pengguna administrator di akun ini untuk melakukan tugas-tugas tersebut.

Tentang menggunakan pengguna administrator untuk membuat sumber daya dan memberikan izin

AWS Identity and Access Management (IAM) merekomendasikan untuk tidak menggunakan kredensial pengguna root perangkat Akun AWS untuk mengajukan permintaan. Sebagai gantinya, buat pengguna IAM atau peran, berikan mereka akses penuh, kemudian gunakan kredensialnya untuk mengajukan permintaan. Kami menyebut ini sebagai pengguna administratif atau peran. Untuk informasi lebih lanjut, buka Pengguna root akun AWS kredensial dan identitas IAM di Referensi Umum AWS dan Praktik Terbaik IAM di Panduan Pengguna IAM.

Semua panduan contoh dalam bagian ini menggunakan kredensial pengguna administrator. Jika Anda belum membuat pengguna administrator untuk Anda Akun AWS, topik menunjukkan caranya.

Untuk masuk ke AWS Management Console menggunakan kredensil pengguna, Anda harus menggunakan URL Masuk pengguna IAM. Konsol IAM menyediakan URL ini untuk Anda Akun AWS. Topik-topik ini juga akan menunjukkan cara mendapatkan URL.

Menyiapkan alat untuk penelusuran

Contoh pengantar (lihatPanduan yang menggunakan kebijakan untuk mengelola akses ke sumber daya Amazon S3) menggunakan AWS Management Console untuk membuat sumber daya dan memberikan izin. Untuk menguji izin, contoh menggunakan alat baris perintah, AWS Command Line Interface (AWS CLI) dan AWS Tools for Windows PowerShell, jadi Anda tidak perlu menulis kode apa pun. Untuk menguji izin, Anda harus menyiapkan salah satu alat ini.

Untuk mengatur AWS CLI

1. Unduh dan konfigurasikan AWS CLI. Untuk instruksi, lihat topik berikut di AWS Command Line Interface Panduan Pengguna:

   Instal atau perbarui ke versi terbaru AWS Command Line Interface

   Memulai dengan AWS Command Line Interface
2. Mengatur profil bawaan.

   Anda menyimpan kredensi pengguna di file AWS CLI konfigurasi. Buat profil default di file konfigurasi menggunakan Akun AWS kredensil Anda. Untuk petunjuk tentang menemukan dan mengedit file AWS CLI konfigurasi Anda, lihat Konfigurasi dan pengaturan file kredensi.

   ```
   [default]
   aws_access_key_id = access key ID
   aws_secret_access_key = secret access key
   region = us-west-2
   ```
3. Verifikasi pengaturan dengan memasukkan perintah berikut pada prompt perintah. Kedua perintah ini tidak memberikan kredensial secara jelas, sehingga kredensial profil bawaan digunakan.

   - Coba `help` perintahnya.

     ```
     aws help
     ```
   - Untuk mendapatkan daftar ember pada akun yang dikonfigurasi, gunakan `aws s3 ls` perintah.

```
aws s3 ls
```

Saat Anda melalui penelusuran, Anda akan membuat pengguna, dan Anda akan menyimpan kredensi pengguna dalam file konfigurasi dengan membuat profil, seperti yang ditunjukkan contoh berikut. Profil ini memiliki nama `AccountAadmin` dan`AccountBadmin`.

```
[profile AccountAadmin]
aws_access_key_id = User AccountAadmin access key ID
aws_secret_access_key = User AccountAadmin secret access key
region = us-west-2

[profile AccountBadmin]
aws_access_key_id = Account B access key ID
aws_secret_access_key = Account B secret access key
region = us-east-1
```

Untuk menjalankan perintah menggunakan kredensial pengguna ini, Anda perlu menambahkan parameter `--profile` yang menentukan nama profil. AWS CLI Perintah berikut mengambil daftar objek di *`examplebucket`* dan menentukan profil. `AccountBadmin`

```
aws s3 ls s3://examplebucket --profile AccountBadmin
```

Atau, Anda dapat mengonfigurasi satu set kredensial pengguna sebagai profil bawaan dengan mengubah variabel lingkungan `AWS_DEFAULT_PROFILE` dari command prompt. Setelah Anda melakukan ini, setiap kali Anda melakukan AWS CLI perintah tanpa `--profile` parameter, AWS CLI menggunakan profil yang Anda atur dalam variabel lingkungan sebagai profil default.

```
$ export AWS_DEFAULT_PROFILE=AccountAadmin
```

Untuk mengatur AWS Tools for Windows PowerShell

1. Unduh dan konfigurasikan AWS Tools for Windows PowerShell. Untuk instruksi, buka Menginstal AWS Tools for Windows PowerShell di Panduan Alat AWS untuk PowerShell Pengguna.

> **Note**
>
> Untuk memuat AWS Tools for Windows PowerShell modul, Anda harus mengaktifkan eksekusi PowerShell skrip. Untuk informasi selengkapnya, lihat Mengaktifkan Eksekusi Skrip di Panduan Alat AWS untuk PowerShell Pengguna.

2. Untuk penelusuran ini, Anda menentukan AWS kredensi per sesi menggunakan perintah. `Set-AWSCredentials` Perintah menyimpan kredensial ke penyimpanan persisten (parameter `-StoreAs`).

```
Set-AWSCredentials -AccessKey AccessKeyID -SecretKey SecretAccessKey -
storeas string
```

3. Verifikasi pengaturan.

   - Untuk mengambil daftar perintah yang tersedia yang dapat Anda gunakan untuk operasi Amazon S3, jalankan `Get-Command` perintah.

     ```
     Get-Command -module awspowershell -noun s3* -StoredCredentials string
     ```

   - Untuk mengambil daftar objek dalam ember, jalankan `Get-S3Object` perintah.

     ```
     Get-S3Object -BucketName bucketname -StoredCredentials string
     ```

Untuk daftar perintah, lihat AWS Alat untuk Referensi PowerShell Cmdlet.

Sekarang Anda siap untuk mencoba penelusuran. Ikuti tautan yang disediakan di awal setiap bagian.

Contoh 1: Pemilik bucket yang memberikan izin bucket kepada penggunanya

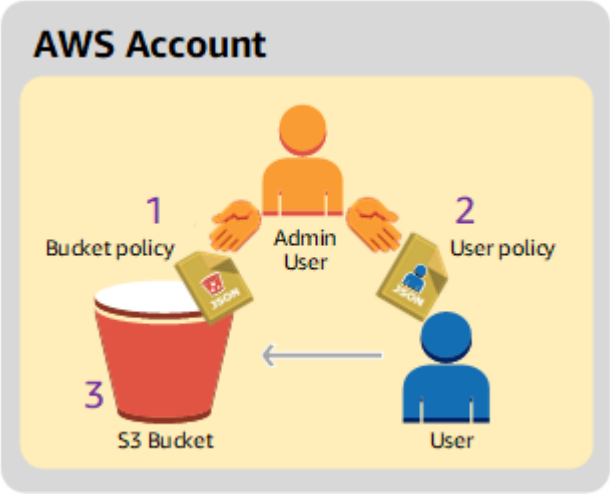
> **Important**
>
> Memberikan izin untuk peran IAM adalah praktik yang lebih baik daripada memberikan izin kepada pengguna individual.Untuk informasi selengkapnya tentang cara memberikan izin ke peran IAM, lihat. Memahami izin lintas akun dan menggunakan peran IAM

Topik

- Bersiap untuk panduan
- Langkah 1: Buat sumber daya di Akun A dan berikan izin
- Langkah 2: Menguji izin

Dalam panduan ini, a Akun AWS memiliki ember, dan akun menyertakan pengguna IAM Secara default, pengguna tidak memiliki izin. Agar pengguna dapat melakukan tugas apa pun, akun induk harus memberikan izin kepada mereka. Pemilik bucket dan akun induk sama. Oleh karena itu, untuk memberikan izin pengguna di bucket, Akun AWS dapat menggunakan kebijakan bucket, kebijakan pengguna, atau keduanya. Pemilik akun akan memberikan beberapa izin menggunakan kebijakan bucket dan beberapa izin lainnya menggunakan kebijakan pengguna.

Langkah-langkah berikut merangkum panduannya:



1. Administrator akun membuat kebijakan bucket yang memberikan serangkaian izin kepada pengguna.

2. Administrator akun melampirkan kebijakan pengguna pada pengguna yang memberikan izin tambahan.
3. Pengguna kemudian mencoba izin yang diberikan baik melalui kebijakan bucket maupun kebijakan pengguna.

Untuk contoh ini, Anda akan membutuhkan Akun AWS. Alih-alih menggunakan kredensial pengguna root akun, Anda akan membuat pengguna administrator (lihatTentang menggunakan pengguna administrator untuk membuat sumber daya dan memberikan izin). Kami merujuk ke Akun AWS dan pengguna administrator seperti yang ditunjukkan pada tabel berikut.

account-id	Akun disebut sebagai	Pengguna administrator pada akun
1111-1111-1111	Akun A	AccountAadmin

> **Note**
>
> Pengguna administrator dalam contoh ini adalah AccountAadmin, yang mengacu pada Akun A, dan bukan AccountAdmin.

Semua tugas membuat pengguna dan memberikan izin dilakukan di AWS Management Console. Untuk memverifikasi izin, penelusuran menggunakan alat baris perintah, AWS Command Line Interface (AWS CLI) dan AWS Tools for Windows PowerShell, jadi Anda tidak perlu menulis kode apa pun.

Bersiap untuk panduan

1. Pastikan Anda memiliki Akun AWS dan memiliki pengguna dengan hak administrator.

   a. Mendaftar untuk Akun AWS, jika diperlukan. Kami menyebut akun ini sebagai Akun A.

      i. Buka https://aws.amazon.com/s3 dan pilih Buat akun AWS .

      ii. Ikuti petunjuk di layar.

      AWS akan memberi tahu Anda melalui email ketika akun Anda aktif dan tersedia untuk Anda gunakan.

b. Di Akun A, buat pengguna administrator**`AccountAadmin`**. Menggunakan kredensial Akun A, masuklah ke Konsol IAM dan lakukan hal-hal berikut ini:

   i. Buat pengguna **`AccountAadmin`** dan catat kredensil keamanan pengguna.

      Untuk petunjuk, lihat Membuat pengguna IAM di Panduan Pengguna IAM Anda Akun AWS.

   ii. Berikan hak administrator AccountAadmindengan melampirkan kebijakan pengguna yang memberikan akses penuh.

      Untuk instruksi, lihat Mengelola kebijakan IAM dalam Panduan Pengguna IAM.

   iii. Perhatikan URL Masuk pengguna IAM untuk. AccountAadmin Anda perlu menggunakan URL ini saat masuk ke AWS Management Console. Untuk informasi selengkapnya tentang tempat menemukan URL masuk, lihat Masuk ke AWS Management Console sebagai pengguna IAM di Panduan Pengguna IAM. Catat URL untuk setiap akun.

2. Siapkan salah satu AWS CLI atau AWS Tools for Windows PowerShell. Pastikan Anda menyimpan kredensi pengguna administrator sebagai berikut:

   - Jika menggunakan AWS CLI, buat profil,`AccountAadmin`, di file konfigurasi.
   - Jika menggunakan AWS Tools for Windows PowerShell, pastikan Anda menyimpan kredensi untuk sesi sebagai. `AccountAadmin`

   Untuk petunjuk, lihat Menyiapkan alat untuk penelusuran.

Langkah 1: Buat sumber daya di Akun A dan berikan izin

Dengan menggunakan kredensil pengguna `AccountAadmin` di Akun A, dan URL login pengguna IAM khusus, masuk ke akun AWS Management Console dan lakukan hal berikut:

1. Buat sumber daya bucket dan pengguna IAM

   a. Pada konsol Amazon S3, buat bucket. Perhatikan Wilayah AWS di mana Anda membuat ember. Untuk petunjuk, lihat Membuat ember tujuan umum.

   b. Di Konsol IAM, lakukan hal berikut:

      i. Buat pengguna bernama Dave.

Untuk step-by-step petunjuk, lihat Membuat pengguna IAM (konsol) di Panduan Pengguna IAM.

ii. Perhatikan `UserDave` kredensialnya.

iii. Perhatikan Nama Sumber Daya Amazon (ARN) untuk pengguna Dave. Di Konsol IAM, pilih pengguna, dan tab Ringkasan menyediakan ARN pengguna.

2. Berikan izin.

Karena pemilik bucket dan akun induk yang dimiliki pengguna adalah sama, mereka Akun AWS dapat memberikan izin pengguna menggunakan kebijakan bucket, kebijakan pengguna, atau keduanya. Dalam contoh ini, Anda melakukan keduanya. Jika objek juga dimiliki oleh akun yang sama, maka pemilik bucket dapat memberikan izin objek dalam kebijakan bucket (atau kebijakan IAM).

a. Di konsol Amazon S3, lampirkan kebijakan bucket berikut ke. *awsexamplebucket1*

Kebijakan ini memiliki dua pernyataan.

- Pernyataan pertama memberikan izin operasi bucket `s3:GetBucketLocation` dan `s3:ListBucket` kepada Dave.
- Pernyataan kedua memberikan izin kepada `s3:GetObject`. Karena Akun A juga memiliki objek, administrator akun dapat memberikan izin kepada `s3:GetObject`.

Pada pernyataan `Principal`, Dave diidentifikasi oleh pengguna ARN-nya. Untuk informasi lebih lanjut tentang elemen-elemen kebijakan, lihat Kebijakan dan izin di Amazon S3.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "statement1",
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::AccountA-ID:user/Dave"
         },
         "Action": [
            "s3:GetBucketLocation",
            "s3:ListBucket"
         ],
```

```
         "Resource": [
            "arn:aws:s3:::awsexamplebucket1"
         ]
      },
      {
         "Sid": "statement2",
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::AccountA-ID:user/Dave"
         },
         "Action": [
             "s3:GetObject"
         ],
         "Resource": [
            "arn:aws:s3:::awsexamplebucket1/*"
         ]
      }
   ]
}
```

b. Buat kebijakan yang selaras untuk pengguna Dave, dengan menggunakan kebijakan berikut. Kebijakan tersebut memberikan izin `s3:PutObject` kepada Dave. Anda perlu memperbarui kebijakan dengan memberikan nama bucket Anda.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "PermissionForObjectOperations",
         "Effect": "Allow",
         "Action": [
            "s3:PutObject"
         ],
         "Resource": [
            "arn:aws:s3:::awsexamplebucket1/*"
         ]
      }
   ]
}
```

Untuk petunjuk, lihat Mengelola IAMpolicies dalam Panduan Pengguna IAM. Perlu diperhatikan bahwa Anda perlu masuk ke konsol dengan menggunakan kredensial Akun A.

Langkah 2: Menguji izin

Dengan menggunakan kredensial Dave, verifikasikan bahwa izin tersebut berfungsi. Anda dapat menggunakan salah satu dari dua prosedur berikut.

Izin uji menggunakan AWS CLI

1. Perbarui file AWS CLI konfigurasi dengan menambahkan `UserDaveAccountA` profil berikut. Untuk informasi selengkapnya, lihat Menyiapkan alat untuk penelusuran.

   ```
   [profile UserDaveAccountA]
   aws_access_key_id = access-key
   aws_secret_access_key = secret-access-key
   region = us-east-1
   ```

2. Verifikasi bahwa Dave dapat melakukan operasi sebagaimana yang diberikan dalam kebijakan pengguna. Unggah objek sampel menggunakan AWS CLI `put-object` perintah berikut.

   Parameter `--body` dalam perintah mengidentifikasi file sumber yang akan diunggah. Misalnya, jika file berada di root drive C: pada Windows mesin, Anda tentukan`c:\HappyFace.jpg`. Parameter `--key` menyediakan nama kunci untuk objek.

   ```
   aws s3api put-object --bucket awsexamplebucket1 --key HappyFace.jpg --
   body HappyFace.jpg --profile UserDaveAccountA
   ```

   Jalankan AWS CLI perintah berikut untuk mendapatkan objek.

   ```
   aws s3api get-object --bucket awsexamplebucket1 --key HappyFace.jpg OutputFile.jpg
    --profile UserDaveAccountA
   ```

Izin uji menggunakan AWS Tools for Windows PowerShell

1. Simpan kredensial Dave sebagai `AccountADave`. Anda kemudian menggunakan kredensil ini ke PUT dan GET objek.

   ```
   set-awscredentials -AccessKey AccessKeyID -SecretKey SecretAccessKey -storeas
    AccountADave
   ```

2. Unggah objek sampel menggunakan AWS Tools for Windows PowerShell `Write-S3Object` perintah menggunakan kredensi tersimpan pengguna Dave.

```
Write-S3Object -bucketname awsexamplebucket1 -key HappyFace.jpg -file HappyFace.jpg
 -StoredCredentials AccountADave
```

Unduh objek yang telah diunggah sebelumnya.

```
Read-S3Object -bucketname awsexamplebucket1 -key HappyFace.jpg -file Output.jpg -
StoredCredentials AccountADave
```

Contoh 2: Pemilik bucket yang memberikan izin bucket lintas akun

> **Important**
>
> Memberikan izin untuk peran IAM adalah praktik yang lebih baik daripada memberikan izin kepada pengguna individu. Untuk mempelajari cara melakukannya, lihat Memahami izin lintas akun dan menggunakan peran IAM.

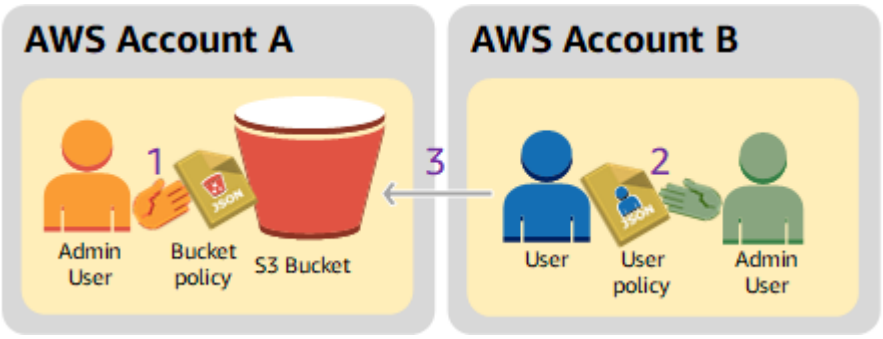
Topik

- Bersiap untuk panduan
- Langkah 1: Melakukan tugas Akun A
- Langkah 2: Melakukan tugas Akun B
- Langkah 3: (Opsional) Coba penolakan eksplisit
- Langkah 4: Membersihkan

Sebuah Akun AWS—misalnya, Akun a—dapat memberikan izin lain Akun AWS, Akun B, untuk mengakses sumber dayanya seperti ember dan objek. Akun B kemudian dapat mendelegasikan izin tersebut kepada pengguna dalam akunnya. Dalam skenario contoh ini, pemilik bucket memberikan izin lintas akun kepada akun lain untuk melakukan operasi bucket tertentu.

> **Note**
>
> Akun A juga dapat secara langsung memberikan izin kepada pengguna di Akun B dengan menggunakan kebijakan bucket. Namun, pengguna masih memerlukan izin dari akun induk, Akun B, tempat pengguna berada, bahkan jika Akun B tidak memiliki izin dari Akun A. Selama pengguna memiliki izin dari pemilik sumber daya dan akun induk, pengguna akan dapat mengakses sumber daya.

Berikut ini adalah ringkasan langkah-langkah panduan:



1. Pengguna administrator Akun A melampirkan kebijakan bucket yang memberikan izin lintas akun kepada Akun B untuk melakukan operasi bucket tertentu.

   Perhatikan bahwa pengguna administrator di Akun B akan secara otomatis mewarisi izin tersebut.
2. Pengguna administrator Akun B melampirkan kebijakan pengguna kepada pengguna yang mendelegasikan izin yang diterima pengguna administrator Akun B dari Akun A.
3. Pengguna di Akun B kemudian memverifikasi izin dengan mengakses objek yang ada dalam bucket yang dimiliki oleh Akun A.

Untuk contoh ini, Anda memerlukan dua akun. Tabel berikut menunjukkan cara kami merujuk akun-akun ini dan pengguna administrator di dalamnya. Sesuai dengan pedoman IAM (lihatTentang menggunakan pengguna administrator untuk membuat sumber daya dan memberikan izin), kami tidak menggunakan kredensi pengguna root dalam panduan ini. Sebagai gantinya, Anda membuat pengguna administrator di setiap akun dan menggunakan kredensial tersebut saat membuat sumber daya dan memberikan mereka izin.

Akun AWS ID	Akun disebut sebagai	Pengguna administrator pada akun
1111-1111-1111	Akun A	AccountAadmin
2222-2222-2222	Akun B	AccountBadmin

Semua tugas membuat pengguna dan memberikan izin dilakukan di AWS Management Console. Untuk memverifikasi izin, panduan menggunakan alat baris perintah, (AWS Command Line Interface CLI) dan AWS Tools for Windows PowerShell, jadi Anda tidak perlu menulis kode apa pun.

Bersiap untuk panduan

1. Pastikan Anda memiliki dua Akun AWS dan bahwa setiap akun memiliki satu pengguna administrator seperti yang ditunjukkan pada tabel di bagian sebelumnya.

   a. Mendaftar untuk Akun AWS, jika diperlukan.

   b. Dengan menggunakan kredensial Akun A, masuk ke Konsol IAM untuk membuat pengguna administrator:

      i. Buat pengguna **`AccountAadmin`** dan catat kredensyal keamanan. Untuk instruksi, lihat Membuat Pengguna IAM dalam Akun AWS Anda, dalam Panduan Pengguna IAM.

      ii. Berikan hak administrator AccountAadmindengan melampirkan kebijakan pengguna yang memberikan akses penuh. Untuk instruksi, lihat Bekerja dengan Kebijakan dalam Panduan Pengguna IAM.

   c. Saat Anda berada di konsol IAM, perhatikan URL Masuk pengguna IAM di Dasbor. Semua pengguna di akun tersebut harus menggunakan URL ini saat masuk ke AWS Management Console.

      Untuk informasi lebih lanjut, lihat Cara Pengguna Masuk ke Akun Anda dalam Panduan Pengguna IAM.

   d. Ulangi langkah sebelumnya dengan menggunakan kredensial Akun B dan buat pengguna administrator **`AccountBadmin`**.

2. Mengatur baik AWS Command Line Interface (AWS CLI) atau AWS Tools for Windows PowerShell. Pastikan Anda menyimpan kredensi pengguna administrator sebagai berikut:

   - Jika menggunakan AWS CLI, buat dua profil, `AccountAadmin` dan`AccountBadmin`, dalam file konfigurasi.
   - Jika menggunakan AWS Tools for Windows PowerShell, pastikan bahwa Anda menyimpan kredensi untuk sesi sebagai `AccountAadmin` dan. `AccountBadmin`

   Untuk petunjuk, lihat Menyiapkan alat untuk penelusuran.

3. Simpan kredensial pengguna administrator, juga disebut sebagai profil. Anda dapat menggunakan nama profil alih-alih menentukan kredensial untuk setiap perintah yang Anda masukkan. Untuk informasi selengkapnya, lihat Menyiapkan alat untuk penelusuran.

   a. Tambahkan profil dalam file AWS CLI kredensyal untuk masing-masing pengguna administrator, `AccountAadmin` dan`AccountBadmin`, di dua akun.

```
[AccountAadmin]
aws_access_key_id = access-key-ID
aws_secret_access_key = secret-access-key
region = us-east-1

[AccountBadmin]
aws_access_key_id = access-key-ID
aws_secret_access_key = secret-access-key
region = us-east-1
```

b. Jika Anda menggunakan AWS Tools for Windows PowerShell, jalankan perintah berikut.

```
set-awscredentials -AccessKey AcctA-access-key-ID -SecretKey AcctA-secret-
access-key -storeas AccountAadmin
set-awscredentials -AccessKey AcctB-access-key-ID -SecretKey AcctB-secret-
access-key -storeas AccountBadmin
```

Langkah 1: Melakukan tugas Akun A

Langkah 1.1: Masuk ke AWS Management Console

Menggunakan URL login pengguna IAM untuk Akun A, pertama-tama masuk ke pengguna AWS Management Console sebagai AccountAadmin. Pengguna ini akan membuat bucket dan melampirkan kebijakan pada bucket tersebut.

Langkah 1.2: Buat bucket

1. Pada konsol Amazon S3, buat bucket. Latihan ini mengasumsikan ember dibuat di AS Timur (Virginia N.) Wilayah AWS dan diberi nama. *amzn-s3-demo-bucket*

   Untuk petunjuk, lihat Membuat ember tujuan umum.
2. Unggah objek sampel ke bucket.

   Untuk instruksi, buka Langkah 2: Unggah objek ke bucket Anda.

Langkah 1.3: Melampirkan kebijakan bucket untuk memberikan izin lintas akun kepada Akun B

Kebijakan bucket memberikan `s3:ListBucket` izin `s3:GetLifecycleConfiguration` dan ke Akun B. Diasumsikan bahwa Anda masih masuk ke konsol menggunakan kredensil AccountAadminpengguna.

1. Melampirkan kebijakan bucket berikut untuk *amzn-s3-demo-bucket*. Kebijakan tersebut memberikan izin Akun B untuk tindakan `s3:GetLifecycleConfiguration` dan `s3:ListBucket`.

   Untuk petunjuk, lihat Menambahkan kebijakan bucket dengan menggunakan konsol Amazon S3.

   ```
   {
      "Version": "2012-10-17",
      "Statement": [
         {
            "Sid": "Example permissions",
            "Effect": "Allow",
            "Principal": {
               "AWS": "arn:aws:iam::AccountB-ID:root"
            },
            "Action": [
               "s3:GetLifecycleConfiguration",
               "s3:ListBucket"
            ],
            "Resource": [
               "arn:aws:s3:::amzn-s3-demo-bucket"
            ]
         }
      ]
   }
   ```

2. Verifikasi bahwa Akun B (dan dengan demikian pengguna administratornya) dapat melakukan operasi.

   - Verifikasi menggunakan AWS CLI

     ```
     aws s3 ls s3://amzn-s3-demo-bucket --profile AccountBadmin
     aws s3api get-bucket-lifecycle-configuration --bucket amzn-s3-demo-bucket --
     profile AccountBadmin
     ```

   - Verifikasi menggunakan AWS Tools for Windows PowerShell

```
get-s3object -BucketName amzn-s3-demo-bucket -StoredCredentials AccountBadmin
get-s3bucketlifecycleconfiguration -BucketName amzn-s3-demo-bucket -
StoredCredentials AccountBadmin
```

Langkah 2: Melakukan tugas Akun B

Sekarang administrator Akun B membuat pengguna, yakni Dave, dan mendelegasikan izin yang diterima dari Akun A.

Langkah 2.1: Masuk ke AWS Management Console

Menggunakan URL login pengguna IAM untuk Akun B, pertama-tama masuk ke pengguna AWS Management Console sebagai AccountBadmin.

Langkah 2.2: Buat Dave pengguna di Akun B

Di Konsol IAM, buat pengguna,**Dave**.

Untuk petunjuk, lihat Membuat pengguna IAM (konsol) di Panduan Pengguna IAM.

Langkah 2.3: Delegasikan izin untuk dave pengguna

Buat kebijakan yang selaras untuk pengguna Dave, dengan menggunakan kebijakan berikut. Anda harus memperbarui kebijakan dengan memberikan nama bucket Anda.

Diasumsikan bahwa Anda masuk ke konsol menggunakan AccountBadminkredensil pengguna.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "Example",
         "Effect": "Allow",
         "Action": [
            "s3:ListBucket"
         ],
         "Resource": [
            "arn:aws:s3:::amzn-s3-demo-bucket"
         ]
      }
```

```
    ]
}
```

Untuk instruksi, lihat Mengelola kebijakan IAM dalam Panduan Pengguna IAM.

Langkah 2.4: Menguji izin

Sekarang Dave di Akun B dapat mencantumkan isi *amzn-s3-demo-bucket* yang dimiliki oleh Akun A. Anda dapat memverifikasi izin menggunakan salah satu prosedur berikut.

Izin uji menggunakan AWS CLI

1. Tambahkan `UserDave` profil ke file AWS CLI konfigurasi. Untuk informasi lebih lanjut tentang file config, lihat Menyiapkan alat untuk penelusuran.

   ```
   [profile UserDave]
   aws_access_key_id = access-key
   aws_secret_access_key = secret-access-key
   region = us-east-1
   ```

2. Pada command prompt, masukkan AWS CLI perintah berikut untuk memverifikasi Dave sekarang bisa mendapatkan daftar objek dari yang *amzn-s3-demo-bucket* dimiliki oleh Account A. Catatan perintah menentukan profil. `UserDave`

   ```
   aws s3 ls s3://amzn-s3-demo-bucket --profile UserDave
   ```

   Dave tidak memiliki izin lain. Jadi, jika dia mencoba operasi lain—misalnya, `get-bucket-lifecycle` konfigurasi berikut—Amazon S3 mengembalikan izin yang ditolak.

   ```
   aws s3api get-bucket-lifecycle-configuration --bucket amzn-s3-demo-bucket --profile
    UserDave
   ```

Izin uji menggunakan AWS Tools for Windows PowerShell

1. Simpan kredensial Dave sebagai `AccountBDave`.

   ```
   set-awscredentials -AccessKey AccessKeyID -SecretKey SecretAccessKey -storeas
    AccountBDave
   ```

2. Coba perintah Cantumkan bucket.

```
get-s3object -BucketName amzn-s3-demo-bucket -StoredCredentials AccountBDave
```

Dave tidak memiliki izin lain. Jadi, jika dia mencoba operasi lain—misalnya, berikut `get-s3bucketlifecycleconfiguration` ini—Amazon S3 mengembalikan izin yang ditolak.

```
get-s3bucketlifecycleconfiguration -BucketName amzn-s3-demo-bucket -
StoredCredentials AccountBDave
```

Langkah 3: (Opsional) Coba penolakan eksplisit

Anda dapat memiliki izin yang diberikan dengan menggunakan daftar kontrol akses (ACL), kebijakan bucket, atau kebijakan pengguna. Tetapi jika ada penolakan eksplisit yang ditetapkan oleh kebijakan bucket atau kebijakan pengguna, penolakan eksplisit lebih diutamakan daripada izin lainnya. Untuk pengujian, perbarui kebijakan bucket dan tolak izin Akun B secara eksplisit. `s3:ListBucket` Kebijakan tersebut juga memberikan `s3:ListBucket` izin. Namun, penolakan eksplisit diutamakan, dan Akun B atau pengguna di Akun B tidak akan dapat mencantumkan objek. *amzn-s3-demo-bucket*

1. Menggunakan kredensial pengguna `AccountAadmin` pada Akun A, ganti kebijakan kelompok dengan yang berikut ini.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "Example permissions",
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::AccountB-ID:root"
         },
         "Action": [
            "s3:GetLifecycleConfiguration",
            "s3:ListBucket"
         ],
         "Resource": [
            "arn:aws:s3:::amzn-s3-demo-bucket"
         ]
      },
      {
```

```
            "Sid": "Deny permission",
            "Effect": "Deny",
            "Principal": {
               "AWS": "arn:aws:iam::AccountB-ID:root"
            },
            "Action": [
               "s3:ListBucket"
            ],
            "Resource": [
               "arn:aws:s3:::amzn-s3-demo-bucket"
            ]
         }
      ]
}
```

2. Sekarang jika Anda mencoba untuk mendapatkan daftar ember menggunakan `AccountBadmin` kredensional, akses ditolak.

   - Menggunakan AWS CLI, jalankan perintah berikut:

     ```
     aws s3 ls s3://amzn-s3-demo-bucket --profile AccountBadmin
     ```

   - Menggunakan AWS Tools for Windows PowerShell, jalankan perintah berikut:

     ```
     get-s3object -BucketName amzn-s3-demo-bucket -StoredCredentials AccountBDave
     ```

Langkah 4: Membersihkan

1. Setelah selesai melakukan pengujian, Anda dapat melakukan hal berikut untuk membersihkan:

   - Masuk ke AWS Management Console (AWS Management Console) menggunakan kredensi Akun A, dan lakukan hal berikut:

     - Pada konsol Amazon S3, hapus kebijakan bucket yang terlampir pada *amzn-s3-demo-bucket*. Dalam bucket Properti, hapus kebijakan yang ada dalam bagian Izin.
     - Jika bucket dibuat untuk latihan ini, pada konsol Amazon S3, hapus objek dan kemudian hapus bucket.
     - Di Konsol IAM, hapus `AccountAadmin` pengguna.

2. Masuk ke Konsol IAM menggunakan kredensil Akun B. Hapus penggunaAccountBadmin. Untuk step-by-step petunjuk, lihat Menghapus pengguna IAM di Panduan Pengguna IAM.

Contoh 3: Pemilik bucket yang memberikan izin kepada penggunanya ke objek yang bukan miliknya

> **Important**
>
> Memberikan izin untuk peran IAM adalah praktik yang lebih baik daripada memberikan izin kepada pengguna individu. Untuk mempelajari cara melakukannya, lihat Memahami izin lintas akun dan menggunakan peran IAM.

Topik

- Langkah 0: Bersiap-siap untuk panduan
- Langkah 1: Melakukan tugas Akun A
- Langkah 2: Melakukan tugas Akun B
- Langkah 3: Menguji izin
- Langkah 4: Membersihkan

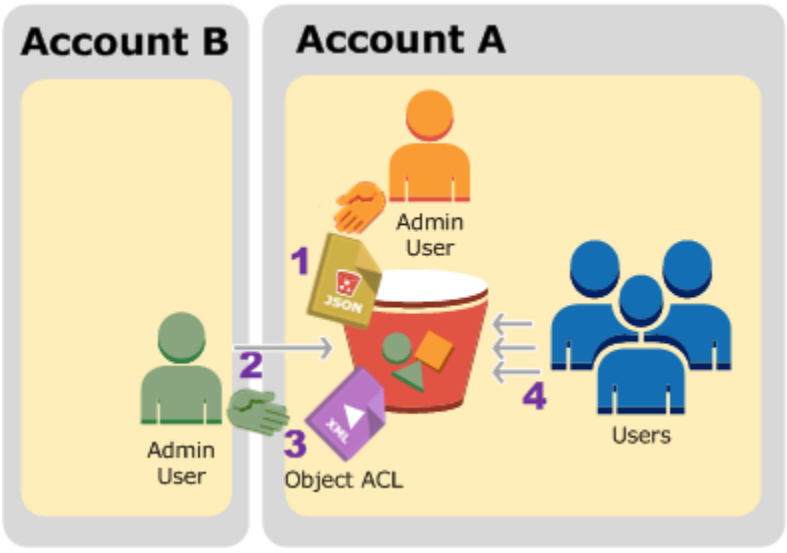
Skenario untuk contoh ini adalah bahwa pemilik bucket ingin memberikan izin untuk mengakses objek, tetapi pemilik bucket tidak memiliki semua objek di bucket. Untuk contoh ini, pemilik bucket mencoba memberikan izin kepada pengguna yang ada di akunnya sendiri.

Pemilik bucket dapat mengaktifkan orang lain Akun AWS untuk mengunggah objek. Secara default, pemilik bucket tidak memiliki objek yang ditulis ke bucket oleh orang lain Akun AWS. Objek dimiliki oleh akun yang menulisnya ke bucket S3. Jika pemilik bucket tidak memiliki objek di bucket, pemilik objek harus terlebih dahulu memberikan izin kepada pemilik bucket menggunakan daftar kontrol akses objek (ACL). Kemudian, pemilik bucket dapat memberikan izin ke objek yang tidak mereka miliki. Untuk informasi selengkapnya, lihat bucket Amazon S3 dan kepemilikan objek.

Jika pemilik bucket menerapkan pengaturan yang diberlakukan pemilik bucket untuk Kepemilikan S3 Object untuk bucket, pemilik bucket akan memiliki semua objek di bucket, termasuk objek yang ditulis oleh yang lain. Akun AWS Pendekatan ini menyelesaikan masalah bahwa objek tidak dimiliki oleh pemilik ember. Kemudian, Anda dapat mendelegasikan izin kepada pengguna di akun Anda sendiri, atau ke orang lain Akun AWS.

> **Note**
>
> S3 Object Ownership adalah setelan tingkat ember Amazon S3 yang dapat Anda gunakan untuk mengontrol kepemilikan objek yang diunggah ke bucket dan untuk menonaktifkan atau mengaktifkan. ACLs Secara default, Kepemilikan Objek disetel ke setelan diberlakukan pemilik Bucket, dan semuanya ACLs dinonaktifkan. Ketika ACLs dinonaktifkan, pemilik bucket memiliki semua objek di bucket dan mengelola akses ke mereka secara eksklusif dengan menggunakan kebijakan manajemen akses.
> Mayoritas kasus penggunaan modern di Amazon S3 tidak lagi memerlukan penggunaan. ACLs Kami menyarankan agar Anda tetap ACLs dinonaktifkan, kecuali dalam keadaan yang tidak biasa di mana Anda perlu mengontrol akses untuk setiap objek secara individual. Dengan ACLs dinonaktifkan, Anda dapat menggunakan kebijakan untuk mengontrol akses ke semua objek di bucket, terlepas dari siapa yang mengunggah objek ke bucket Anda. Untuk informasi selengkapnya, lihat Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.

Dalam contoh ini, kami menganggap pemilik bucket belum menerapkan pengaturan yang diberlakukan pemilik bucket untuk Kepemilikan Objek. Pemilik bucket mendelegasikan izin kepada pengguna di akunnya sendiri. Berikut ini adalah ringkasan dari langkah-langkah penelusuran:



1. Pengguna administrator Akun A melampirkan kebijakan bucket dengan dua pernyataan.
   - Memungkinkan izin lintas akun ke Akun B untuk mengunggah objek.
   - Memungkinkan pengguna dalam akunnya sendiri untuk mengakses objek yang ada dalam bucket.

2. Pengguna administrator Akun B mengunggah objek ke bucket yang dimiliki oleh Akun A.
3. Administrator Akun B memperbarui ACL objek yang menambahkan pemberian yang memberikan pemilik bucket izin kontrol penuh pada objek.
4. Pengguna di Akun A memverifikasi dengan mengakses objek di dalam bucket, terlepas dari siapa pemiliknya.

Untuk contoh ini, Anda memerlukan dua akun. Tabel berikut menunjukkan cara kami merujuk akun-akun ini, dan pengguna administrator yang ada dalam akun-akun tersebut. Dalam panduan ini, Anda tidak menggunakan kredensial pengguna root, menurut pedoman IAM yang direkomendasikan. Untuk informasi selengkapnya, lihat Tentang menggunakan pengguna administrator untuk membuat sumber daya dan memberikan izin. Sebagai gantinya, Anda membuat administrator di setiap akun dan menggunakan kredensial tersebut untuk membuat sumber daya dan memberikan mereka izin.

Akun AWS ID	Akun disebut sebagai	Administrator pada akun
1111-1111-1111	Akun A	AccountAadmin
2222-2222-2222	Akun B	AccountBadmin

Semua tugas membuat pengguna dan memberikan izin dilakukan di AWS Management Console. Untuk memverifikasi izin, penelusuran menggunakan alat baris perintah, AWS Command Line Interface (AWS CLI) dan AWS Tools for Windows PowerShell, jadi Anda tidak perlu menulis kode apa pun.

Langkah 0: Bersiap-siap untuk panduan

1. Pastikan Anda memiliki dua Akun AWS dan setiap akun memiliki satu administrator seperti yang ditunjukkan pada tabel di bagian sebelumnya.

   a. Mendaftar untuk Akun AWS, jika diperlukan.

   b. Menggunakan kredensi Akun A, masuk ke Konsol IAM dan lakukan hal berikut untuk membuat pengguna administrator:

      - Buat pengguna **`AccountAadmin`** dan catat kredensil keamanan pengguna. Untuk informasi lebih lanjut tentang penambahan pengguna, lihat Membuat Pengguna IAM dalam Akun AWS Anda dalam Panduan Pengguna IAM.

- Berikan izin administrator AccountAadmindengan melampirkan kebijakan pengguna yang memberikan akses penuh. Untuk instruksi, lihat Mengelola kebijakan IAM dalam Panduan Pengguna IAM.
- Di Dasbor Konsol IAM, perhatikan URL Masuk Pengguna IAM. Semua pengguna di akun tersebut harus menggunakan URL ini saat masuk ke AWS Management Console. Untuk informasi lebih lanjut, lihat Cara pengguna masuk ke akun anda dalam Panduan Pengguna IAM.

c. Ulangi langkah sebelumnya dengan menggunakan kredensial Akun B dan buat pengguna administrator **`AccountBadmin`**.

2. Siapkan salah satu AWS CLI atau Tools untuk Windows PowerShell. Pastikan Anda menyimpan kredensial administrator seperti berikut ini:

   - Jika menggunakan AWS CLI, buat dua profil, `AccountAadmin` dan`AccountBadmin`, dalam file konfigurasi.
   - Jika menggunakan Tools untuk Windows PowerShell, pastikan bahwa Anda menyimpan kredensi untuk sesi sebagai `AccountAadmin` dan. `AccountBadmin`

   Untuk petunjuk, lihat Menyiapkan alat untuk penelusuran.

Langkah 1: Melakukan tugas Akun A

Lakukan langkah-langkah berikut untuk Akun A:

Langkah 1.1: Masuk ke konsol tersebut

Menggunakan URL login pengguna IAM untuk Akun A, masuk ke pengguna AWS Management Console sebagai**`AccountAadmin`**. Pengguna ini akan membuat bucket dan melampirkan kebijakan pada bucket tersebut.

Langkah 1.2: Buat bucket dan pengguna, dan menambahkan kebijakan bucket yang memberikan izin pengguna

1. Pada konsol Amazon S3, buat bucket. Latihan ini mengasumsikan bahwa ember dibuat di AS Timur (Virginia N.) Wilayah AWS, dan namanya adalah. *`amzn-s3-demo-bucket1`*

   Untuk petunjuk, lihat Membuat ember tujuan umum.

2. Di Konsol IAM, buat pengguna**`Dave`**.

Untuk step-by-step petunjuk, lihat Membuat pengguna IAM (konsol) di Panduan Pengguna IAM.

3. Perhatikan kredensi Dave pengguna.
4. Pada konsol Amazon S3, lampirkan kebijakan bucket berikut ini ke bucket *amzn-s3-demo-bucket1*. Untuk petunjuk, lihat Menambahkan kebijakan bucket dengan menggunakan konsol Amazon S3. Ikuti langkah-langkah untuk menambahkan kebijakan bucket. Untuk informasi tentang cara menemukan akun IDs, lihat Menemukan Akun AWS ID Anda.

   Kebijakan tersebut memberikan izin `s3:PutObject` dan `s3:ListBucket` kepada Akun B. Kebijakan ini juga memberikan `s3:GetObject` izin `Dave` kepada pengguna.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "Statement1",
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::AccountB-ID:root"
         },
         "Action": [
            "s3:PutObject",
            "s3:ListBucket"
         ],
         "Resource": [
            "arn:aws:s3:::amzn-s3-demo-bucket1/*",
            "arn:aws:s3:::amzn-s3-demo-bucket1"
         ]
      },
      {
         "Sid": "Statement3",
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::AccountA-ID:user/Dave"
         },
         "Action": [
            "s3:GetObject"
         ],
         "Resource": [
            "arn:aws:s3:::amzn-s3-demo-bucket1/*"
         ]
      }
```

```
    ]
}
```

Langkah 2: Melakukan tugas Akun B

Setelah Akun B memiliki izin untuk melakukan operasi pada bucket Akun A, administrator Akun B melakukan hal berikut:

- Mengunggah objek ke bucket Akun A
- Menambahkan hibah di objek ACL untuk mengizinkan Akun A, pemilik bucket, kontrol penuh

Menggunakan AWS CLI

1. Menggunakan `put-object` AWS CLI perintah, unggah objek. Parameter `--body` dalam perintah mengidentifikasi file sumber yang akan diunggah. Misalnya, jika file ada di `C:` drive Windows mesin, tentukan`c:\HappyFace.jpg`. Parameter `--key` menyediakan nama kunci untuk objek.

   ```
   aws s3api put-object --bucket amzn-s3-demo-bucket1 --key HappyFace.jpg --body
    HappyFace.jpg --profile AccountBadmin
   ```

2. Tambahkan pemberian pada ACL objek untuk memungkinkan pemilik bucket mengontrol penuh objek. Untuk informasi tentang cara menemukan ID pengguna kanonik, lihat Menemukan ID pengguna kanonik untuk Anda Akun AWS di Panduan Referensi Manajemen AWS Akun.

   ```
   aws s3api put-object-acl --bucket amzn-s3-demo-bucket1 --key HappyFace.jpg --grant-
   full-control id="AccountA-CanonicalUserID" --profile AccountBadmin
   ```

Menggunakan Alat untuk Windows PowerShell

1. Menggunakan `Write-S3Object` perintah, unggah objek.

   ```
   Write-S3Object -BucketName amzn-s3-demo-bucket1 -key HappyFace.jpg -file
    HappyFace.jpg -StoredCredentials AccountBadmin
   ```

2. Tambahkan pemberian pada ACL objek untuk memungkinkan pemilik bucket mengontrol penuh objek.

```
Set-S3ACL -BucketName amzn-s3-demo-bucket1 -Key HappyFace.jpg -CannedACLName
 "bucket-owner-full-control" -StoredCreden
```

Langkah 3: Menguji izin

Sekarang, verifikasi pengguna Dave di Akun A apakah dapat mengakses objek yang dimiliki oleh Akun B.

Menggunakan AWS CLI

1. Tambahkan kredensi Dave pengguna ke file AWS CLI konfigurasi dan buat profil baru,. `UserDaveAccountA` Untuk informasi selengkapnya, lihat Menyiapkan alat untuk penelusuran.

```
[profile UserDaveAccountA]
aws_access_key_id = access-key
aws_secret_access_key = secret-access-key
region = us-east-1
```

2. Jalankan perintah `get-object` CLI untuk mengunduh `HappyFace.jpg` dan menyimpannya secara lokal. Anda memberikan kredensial pengguna Dave dengan menambahkan parameter `--profile`.

```
aws s3api get-object --bucket amzn-s3-demo-bucket1 --key
 HappyFace.jpg Outputfile.jpg --profile UserDaveAccountA
```

Menggunakan Alat untuk Windows PowerShell

1. Simpan AWS kredensi Dave pengguna, seperti`UserDaveAccountA`, ke toko persisten.

```
Set-AWSCredentials -AccessKey UserDave-AccessKey -SecretKey UserDave-
SecretAccessKey -storeas UserDaveAccountA
```

2. Jalankan perintah `Read-S3Object` untuk mengunduh objek `HappyFace.jpg` dan menyimpannya secara lokal. Anda memberikan kredensial ke pengguna Dave dengan menambahkan parameter `-StoredCredentials`.

```
Read-S3Object -BucketName amzn-s3-demo-bucket1 -Key HappyFace.jpg -file
 HappyFace.jpg  -StoredCredentials UserDaveAccountA
```

Langkah 4: Membersihkan

1. Setelah selesai melakukan pengujian, Anda dapat melakukan hal berikut untuk membersihkan:
   - Masuk ke AWS Management Console menggunakan kredensial Akun A, dan lakukan hal-hal berikut ini:
     - Pada konsol Amazon S3, hapus kebijakan bucket yang terlampir pada *amzn-s3-demo-bucket1*. Dalam bucket Properti, hapus kebijakan yang ada dalam bagian Izin.
     - Jika bucket dibuat untuk latihan ini, pada konsol Amazon S3, hapus objek dan kemudian hapus bucket.
     - Di Konsol IAM, hapus AccountAadminpengguna. Untuk step-by-step petunjuk, lihat Menghapus pengguna IAM di Panduan Pengguna IAM.
2. Masuklah ke AWS Management Console menggunakan kredensial Akun B. Di Konsol IAM, hapus pengguna AccountBadmin.

Contoh 4 - Pemilik bucket memberikan izin lintas akun ke objek yang tidak dimilikinya

Topik

- Memahami izin lintas akun dan menggunakan peran IAM
- Langkah 0: Bersiap-siap untuk panduan
- Langkah 1: Melakukan tugas Akun A
- Langkah 2: Melakukan tugas Akun B
- Langkah 3: Lakukan tugas Akun C
- Langkah 4: Membersihkan
- Sumber daya terkait

Dalam skenario contoh ini, Anda memiliki bucket dan Anda telah mengaktifkan orang lain Akun AWS untuk mengunggah objek. Jika Anda telah menerapkan pengaturan yang diberlakukan pemilik bucket untuk Kepemilikan S3 Object untuk bucket, Anda akan memiliki semua objek di bucket,

termasuk objek yang ditulis oleh objek lain. Akun AWS Pendekatan ini menyelesaikan masalah bahwa objek tidak dimiliki oleh Anda, pemilik ember. Kemudian, Anda dapat mendelegasikan izin kepada pengguna di akun Anda sendiri, atau ke orang lain Akun AWS. Misalkan pengaturan yang diberlakukan pemilik bucket untuk Kepemilikan S3 Object tidak diaktifkan. Itu berarti, bucket Anda dapat memiliki objek yang dimiliki Akun AWS lainnya.

Sekarang, seandainya sebagai pemilik bucket, Anda perlu memberikan izin lintas akun atas objek, terlepas dari siapa pemilik objek itu, kepada pengguna yang ada di akun lain. Sebagai contoh, pengguna tersebut bisa berupa aplikasi penagihan yang memerlukan akses terhadap metadata objek. Ada dua masalah inti:

- Pemilik bucket tidak memiliki izin pada objek yang dibuat oleh Akun AWS lain. Agar pemilik bucket dapat memberikan izin pada objek yang tidak dimilikinya, pemilik objek harus terlebih dahulu memberikan izin kepada pemilik bucket. Pemilik objek adalah Akun AWS yang menciptakan objek. Pemilik bucket dapat mendelegasikan izin tersebut.
- Akun pemilik bucket dapat mendelegasikan izin kepada pengguna di akunnya sendiri (lihatContoh 3: Pemilik bucket yang memberikan izin kepada penggunanya ke objek yang bukan miliknya). Namun, akun pemilik bucket tidak dapat mendelegasikan izin ke akun lain Akun AWS karena delegasi lintas akun tidak didukung.

Dalam skenario ini, pemilik bucket dapat membuat peran AWS Identity and Access Management (IAM) dengan izin untuk mengakses objek Kemudian, pemilik bucket dapat memberikan Akun AWS izin lain untuk mengambil peran tersebut, untuk sementara memungkinkannya mengakses objek di bucket.

> **Note**
>
> S3 Object Ownership adalah setelan tingkat ember Amazon S3 yang dapat Anda gunakan untuk mengontrol kepemilikan objek yang diunggah ke bucket dan untuk menonaktifkan atau mengaktifkan. ACLs Secara default, Kepemilikan Objek disetel ke setelan diberlakukan pemilik Bucket, dan semuanya ACLs dinonaktifkan. Ketika ACLs dinonaktifkan, pemilik bucket memiliki semua objek di bucket dan mengelola akses ke mereka secara eksklusif dengan menggunakan kebijakan manajemen akses.
> Mayoritas kasus penggunaan modern di Amazon S3 tidak lagi memerlukan penggunaan. ACLs Kami menyarankan agar Anda tetap ACLs dinonaktifkan, kecuali dalam keadaan yang tidak biasa di mana Anda perlu mengontrol akses untuk setiap objek secara individual. Dengan ACLs dinonaktifkan, Anda dapat menggunakan kebijakan untuk mengontrol akses ke

semua objek di bucket, terlepas dari siapa yang mengunggah objek ke bucket Anda. Untuk informasi selengkapnya, lihat Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.

Memahami izin lintas akun dan menggunakan peran IAM

Peran IAM memungkinkan beberapa skenario untuk mendelegasikan akses ke sumber daya Anda, dan akses lintas akun adalah salah satu skenario utamanya. Dalam contoh ini, pemilik bucket, Akun A, menggunakan peran IAM untuk sementara mendelegasikan akses objek lintas akun ke pengguna lain Akun AWS, Akun C. Setiap peran IAM yang Anda buat memiliki dua kebijakan berikut yang dilampirkan padanya:

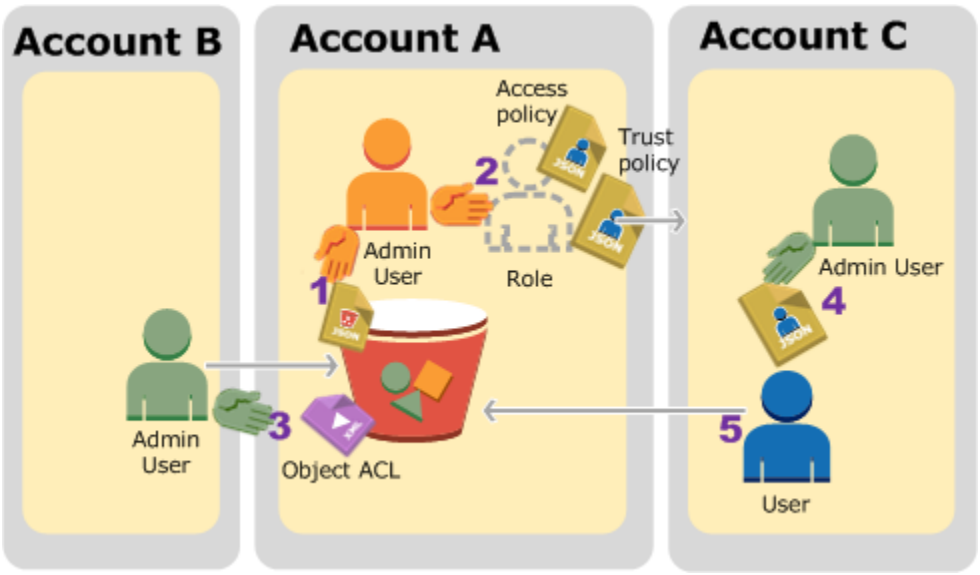
- Kebijakan kepercayaan mengidentifikasi orang lain Akun AWS yang dapat mengambil peran.
- Kebijakan akses yang menentukan izin—misalnya, `s3:GetObject`—yang diperkenankan saat seseorang mengambil peran tersebut. Untuk daftar izin yang dapat Anda tentukan dalam kebijakan, lihat Tindakan kebijakan untuk Amazon S3.

Yang Akun AWS diidentifikasi dalam kebijakan kepercayaan kemudian memberikan izin penggunanya untuk mengambil peran. Pengguna kemudian dapat melakukan hal-hal berikut untuk mengakses objek:

- Mengambil peran dan, sebagai tanggapan, mendapatkan kredensial keamanan sementara.
- Dengan menggunakan kredensial keamanan sementara, pengguna dapat mengakses objek yang ada dalam bucket.

Untuk informasi selengkapnya tentang peran IAM, lihat Peran IAM dalam Panduan Pengguna IAM.

Berikut ini adalah ringkasan dari langkah-langkah penelusuran:



1. Akun Pengguna administrator melampirkan kebijakan bucket yang memberikan izin bersyarat kepada Akun B untuk mengunggah objek.
2. Akun Administrator membuat peran IAM, membangun kepercayaan dengan Akun C, sehingga pengguna dalam akun tersebut dapat mengakses Akun A. Kebijakan akses yang terlampir pada peran tersebut membatasi apa yang dapat dilakukan pengguna yang ada di Akun C saat pengguna mengakses Akun A.
3. Administrator Akun B mengunggah objek ke bucket yang dimiliki oleh Akun A, memberikan izin kendali penuh kepada pemilik bucket.
4. Administrator Akun C membuat pengguna dan melampirkan kebijakan pengguna yang memungkinkan pengguna untuk mengambil peran tersebut.
5. Pengguna di Akun C terlebih dahulu harus mengambil peran tersebut, yang akan mengembalikan kredensial keamanan sementara kepada pengguna. Menggunakan kredensial sementara, pengguna kemudian dapat mengakses objek dalam bucket.

Untuk contoh ini, Anda memerlukan tiga akun. Tabel berikut menunjukkan cara kami merujuk akun-akun ini, dan pengguna administrator yang ada dalam akun-akun tersebut. Sesuai dengan pedoman IAM (lihatTentang menggunakan pengguna administrator untuk membuat sumber daya dan memberikan izin), kami tidak menggunakan Pengguna root akun AWS kredensil dalam panduan ini. Sebagai gantinya, Anda membuat pengguna administrator di setiap akun dan menggunakan kredensial tersebut saat membuat sumber daya dan memberikan mereka izin.

Akun AWS ID	Akun disebut sebagai	Pengguna administrator pada akun
1111-1111-1111	Akun A	AccountAadmin
2222-2222-2222	Akun B	AccountBadmin
3333-3333-3333	Akun C	AccountCadmin

Langkah 0: Bersiap-siap untuk panduan

> **Note**
>
> Anda mungkin ingin membuka editor teks, dan menuliskan beberapa informasi saat Anda melalui langkah-langkahnya. Secara khusus, Anda akan memerlukan akun IDs, pengguna kanonik IDs, pengguna IAM Masuk URLs untuk setiap akun untuk terhubung ke konsol, dan Nama Sumber Daya Amazon (ARNs) dari pengguna IAM, dan peran.

1. Pastikan Anda memiliki tiga Akun AWS dan setiap akun memiliki satu pengguna administrator seperti yang ditunjukkan pada tabel di bagian sebelumnya.

   a. Mendaftar untuk Akun AWS, sesuai kebutuhan. Kami menyebut akun ini sebagai Akun A, Akun B, dan Akun C.

   b. Menggunakan kredensial Akun A, masuklah ke Konsol IAM dan lakukan hal berikut ini untuk membuat pengguna administrator:

      - Buat pengguna **`AccountAadmin`** dan catat kredensyal keamanannya. Untuk informasi lebih lanjut tentang penambahan pengguna, lihat Membuat Pengguna IAM dalam Akun AWS Anda dalam Panduan Pengguna IAM.
      - Berikan hak administrator AccountAadmindengan melampirkan kebijakan pengguna yang memberikan akses penuh. Untuk instruksi, lihat Mengelola kebijakan IAM dalam Panduan Pengguna IAM.
      - Di Dasbor Konsol IAM, perhatikan URL Masuk Pengguna IAM. Semua pengguna di akun tersebut harus menggunakan URL ini saat masuk ke AWS Management Console.

Untuk informasi selengkapnya, lihat Masuk ke pengguna IAM AWS Management Console sebagai pengguna IAM di Panduan Pengguna IAM.

c. Ulangi langkah-langkah sebelumnya untuk membuat pengguna administrator di Akun B dan Akun C.

2. Untuk Akun C, catat ID pengguna kanonik.

   Saat Anda membuat peran IAM di Akun A, kebijakan kepercayaan memberikan izin kepada Akun C untuk mengambil peran tersebut dengan menyebutkan ID akunnya. Anda dapat menemukan informasi akun seperti berikut ini:

   a. Gunakan Akun AWS ID atau alias akun, nama pengguna IAM, dan kata sandi Anda untuk masuk ke konsol Amazon S3.

   b. Pilih nama bucket Amazon S3 untuk melihat detail tentang bucket tersebut.

   c. Pilih tab Izin dan kemudian pilih Daftar Kontrol Akses.

   d. Pada bagian Akses Akun AWS Anda, pada kolom Akun adalah pengidentifikasi panjang, seperti `c1daexampleaaf850ea79cf0430f33d72579fd1611c97f7ded193374c0b163b6`. Ini adalah ID pengguna kanonis Anda.

3. Saat membuat kebijakan bucket, Anda akan memerlukan informasi berikut. Perhatikan nilai-nilai ini:

   - ID pengguna Kanonik Akun A–Ketika administrator Akun A memberikan izin unggah objek bersyarat kepada administrator Akun B, kondisinya menetapkan ID pengguna kanonik dari Akun A yang harus mendapatkan kontrol penuh atas objek.

     > **Note**
     >
     > ID pengguna kanonik adalah konsep Amazon S3 saja. ID ini adalah ID akun 64 karakter yang dikaburkan.

   - Pengguna ARN untuk Administrator Akun B — Anda dapat menemukan ARN pengguna di Konsol IAM. Anda harus memilih pengguna dan menemukan ARN pengguna di tab Ringkasan.

     Dalam kebijakan ember, Anda memberikan kepada `AccountBadmin` izin untuk mengunggah objek dan Anda menentukan pengguna dengan menggunakan ARN. Berikut ini adalah contoh nilai ARN:

```
arn:aws:iam::AccountB-ID:user/AccountBadmin
```

4. Siapkan AWS Command Line Interface (CLI) atau. AWS Tools for Windows PowerShell Pastikan Anda menyimpan kredensi pengguna administrator sebagai berikut:

   - Jika menggunakan AWS CLI, buat profil, `AccountAadmin` dan`AccountBadmin`, dalam file konfigurasi.
   - Jika menggunakan AWS Tools for Windows PowerShell, pastikan bahwa Anda menyimpan kredensi untuk sesi sebagai `AccountAadmin` dan. `AccountBadmin`

   Untuk petunjuk, lihat Menyiapkan alat untuk penelusuran.

Langkah 1: Melakukan tugas Akun A

Dalam contoh ini, Akun A adalah pemilik bucket. Jadi pengguna AccountAadmin di Akun A akan melakukan hal berikut:

- Buat bucket.
- Lampirkan kebijakan bucket yang memberikan izin administrator Akun B untuk mengunggah objek.
- Buat peran IAM yang memberikan izin Akun C untuk mengambil peran sehingga dapat mengakses objek di bucket.

Langkah 1.1: Masuk ke AWS Management Console

Menggunakan URL Masuk pengguna IAM untuk Akun A, pertama-tama masuk ke pengguna AWS Management Console sebagai**`AccountAadmin`**. Pengguna ini akan membuat bucket dan melampirkan kebijakan pada bucket tersebut.

Langkah 1.2: Buat bucket dan melampirkan kebijakan bucket

Di konsol Amazon S3, lakukan berikut ini:

1. Buat bucket. Latihan ini mengasumsikan bahwa nama bucketnya adalah *`amzn-s3-demo-bucket1`*.

   Untuk petunjuk, lihat Membuat ember tujuan umum.

2. Lampirkan kebijakan bucket berikut. Kebijakan memberikan izin bersyarat kepada administrator Akun B untuk mengunggah objek.

   Perbarui kebijakan dengan memberikan nilai Anda sendiri untuk *amzn-s3-demo-bucket1AccountB-ID*,, dan*CanonicalUserId-of-AWSaccountA-BucketOwner*.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "111",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::AccountB-ID:user/AccountBadmin"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*"
        },
        {
            "Sid": "112",
            "Effect": "Deny",
            "Principal": {
                "AWS": "arn:aws:iam::AccountB-ID:user/AccountBadmin"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*",
            "Condition": {
                "StringNotEquals": {
                    "s3:x-amz-grant-full-control": "id=CanonicalUserId-of-
AWSaccountA-BucketOwner"
                }
            }
        }
    ]
}
```

Langkah 1.3: Buat peran IAM untuk memungkinkan akses lintas akun Akun C di Akun A

Di Konsol IAM, buat peran IAM (**examplerole**) yang memberikan izin Akun C untuk mengambil peran tersebut. Pastikan Anda masih masuk sebagai administrator Akun A karena peran harus dibuat di Akun A.

1. Sebelum membuat peran, siapkan kebijakan terkelola yang menentukan izin yang diperlukan oleh peran tersebut. Anda kemudian perlu melampirkan kebijakan ini ke peran tersebut pada langkah berikutnya.

   a. Di panel navigasi di sebelah kiri, pilih Kebijakan, lalu pilih Buat Kebijakan.

   b. Di sebelah Buat Kebijakan Anda Sendiri, pilih Pilih.

   c. Masukkan **`access-accountA-bucket`** dalam kolom Nama Kebijakan.

   d. Salin kebijakan akses berikut ini dan tempelkan ke kolom Dokumen Kebijakan. Kebijakan akses memberikan `s3:GetObject` izin peran sehingga, ketika pengguna Akun C mengambil peran, kebijakan tersebut hanya dapat melakukan operasi. `s3:GetObject`

   ```
   {
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Action": "s3:GetObject",
         "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*"
       }
     ]
   }
   ```

   e. Pilih Buat Kebijakan.

   Kebijakan baru muncul di daftar kebijakan terkelola.

2. Di panel navigasi di sebelah kiri, pilih Peran dan kemudian pilih Buat Peran Baru.
3. Di bawah Pilih Jenis Peran, pilih Peran untuk Akses Lintas Akun, lalu pilih tombol Pilih di samping Menyediakan akses di antara Akun AWS Anda sendiri.
4. Masukkan ID akun dari Akun C.

   Untuk panduan ini, Anda tidak perlu meminta pengguna untuk memiliki otentikasi multi-faktor (MFA) untuk mengambil peran, jadi biarkan opsi itu tidak dipilih.
5. Pilih Langkah Berikutnya untuk mengatur izin yang akan dikaitkan dengan peran.
6. Pilih kotak centang di samping kebijakan Access-Accounta-bucket yang Anda buat, lalu pilih Langkah Berikutnya.

   Halaman Tinjauan muncul sehingga Anda dapat mengonfirmasi pengaturan untuk peran tersebut sebelum dibuat. Satu item yang sangat penting untuk dicatat di halaman ini adalah tautan yang

dapat Anda kirim kepada pengguna Anda yang perlu menggunakan peran ini. Pengguna yang menggunakan tautan langsung menuju halaman Switch Role dengan kolom Account ID dan Role Name yang sudah diisi. Anda juga dapat melihat tautan ini nanti di halaman Ringkasan Peran untuk peran lintas akun apa pun.

7. Masukkan `examplerole` nama peran, lalu pilih Langkah Berikutnya.
8. Setelah meninjau peran, pilih Buat Peran.

   Peran `examplerole` ditampilkan dalam daftar peran.
9. Pilih nama peran`examplerole`.
10. Pilih tab Hubungan Kepercayaan.
11. Pilih Tampilkan dokumen kebijakan dan verifikasi kebijakan kepercayaan yang ditampilkan cocok dengan kebijakan berikut.

    Kebijakan kepercayaan berikut ini membangun kepercayaan terhadap Akun C, dengan memungkinkannya untuk melakukan tindakan `sts:AssumeRole`. Untuk informasi selengkapnya, lihat AssumeRole di dalam Referensi API AWS Security Token Service .

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::AccountC-ID:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

12. Perhatikan Nama Sumber Daya Amazon (ARN) dari `examplerole` peran yang Anda buat.

    Kemudian pada langkah-langkah berikut ini, Anda melampirkan kebijakan pengguna untuk memungkinkan pengguna IAM mengambil peran ini, dan kemudian Anda mengidentifikasi peran berdasarkan nilai ARN.

Langkah 2: Melakukan tugas Akun B

Contoh bucket yang dimiliki oleh Akun A memerlukan objek yang dimiliki oleh akun lain. Pada langkah ini, administrator Akun B mengunggah objek dengan menggunakan alat baris perintah.

- Menggunakan `put-object` AWS CLI perintah, unggah objek ke*amzn-s3-demo-bucket1*.

  ```
  aws s3api put-object --bucket amzn-s3-demo-bucket1 --key HappyFace.jpg --
  body HappyFace.jpg --grant-full-control id="canonicalUserId-ofTheBucket0wner" --
  profile AccountBadmin
  ```

  Perhatikan hal berikut:

  - `--Profile`Parameter menentukan `AccountBadmin` profil, sehingga objek dimiliki oleh Akun B.
  - Parameter `grant-full-control` memberikan izin kendali penuh kepada pemilik bucket pada objek sebagaimana yang diwajibkan oleh kebijakan bucket.
  - Parameter `--body` mengidentifikasi file sumber yang akan diunggah. Misalnya, jika file tersebut ada di drive C: Windows komputer, Anda tentukan`c:\HappyFace.jpg`.

Langkah 3: Lakukan tugas Akun C

Pada langkah-langkah sebelumnya, Akun A telah menciptakan peran`examplerole`, membangun kepercayaan dengan Akun C. Peran ini memungkinkan pengguna di Akun C untuk mengakses Akun A. Pada langkah ini, administrator Akun C membuat pengguna (Dave) dan mendelegasikan kepadanya `sts:AssumeRole` izin yang diterima dari Akun A. Pendekatan ini memungkinkan Dave untuk mengasumsikan `examplerole` dan untuk sementara mendapatkan akses ke Akun A. Kebijakan akses yang Akun A melekat pada peran membatasi apa yang dapat dilakukan Dave ketika dia mengakses Akun A—khususnya, masukkan objek. *amzn-s3-demo-bucket1*

Langkah 3.1: Buat pengguna di Akun C dan delegasikan izin untuk berasumsi examplerole

1. Menggunakan URL login pengguna IAM untuk Akun C, pertama-tama masuk ke pengguna AWS Management Console sebagai**`AccountCadmin`**.

2. Di Konsol IAM, buat pengguna, Dave.

   Untuk step-by-step petunjuk, lihat Membuat pengguna IAM (AWS Management Console) di Panduan Pengguna IAM.

3. Perhatikan kredensil Dave. Dave akan memerlukan kredensial ini untuk mengambil peran `examplerole`.
4. Buat kebijakan inline untuk pengguna Dave IAM untuk mendelegasikan `sts:AssumeRole` izin kepada Dave pada peran di Akun A. `examplerole`

   a. Pada panel navigasi di sebelah kiri, pilih Pengguna.

   b. Pilih nama pengguna Dave.

   c. Pada halaman rincian pengguna, pilih tab Izin dan kemudian perluas bagian Kebijakan Selaras.

   d. Pilih klik di sini (atau Buat Kebijakan Pengguna).

   e. Pilih Kebijakan Kustom, lalu pilih Pilihan.

   f. Masukkan nama untuk kebijakan pada kolom Nama Kebijakan.

   g. Salin kebijakan berikut ke kolom Dokumen Kebijakan.

      Anda harus memperbarui kebijakan dengan menyediakan*AccountA-ID*.

      ```
      {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Effect": "Allow",
            "Action": ["sts:AssumeRole"],
            "Resource": "arn:aws:iam::AccountA-ID:role/examplerole"
          }
        ]
      }
      ```

   h. Pilih Terapkan Kebijakan.

5. Simpan kredensi Dave ke file konfigurasi AWS CLI dengan menambahkan profil lain,. `AccountCDave`

   ```
   [profile AccountCDave]
   aws_access_key_id = UserDaveAccessKeyID
   aws_secret_access_key = UserDaveSecretAccessKey
   region = us-west-2
   ```

Langkah 3.2: Asumsikan peran (examplerole) dan akses objek

Kini Dave dapat mengakses objek yang ada dalam bucket yang dimiliki oleh Akun A seperti berikut ini:

- Dave terlebih dahulu harus mengambil `examplerole` menggunakan kredensialnya sendiri. Hal ini akan mengembalikan kredensial sementara.
- Dengan menggunakan kredensial sementara, Dave kemudian akan dapat mengakses objek yang ada dalam bucket Akun A.

1. Pada prompt perintah, jalankan AWS CLI `assume-role` perintah berikut menggunakan `AccountCDave` profil.

   Anda harus memperbarui nilai ARN dalam perintah dengan memberikan di *`AccountA-ID`* mana `examplerole` didefinisikan.

   ```
   aws sts assume-role --role-arn arn:aws:iam::AccountA-ID:role/examplerole --profile
    AccountCDave --role-session-name test
   ```

   Sebagai tanggapan, AWS Security Token Service (AWS STS) mengembalikan kredensyal keamanan sementara (ID kunci akses, kunci akses rahasia, dan token sesi).

2. Simpan kredensil keamanan sementara di file AWS CLI konfigurasi di bawah profil. `TempCred`

   ```
   [profile TempCred]
   aws_access_key_id = temp-access-key-ID
   aws_secret_access_key = temp-secret-access-key
   aws_session_token = session-token
   region = us-west-2
   ```

3. Pada prompt perintah, jalankan AWS CLI perintah berikut untuk mengakses objek menggunakan kredenal sementara. Misalnya, perintah menentukan head-object API untuk mengambil metadata objek untuk `HappyFace.jpg` objek.

   ```
   aws s3api get-object --bucket amzn-s3-demo-bucket1 --
   key HappyFace.jpg SaveFileAs.jpg --profile TempCred
   ```

Karena kebijakan akses terlampir pada `examplerole` memungkinkan tindakan, Amazon S3 memproses permintaan. Anda dapat mencoba tindakan lain apa pun pada objek lain yang ada dalam bucket.

Jika Anda mencoba tindakan lain—misalnya, `get-object-acl` —Anda akan mendapatkan izin ditolak karena peran tersebut tidak diizinkan tindakan itu.

```
aws s3api get-object-acl --bucket amzn-s3-demo-bucket1 --key HappyFace.jpg --
profile TempCred
```

Kita menggunakan pengguna Dave untuk mengambil peran dan mengakses objek dengan menggunakan kredensial sementara. Ini juga bisa menjadi aplikasi di Akun C yang mengakses objek di *amzn-s3-demo-bucket1*. Aplikasi dapat memperoleh kredensial keamanan sementara, dan Akun C dapat mendelegasikan izin aplikasi untuk mengambil `examplerole`.

Langkah 4: Membersihkan

1. Setelah selesai melakukan pengujian, Anda dapat melakukan hal berikut untuk membersihkan:
   - Masuk ke AWS Management Console menggunakan kredensial Akun A, dan lakukan hal-hal berikut ini:
     - Pada konsol Amazon S3, hapus kebijakan bucket yang terlampir pada *amzn-s3-demo-bucket1*. Dalam bucket Properti, hapus kebijakan yang ada dalam bagian Izin.
     - Jika bucket dibuat untuk latihan ini, pada konsol Amazon S3, hapus objek dan kemudian hapus bucket.
     - Di Konsol IAM, hapus yang `examplerole` Anda buat di Akun A. Untuk step-by-step petunjuk, lihat Menghapus pengguna IAM di Panduan Pengguna IAM.
     - Di Konsol IAM, hapus AccountAadminpengguna.
2. Masuk ke Konsol IAM dengan menggunakan kredensil Akun B. Hapus pengguna AccountBadmin.
3. Masuk ke Konsol IAM dengan menggunakan kredensi Akun C. Hapus AccountCadmindan pengguna Dave.

Sumber daya terkait

Untuk informasi selengkapnya yang terkait dengan panduan ini, lihat sumber daya berikut di Panduan Pengguna IAM:

- Membuat peran untuk mendelegasikan izin ke pengguna IAM
- Tutorial: Mendelegasikan Akses di Seluruh Akun AWS Menggunakan Peran IAM
- Mengelola kebijakan IAM

Menggunakan Peran Terkait Layanan untuk Lensa Penyimpanan Amazon S3

Untuk menggunakan Lensa Penyimpanan Amazon S3 untuk mengumpulkan dan mengumpulkan metrik di semua akun Anda di AWS Organizations, Anda harus terlebih dahulu memastikan bahwa Lensa Penyimpanan S3 memiliki akses tepercaya yang diaktifkan oleh akun kelola di organisasi Anda. S3 Storage Lens membuat peran terkait layanan (SLR) untuk memungkinkannya mendapatkan daftar Akun AWS milik organisasi Anda. Daftar akun ini digunakan oleh Lensa Penyimpanan S3 untuk mengumpulkan metrik untuk sumber daya S3 di semua akun anggota ketika dasbor Lensa Penyimpanan S3 atau konfigurasi dibuat atau diperbarui.

Amazon S3 Storage Lens menggunakan peran terkait AWS Identity and Access Management layanan (IAM). Peran terkait layanan adalah tipe peran IAM unik yang ditautkan langsung ke Lensa Penyimpanan S3. Peran terkait layanan telah ditentukan sebelumnya oleh S3 Storage Lens dan mencakup semua izin yang diperlukan layanan untuk memanggil orang lain atas nama Anda. Layanan AWS

Peran terkait layanan membuat pengaturan Lensa Penyimpanan S3 menjadi lebih mudah karena Anda tidak perlu menambahkan izin yang diperlukan secara manual. Lensa Penyimpanan S3 menentukan izin peran terkait layanan, kecuali jika ditentukan berbeda, hanya Lensa Penyimpanan S3 yang dapat mengasumsikan peran tersebut. Izin yang ditentukan mencakup kebijakan kepercayaan dan kebijakan izin, dan kebijakan izin tersebut tidak dapat dilampirkan ke entitas IAM lainnya.

Anda dapat menghapus peran terkait layanan hanya setelah menghapus sumber daya terkait terlebih dahulu. Ini melindungi sumber daya Lensa Penyimpanan S3 Anda karena Anda tidak dapat secara tidak sengaja menghapus izin untuk mengakses sumber daya.

Untuk informasi tentang layanan lain yang mendukung peran terkait layanan, lihat Layanan AWS yang berfungsi dengan IAM dan cari layanan yang memiliki Ya di dalam Peran Terkait Layanan. Pilih Ya bersama tautan untuk melihat dokumentasi peran tertaut layanan untuk layanan tersebut.

Menggunakan peran terkait layanan untuk Lensa Penyimpanan Amazon S3

Lensa Penyimpanan S3 menggunakan peran terkait layanan bernama AWSServiceRoleForS3 StorageLens — Ini memungkinkan akses ke AWS layanan dan sumber daya yang digunakan atau dikelola oleh S3 Storage Lens. Hal ini memungkinkan S3 Storage Lens untuk mengakses AWS Organizations sumber daya atas nama Anda.

Peran yang berkaitan dengan layanan Lensa Penyimpanan S3 memercayakan layanan berikut di penyimpanan organisasi Anda:

- `storage-lens.s3.amazonaws.com`

Kebijakan izin peran mengizinkan Lensa Penyimpanan S3 untuk menyelesaikan tindakan berikut:

- `organizations:DescribeOrganization`

  `organizations:ListAccounts`

  `organizations:ListAWSServiceAccessForOrganization`

  `organizations:ListDelegatedAdministrators`

Anda harus mengonfigurasi izin untuk mengizinkan entitas IAM (seperti pengguna, grup, atau peran) untuk membuat, mengedit, atau menghapus peran terkait layanan. Untuk informasi selengkapnya, lihat Izin peran tertaut layanan dalam Panduan Pengguna IAM.

Membuat peran terkait layanan untuk Lensa Penyimpanan S3

Anda tidak perlu membuat peran terkait layanan secara manual. Saat Anda menyelesaikan salah satu tugas berikut saat masuk ke AWS Organizations manajemen atau akun administrator delegasi, S3 Storage Lens akan membuat peran terkait layanan untuk Anda:

- Buat konfigurasi dasbor Lensa Penyimpanan S3 untuk organisasi Anda di konsol Amazon S3.
- PUTkonfigurasi Lensa Penyimpanan S3 untuk organisasi Anda menggunakan REST API, AWS CLI dan SDKs.

> **Note**
>
> S3 Storage Lens akan mendukung maksimal lima administrator yang didelegasikan per organisasi.

Jika Anda menghapus peran terkait layanan ini, tindakan sebelumnya akan dibuat ulang sesuai kebutuhan.

Kebijakan contoh untuk peran terkait layanan Lensa Penyimpanan S3

Example Kebijakan izin untuk peran terkait layanan Lensa Penyimpanan Amazon S3

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AwsOrgsAccess",
            "Effect": "Allow",
            "Action": [
                "organizations:DescribeOrganization",
                "organizations:ListAccounts",
                "organizations:ListAWSServiceAccessForOrganization",
                "organizations:ListDelegatedAdministrators"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Mengedit peran terkait layanan untuk Lensa Penyimpanan Amazon S3

S3 Storage Lens tidak memungkinkan Anda mengedit peran AWSServiceRoleForS3StorageLens terkait layanan. Setelah membuat peran terkait layanan, Anda tidak dapat mengubah nama peran karena berbagai entitas mungkin mereferensikan peran tersebut. Namun, Anda dapat menyunting penjelasan peran menggunakan IAM. Untuk informasi selengkapnya, lihat Mengedit peran tertaut layanan dalam Panduan Pengguna IAM.

Menghapus peran terkait layanan untuk Lensa Penyimpanan Amazon S3

Jika Anda tidak perlu lagi menggunakan peran terkait layanan, kami rekomendasikan agar Anda menghapus peran tersebut. Dengan begitu, Anda tidak memiliki entitas yang tidak digunakan yang tidak dipantau atau dipelihara secara aktif. Tetapi, Anda harus membersihkan sumber daya peran terkait layanan sebelum menghapusnya secara manual.

> **Note**
>
> Jika layanan Lensa Penyimpanan Amazon S3 menggunakan peran tersebut ketika Anda mencoba menghapus sumber daya, penghapusan mungkin gagal. Jika hal itu terjadi, tunggu beberapa menit dan coba mengoperasikannya lagi.

Untuk menghapus AWSServiceRoleForS3StorageLens Anda harus menghapus semua konfigurasi Lensa Penyimpanan S3 tingkat organisasi yang ada di semua Wilayah AWS menggunakan AWS Organizations manajemen atau akun administrator delegasi.

Sumber daya adalah konfigurasi Lensa Penyimpanan S3 tingkat organisasi. Gunakan S3 Storage Lens untuk membersihkan sumber daya dan kemudian gunakan Konsol IAM, CLI, REST API, atau AWS SDK untuk menghapus peran.

Di REST API, AWS CLI, dan SDKs, konfigurasi Lensa Penyimpanan S3 dapat ditemukan menggunakan `ListStorageLensConfigurations` di semua Wilayah tempat organisasi Anda membuat konfigurasi Lensa Penyimpanan S3. Gunakan tindakan `DeleteStorageLensConfiguration` untuk menghapus konfigurasi ini sehingga Anda dapat menghapus peran.

> **Note**
>
> Untuk menghapus peran terkait layanan, Anda harus menghapus semua konfigurasi Lensa Penyimpanan S3 tingkat organisasi di semua Wilayah tempat mereka ada.

Untuk menghapus sumber daya Lensa Penyimpanan Amazon S3 yang digunakan oleh SLR AWSServiceRoleForS3StorageLens

1. Untuk mendapatkan daftar konfigurasi tingkat organisasi Anda, Anda harus menggunakan `ListStorageLensConfigurations` di setiap Wilayah yang memiliki konfigurasi Lensa Penyimpanan S3. Daftar ini juga dapat diperoleh dari konsol Amazon S3.
2. Hapus konfigurasi ini dari titik akhir Regional yang sesuai dengan menjalankan panggilan `DeleteStorageLensConfiguration` API atau dengan menggunakan konsol Amazon S3.

Untuk menghapus peran tertaut layanan secara manual menggunakan IAM

Setelah Anda menghapus konfigurasi, hapus AWSServiceRoleForS3StorageLens SLR dari Konsol IAM atau dengan menjalankan API IAM`DeleteServiceLinkedRole`, atau menggunakan atau SDK. AWS CLI AWS Untuk informasi selengkapnya, lihat Menghapus peran tertaut layanan dalam Panduan Pengguna IAM.

Wilayah yang Di Support untuk Peran Terkait Layanan Lensa Penyimpanan S3

S3 Storage Lens mendukung penggunaan peran terkait layanan di semua Wilayah AWS tempat layanan tersedia. Untuk informasi lebih lanjut, lihat Wilayah dan Titik Akhir Amazon S3.

Memecahkan masalah identitas dan akses Amazon S3

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan Amazon S3 dan IAM.

Topik

- Saya menerima kesalahan akses ditolak
- Saya tidak berwenang untuk melakukan tindakan di Amazon S3
- Saya tidak berwenang untuk melakukan iam: PassRole
- Saya ingin mengizinkan orang di luar saya Akun AWS mengakses sumber daya Amazon S3 saya
- Memecahkan masalah kesalahan akses ditolak (403 Forbidden) di Amazon S3

Saya menerima kesalahan akses ditolak

Verifikasi bahwa tidak ada `Deny` pernyataan eksplisit terhadap pemohon yang Anda coba berikan izin baik dalam kebijakan bucket atau kebijakan berbasis identitas.

Untuk informasi rinci tentang pemecahan masalah akses ditolak kesalahan, lihat. Memecahkan masalah kesalahan akses ditolak (403 Forbidden) di Amazon S3

Saya tidak berwenang untuk melakukan tindakan di Amazon S3

Jika Anda menerima pesan kesalahan bahwa Anda tidak memiliki otorisasi untuk melakukan tindakan, kebijakan Anda harus diperbarui agar Anda dapat melakukan tindakan tersebut.

Contoh kesalahan berikut terjadi ketika pengguna IAM `mateojackson` mencoba menggunakan konsol untuk melihat detail tentang suatu sumber daya *`my-example-widget`* rekaan, tetapi tidak memiliki izin `s3:`*`GetWidget`* rekaan.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 s3:GetWidget on resource: my-example-widget
```

Dalam hal ini, kebijakan untuk pengguna `mateojackson` harus diperbarui untuk mengizinkan akses ke sumber daya *`my-example-widget`* dengan menggunakan tindakan `s3:`*`GetWidget`*.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya tidak berwenang untuk melakukan iam: PassRole

Jika Anda menerima kesalahan bahwa Anda tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran ke Amazon S3.

Beberapa Layanan AWS memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi ketika pengguna IAM bernama `marymajor` mencoba menggunakan konsol untuk melakukan tindakan di Amazon S3. Namun, tindakan tersebut memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya ingin mengizinkan orang di luar saya Akun AWS mengakses sumber daya Amazon S3 saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACLs), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa referensi berikut:

- Untuk mempelajari apakah Amazon S3 mendukung fitur-fitur ini, lihat. Bagaimana Amazon S3 bekerja dengan IAM
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat Menyediakan akses ke pengguna IAM di pengguna lain Akun AWS yang Anda miliki di Panduan Pengguna IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda kepada pihak ketiga Akun AWS, lihat Menyediakan akses yang Akun AWS dimiliki oleh pihak ketiga dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat Menyediakan akses ke pengguna terautentikasi eksternal (federasi identitas) dalam Panduan Pengguna IAM.
- Untuk mempelajari perbedaan antara menggunakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Akses sumber daya lintas akun di IAM di Panduan Pengguna IAM.

Memecahkan masalah kesalahan akses ditolak (403 Forbidden) di Amazon S3

Kesalahan akses ditolak (HTTP`403 Forbidden`) muncul ketika AWS secara eksplisit atau implisit menolak permintaan otorisasi.

- Penolakan eksplisit terjadi ketika kebijakan berisi `Deny` pernyataan untuk tindakan tertentu AWS .
- Penolakan implisit terjadi ketika tidak ada pernyataan yang berlaku dan juga tidak ada `Deny` pernyataan yang berlaku`Allow`.

Karena kebijakan AWS Identity and Access Management (IAM) secara implisit menyangkal prinsip IAM secara default, kebijakan tersebut harus secara eksplisit mengizinkan prinsipal untuk melakukan suatu tindakan. Jika tidak, kebijakan tersebut secara implisit menolak akses. Untuk informasi selengkapnya, lihat Perbedaan antara penolakan eksplisit dan implisit dalam Panduan Pengguna IAM. Untuk informasi tentang logika evaluasi kebijakan yang menentukan apakah permintaan akses diizinkan atau ditolak, lihat Logika evaluasi kebijakan di Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang izin operasi S3 API menurut jenis sumber daya S3, lihat. Izin yang diperlukan untuk operasi API Amazon S3

Topik berikut mencakup penyebab paling umum kesalahan akses ditolak di Amazon S3.

> **Note**
>
> Untuk kesalahan akses ditolak (HTTP`403 Forbidden`), Amazon S3 tidak membebankan biaya kepada pemilik bucket saat permintaan dimulai di luar AWS akun individu pemilik bucket atau organisasi pemilik bucket. AWS

Topik

- Akses contoh pesan yang ditolak dan cara memecahkan masalah
- Kebijakan bucket dan kebijakan IAM
- Pengaturan Amazon S3 ACL
- Pengaturan S3 Block Public Access
- Pengaturan enkripsi Amazon S3
- Pengaturan Kunci Objek S3
- Kebijakan VPC endpoint
- AWS Organizations kebijakan
- CloudFront akses distribusi
- Pengaturan titik akses
- Sumber daya tambahan

> **Note**
>
> Jika Anda mencoba memecahkan masalah izin, mulailah dengan the section called “Akses contoh pesan yang ditolak dan cara memecahkan masalah” bagian tersebut, lalu buka

bagian tersebut. ??? Pastikan juga untuk mengikuti panduan dithe section called “Kiat untuk memeriksa izin”.

Akses contoh pesan yang ditolak dan cara memecahkan masalah

> **Important**
>
> Dalam beberapa minggu mendatang, mulai 16 Juni 2025, Amazon S3 akan mulai menyediakan pesan kesalahan yang ditolak akses yang ditingkatkan untuk permintaan yang dibuat ke sumber daya di akun dalam organisasi yang sama di. AWS Organizations Pesan-pesan ini akan tersedia di semua Wilayah AWS, termasuk AWS GovCloud (US) Regions dan Wilayah Tiongkok.

Amazon S3 sekarang menyertakan konteks tambahan dalam kesalahan akses ditolak (HTTP`403 Forbidden`) untuk permintaan yang dibuat ke sumber daya dalam organisasi yang sama Akun AWS atau sama di. AWS Organizations Konteks baru ini mencakup jenis kebijakan yang menolak akses, alasan penolakan, dan informasi tentang pengguna IAM atau peran yang meminta akses ke sumber daya.

Konteks tambahan ini membantu Anda memecahkan masalah akses, mengidentifikasi akar penyebab kesalahan akses ditolak, dan memperbaiki kontrol akses yang salah dengan memperbarui kebijakan yang relevan. Konteks tambahan ini juga tersedia di AWS CloudTrail log. Pesan kesalahan yang ditolak akses yang ditingkatkan untuk permintaan akun yang sama atau organisasi yang sama sekarang tersedia di semua Wilayah AWS, termasuk Wilayah dan Wilayah Tiongkok AWS GovCloud (US) Regions .

Sebagian besar pesan kesalahan akses ditolak muncul dalam format`User` *`user-arn`* `is not authorized to perform` *`action`* `on "`*`resource-arn`*`" because` *`context`*. Dalam contoh ini, *`user-arn`* adalah Nama Sumber Daya Amazon (ARN) pengguna yang tidak menerima akses, *`action`* adalah tindakan layanan yang ditolak kebijakan, dan *`resource-arn`* merupakan ARN sumber daya tempat kebijakan tersebut bertindak. *`context`*Kolom ini mewakili konteks tambahan tentang jenis kebijakan yang menjelaskan mengapa kebijakan menolak akses.

Jika kebijakan secara eksplisit menolak akses karena kebijakan berisi `Deny` pernyataan, maka pesan kesalahan akses ditolak menyertakan frasa. `with an explicit deny in a` *`type`* `policy`

Ketika kebijakan secara implisit menolak akses, maka pesan kesalahan akses ditolak menyertakan frasa. `because no type policy allows the action action`

Important

- Pesan yang ditolak akses yang disempurnakan dikembalikan hanya untuk permintaan akun yang sama atau untuk permintaan dalam organisasi yang sama di. AWS Organizations Permintaan lintas akun di luar organisasi yang sama menampilkan pesan umum`Access Denied`.

  Untuk informasi tentang logika evaluasi kebijakan yang menentukan apakah permintaan akses lintas akun diizinkan atau ditolak, lihat Logika evaluasi kebijakan lintas akun di Panduan Pengguna IAM. Untuk panduan yang menunjukkan cara memberikan akses lintas akun, lihat. the section called “memberikan izin lintas akun”
- Untuk permintaan dalam organisasi yang sama di AWS Organizations:
  - Pesan yang ditolak akses yang disempurnakan tidak dikembalikan jika penolakan terjadi karena kebijakan titik akhir virtual private cloud (VPC).
  - Pesan yang ditolak akses yang disempurnakan disediakan setiap kali pemilik bucket dan akun penelepon milik organisasi yang sama. AWS Organizations Meskipun bucket yang dikonfigurasi dengan pemilik Object Ownership Bucket S3 lebih disukai atau pengaturan Object writer mungkin berisi objek yang dimiliki oleh akun yang berbeda, kepemilikan objek tidak memengaruhi peningkatan akses pesan yang ditolak. Pesan yang ditolak akses yang disempurnakan dikembalikan untuk semua permintaan objek selama pemilik bucket dan penelepon berada di organisasi yang sama, terlepas dari siapa yang memiliki objek tertentu. Untuk informasi tentang setelan dan konfigurasi Kepemilikan Objek, lihatthe section called “Mengontrol kepemilikan objek”.
- Pesan kesalahan ditolak akses yang disempurnakan tidak dikembalikan untuk permintaan yang dibuat ke bucket direktori. Permintaan bucket direktori mengembalikan `Access Denied` pesan umum.
- Jika beberapa kebijakan dari jenis kebijakan yang sama menolak permintaan otorisasi, pesan kesalahan akses ditolak tidak menentukan jumlah kebijakan.
- Jika beberapa jenis kebijakan menolak permintaan otorisasi, pesan kesalahan hanya menyertakan salah satu jenis kebijakan tersebut.
- Jika permintaan akses ditolak karena beberapa alasan, pesan kesalahan hanya mencakup salah satu alasan penolakan.

Contoh berikut menunjukkan format untuk berbagai jenis pesan kesalahan yang ditolak akses dan cara memecahkan masalah setiap jenis pesan.

Akses ditolak karena kebijakan kontrol sumber daya — penolakan eksplisit

1. Periksa `Deny` pernyataan untuk tindakan dalam kebijakan kontrol sumber daya Anda (RCPs). Untuk contoh berikut, tindakannya adalah`s3:GetObject`.
2. Perbarui RCP Anda dengan menghapus `Deny` pernyataan. Untuk informasi selengkapnya, lihat Memperbarui kebijakan kontrol sumber daya (RCP) di Panduan AWS Organizations Pengguna.

```
An error occurred (AccessDenied) when calling the GetObject operation:
User: arn:aws:iam::777788889999:user/MaryMajor is not authorized to perform:
s3:GetObject on resource: "arn:aws:s3:::amzn-s3-demo-bucket1/object-name"
with an explicit deny in a resource control policy
```

Akses ditolak karena Kebijakan Kontrol Layanan — penolakan implisit

1. Periksa `Allow` pernyataan yang hilang untuk tindakan dalam kebijakan kontrol layanan Anda (SCPs). Untuk contoh berikut, tindakannya adalah`s3:GetObject`.
2. Perbarui SCP Anda dengan menambahkan `Allow` pernyataan. Untuk informasi selengkapnya, lihat Memperbarui SCP di Panduan AWS Organizations Pengguna.

```
User: arn:aws:iam::777788889999:user/MaryMajor is not authorized to perform:
s3:GetObject because no service control policy allows the s3:GetObject action
```

Akses ditolak karena Kebijakan Kontrol Layanan — penolakan eksplisit

1. Periksa `Deny` pernyataan untuk tindakan dalam Kebijakan Kontrol Layanan Anda (SCPs). Untuk contoh berikut, tindakannya adalah`s3:GetObject`.
2. Perbarui SCP Anda dengan mengubah `Deny` pernyataan untuk memungkinkan pengguna mengakses yang diperlukan. Untuk contoh bagaimana Anda dapat melakukannya, lihat Mencegah pengguna dan peran IAM membuat perubahan tertentu, dengan pengecualian untuk peran admin yang ditentukan dalam Panduan AWS Organizations Pengguna. Untuk informasi selengkapnya tentang memperbarui SCP Anda, lihat Memperbarui SCP di AWS Organizations Panduan Pengguna.

```
User: arn:aws:iam::777788889999:user/MaryMajor is not authorized to perform:
s3:GetObject with an explicit deny in a service control policy
```

Akses ditolak karena kebijakan titik akhir VPC — penolakan implisit

1. Periksa `Allow` pernyataan yang hilang untuk tindakan dalam kebijakan titik akhir virtual private cloud (VPC) Anda. Untuk contoh berikut, tindakannya adalah`s3:GetObject`.
2. Perbarui kebijakan titik akhir VPC Anda dengan menambahkan pernyataan. `Allow` Untuk informasi selengkapnya, lihat Memperbarui kebijakan titik akhir VPC di Panduan.AWS PrivateLink

```
User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
s3:GetObject because no VPC endpoint policy allows the s3:GetObject action
```

Akses ditolak karena kebijakan titik akhir VPC — penolakan eksplisit

1. Periksa `Deny` pernyataan eksplisit untuk tindakan dalam kebijakan titik akhir virtual private cloud (VPC) Anda. Untuk contoh berikut, tindakannya adalah`s3:GetObject`.
2. Perbarui kebijakan titik akhir VPC Anda dengan mengubah `Deny` pernyataan untuk memungkinkan pengguna mengakses yang diperlukan. Misalnya, Anda dapat memperbarui `Deny` pernyataan Anda untuk menggunakan kunci `aws:PrincipalAccount` kondisi dengan operator `StringNotEquals` kondisi untuk mengizinkan akses utama tertentu, seperti yang ditunjukkan padathe section called “Contoh 7: Mengecualikan prinsipal dari pernyataan Deny”. Untuk informasi selengkapnya tentang memperbarui kebijakan titik akhir VPC Anda, lihat Memperbarui kebijakan titik akhir VPC di Panduan.AWS PrivateLink

```
User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
s3:GetObject on resource: "arn:aws:s3:::amzn-s3-demo-bucket1/object-name" with
an explicit deny in a VPC endpoint policy
```

Akses ditolak karena batas izin — penolakan implisit

1. Periksa `Allow` pernyataan yang hilang untuk tindakan di batas izin Anda. Untuk contoh berikut, tindakannya adalah`s3:GetObject`.
2. Perbarui batas izin Anda dengan menambahkan `Allow` pernyataan ke kebijakan IAM Anda. Untuk informasi selengkapnya, lihat Batas izin untuk entitas IAM dan Mengedit kebijakan IAM di Panduan Pengguna IAM.

```
User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
s3:GetObject on resource: "arn:aws:s3:::amzn-s3-demo-bucket1/object-name"
because no permissions boundary allows the s3:GetObject action
```

Akses ditolak karena batas izin — penolakan eksplisit

1. Periksa `Deny` pernyataan eksplisit untuk tindakan di batas izin Anda. Untuk contoh berikut, tindakannya adalah`s3:GetObject`.
2. Perbarui batas izin Anda dengan mengubah `Deny` pernyataan dalam kebijakan IAM Anda untuk memungkinkan pengguna mengakses yang diperlukan. Misalnya, Anda dapat memperbarui `Deny` pernyataan Anda untuk menggunakan kunci `aws:PrincipalAccount` kondisi dengan operator `StringNotEquals` kondisi untuk mengizinkan akses utama tertentu, seperti yang ditunjukkan aws:PrincipalAccountdalam Panduan Pengguna IAM. Untuk informasi selengkapnya, lihat Batas izin untuk entitas IAM dan Mengedit kebijakan IAM di Panduan Pengguna IAM.

```
User: arn:aws:iam::777788889999:user/MaryMajor is not authorized to perform:
s3:GetObject with an explicit deny in a permissions boundary
```

Akses ditolak karena kebijakan sesi — penolakan implisit

1. Periksa `Allow` pernyataan yang hilang untuk tindakan dalam kebijakan sesi Anda. Untuk contoh berikut, tindakannya adalah`s3:GetObject`.
2. Perbarui kebijakan sesi Anda dengan menambahkan `Allow` pernyataan. Untuk informasi selengkapnya, lihat Kebijakan sesi dan Mengedit kebijakan IAM di Panduan Pengguna IAM.

```
User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
s3:GetObject because no session policy allows the s3:GetObject action
```

Akses ditolak karena kebijakan sesi — penolakan eksplisit

1. Periksa `Deny` pernyataan eksplisit untuk tindakan dalam kebijakan sesi Anda. Untuk contoh berikut, tindakannya adalah`s3:GetObject`.
2. Perbarui kebijakan sesi Anda dengan mengubah `Deny` pernyataan untuk memungkinkan pengguna mengakses yang diperlukan. Misalnya, Anda dapat memperbarui `Deny` pernyataan Anda untuk menggunakan kunci `aws:PrincipalAccount` kondisi dengan operator `StringNotEquals` kondisi untuk mengizinkan akses utama tertentu, seperti yang ditunjukkan

padathe section called “Contoh 7: Mengecualikan prinsipal dari pernyataan Deny”. Untuk informasi selengkapnya tentang memperbarui kebijakan sesi Anda, lihat Kebijakan sesi dan Mengedit kebijakan IAM di Panduan Pengguna IAM.

```
User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
s3:GetObject on resource: "arn:aws:s3:::amzn-s3-demo-bucket1/object-name" with
an explicit deny in a session policy
```

Akses ditolak karena kebijakan berbasis sumber daya — penolakan implisit

> **Note**
>
> Kebijakan berbasis sumber daya berarti kebijakan seperti kebijakan bucket dan kebijakan jalur akses.

1. Periksa `Allow` pernyataan yang hilang untuk tindakan dalam kebijakan berbasis sumber daya Anda. Periksa juga apakah pengaturan `IgnorePublicAcls` S3 Block Public Access diterapkan pada bucket, access point, atau level akun. Untuk contoh berikut, tindakannya adalah`s3:GetObject`.
2. Perbarui kebijakan Anda dengan menambahkan `Allow` pernyataan. Untuk informasi selengkapnya, lihat Kebijakan berbasis sumber daya dan Mengedit kebijakan IAM di Panduan Pengguna IAM.

   Anda mungkin juga perlu menyesuaikan pengaturan `IgnorePublicAcls` blokir akses publik untuk bucket, titik akses, atau akun. Untuk informasi selengkapnya, lihat the section called “Akses ditolak karena pengaturan Blokir Akses Publik” dan the section called “Mengonfigurasi pengaturan bucket dan titik akses”.

```
User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
s3:GetObject because no resource-based policy allows the s3:GetObject action
```

Akses ditolak karena kebijakan berbasis sumber daya — penolakan eksplisit

> **Note**
>
> Kebijakan berbasis sumber daya berarti kebijakan seperti kebijakan bucket dan kebijakan jalur akses.

1. Periksa `Deny` pernyataan eksplisit untuk tindakan dalam kebijakan berbasis sumber daya Anda. Periksa juga apakah pengaturan `RestrictPublicBuckets` S3 Block Public Access diterapkan pada bucket, access point, atau level akun. Untuk contoh berikut, tindakannya adalah`s3:GetObject`.
2. Perbarui kebijakan Anda dengan mengubah `Deny` pernyataan untuk memungkinkan pengguna mengakses yang diperlukan. Misalnya, Anda dapat memperbarui `Deny` pernyataan Anda untuk menggunakan kunci `aws:PrincipalAccount` kondisi dengan operator `StringNotEquals` kondisi untuk mengizinkan akses utama tertentu, seperti yang ditunjukkan padathe section called “Contoh 7: Mengecualikan prinsipal dari pernyataan `Deny`”. Untuk informasi selengkapnya tentang memperbarui kebijakan berbasis sumber daya, lihat Kebijakan berbasis sumber daya dan Mengedit kebijakan IAM di Panduan Pengguna IAM.

   Anda mungkin juga perlu menyesuaikan pengaturan `RestrictPublicBuckets` blokir akses publik untuk bucket, titik akses, atau akun. Untuk informasi selengkapnya, lihat the section called “Akses ditolak karena pengaturan Blokir Akses Publik” dan the section called “Mengonfigurasi pengaturan bucket dan titik akses”.

```
User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
s3:GetObject on resource: "arn:aws:s3:::amzn-s3-demo-bucket1/object-name" with
an explicit deny in a resource-based policy
```

Akses ditolak karena kebijakan berbasis identitas — penolakan implisit

1. Periksa `Allow` pernyataan yang hilang untuk tindakan dalam kebijakan berbasis identitas yang dilampirkan pada identitas. Untuk contoh berikut, tindakan `s3:GetObject` dilampirkan ke penggunaMaryMajor.
2. Perbarui kebijakan Anda dengan menambahkan `Allow` pernyataan. Untuk informasi selengkapnya, lihat Kebijakan berbasis identitas dan Mengedit kebijakan IAM di Panduan Pengguna IAM.

```
User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
s3:GetObject because no identity-based policy allows the s3:GetObject action
```

Akses ditolak karena kebijakan berbasis identitas — penolakan eksplisit

1. Periksa `Deny` pernyataan eksplisit untuk tindakan dalam kebijakan berbasis identitas yang dilampirkan pada identitas. Untuk contoh berikut, tindakan `s3:GetObject` dilampirkan ke penggunaMaryMajor.
2. Perbarui kebijakan Anda dengan mengubah `Deny` pernyataan untuk memungkinkan pengguna mengakses yang diperlukan. Misalnya, Anda dapat memperbarui `Deny` pernyataan Anda untuk menggunakan kunci `aws:PrincipalAccount` kondisi dengan operator `StringNotEquals` kondisi untuk mengizinkan akses utama tertentu, seperti yang ditunjukkan aws:PrincipalAccountdalam Panduan Pengguna IAM. Untuk informasi selengkapnya, lihat Kebijakan berbasis identitas dan Mengedit kebijakan IAM di Panduan Pengguna IAM.

```
User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
s3:GetObject on resource: "arn:aws:s3:::amzn-s3-demo-bucket1/object-name" with
an explicit deny in an identity-based policy
```

Akses ditolak karena pengaturan Blokir Akses Publik

Fitur Blokir Akses Publik Amazon S3 menyediakan pengaturan untuk titik akses, bucket, dan akun untuk membantu Anda mengelola akses publik ke sumber daya Amazon S3. Untuk informasi selengkapnya tentang cara Amazon S3 mendefinisikan “publik,” lihat Arti “publik”.

Secara bawaan, bucket baru, titik akses, dan objek baru tidak mengizinkan akses publik. Namun, pengguna dapat mengubah kebijakan bucket, kebijakan titik akses, kebijakan pengguna IAM, izin objek, atau daftar kontrol akses (ACLs) untuk mengizinkan akses publik. Setelan Akses Publik Blok S3 mengesampingkan kebijakan, izin, dan. ACLs Sejak April 2023, semua pengaturan Blokir Akses Publik diaktifkan secara default untuk bucket baru.

Saat Amazon S3 menerima permintaan untuk mengakses bucket atau objek, itu akan menentukan apakah akun dari bucket atau pemilik bucket menerapkan blokir akses publik. Jika permintaan tersebut dibuat melalui titik akses, maka Amazon S3 juga memeriksa pengaturan blokir akses publik untuk titik akses tersebut. Jika ada pengaturan blokir akses publik yang sudah ada yang melarang akses yang diminta tersebut, maka Amazon S3 akan menolak permintaan tersebut.

Blokir Akses Publik Amazon S3 menyediakan empat pengaturan. Pengaturan ini bersifat independen dan dapat digunakan dalam kombinasi apa pun. Setiap pengaturan dapat diterapkan ke titik akses, ember, atau seluruh AWS akun. Jika pengaturan blokir akses publik untuk titik akses, bucket, atau akun berbeda, maka Amazon S3 akan menerapkan kombinasi yang paling ketat dari pengaturan titik akses, bucket, dan akun.

Ketika Amazon S3 mengevaluasi apakah suatu operasi dilarang oleh pengaturan blokir akses publik, itu akan menolak permintaan apa pun yang melanggar titik akses, bucket, atau pengaturan akun.

Empat pengaturan yang disediakan oleh Amazon S3 Block Public Access adalah sebagai berikut:

- `BlockPublicAcls`— Pengaturan ini berlaku untuk`PutBucketAcl`,`PutObjectAcl`,`PutObject`,`CreateBucket`,`CopyObject`, dan `POST Object` permintaan. `BlockPublicAcls`Pengaturan menyebabkan perilaku berikut:
  - `PutBucketAcl`dan `PutObjectAcl` panggilan gagal jika daftar kontrol akses yang ditentukan (ACL) bersifat publik.
  - `PutObject`panggilan gagal jika permintaan menyertakan ACL publik.
  - Jika pengaturan ini diterapkan ke akun, maka `CreateBucket` panggilan gagal dengan respons HTTP `400` (`Bad Request`) jika permintaan menyertakan ACL publik.

  Misalnya, ketika akses ditolak untuk `CopyObject` permintaan karena `BlockPublicAcls` pengaturan, Anda menerima pesan berikut:

  ```
  An error occurred (AccessDenied) when calling the CopyObject operation:
  User: arn:aws:sts::123456789012:user/MaryMajor is not authorized to
  perform: s3:CopyObject on resource: "arn:aws:s3:::amzn-s3-demo-bucket1/object-name"
  because public access control lists (ACLs) are blocked by the BlockPublicAcls block
  public access setting.
  ```

- `IgnorePublicAcls`— `IgnorePublicAcls` Pengaturan menyebabkan Amazon S3 mengabaikan semua publik ACLs di ember dan objek apa pun yang dikandungnya. Jika izin permintaan Anda hanya diberikan oleh ACL publik, maka pengaturan `IgnorePublicAcls` menolak permintaan tersebut.

  Setiap penolakan yang dihasilkan dari `IgnorePublicAcls` pengaturan adalah implisit. Misalnya, jika `IgnorePublicAcls` menolak `GetObject` permintaan karena ACL publik, Anda menerima pesan berikut:

  ```
  User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
  ```

```
s3:GetObject because no resource-based policy allows the s3:GetObject action
```

- `BlockPublicPolicy`— Pengaturan ini berlaku untuk `PutBucketPolicy` dan `PutAccessPointPolicy` permintaan.

  Menyetel `BlockPublicPolicy` bucket menyebabkan Amazon S3 menolak panggilan `PutBucketPolicy` jika kebijakan bucket yang ditentukan mengizinkan akses publik. Pengaturan ini juga menyebabkan Amazon S3 menolak panggilan ke `PutAccessPointPolicy` semua titik akses akun yang sama pada bucket jika kebijakan yang ditentukan mengizinkan akses publik.

  Pengaturan `BlockPublicPolicy` untuk titik akses menyebabkan Amazon S3 menolak panggilan ke `PutAccessPointPolicy` dan `PutBucketPolicy` yang dilakukan melalui jalur akses jika kebijakan yang ditentukan (baik untuk titik akses atau bucket yang mendasarinya) mengizinkan akses publik.

  Misalnya, ketika akses ditolak atas `PutBucketPolicy` permintaan karena `BlockPublicPolicy` pengaturan, Anda menerima pesan berikut:

  ```
  An error occurred (AccessDenied) when calling the PutBucketPolicy operation:
  User: arn:aws:sts::123456789012:user/MaryMajor is not authorized to
  perform: s3:PutBucketPolicy on resource: "arn:aws:s3:::amzn-s3-demo-bucket1/object-
  name"
  because public policies are blocked by the BlockPublicPolicy block public
  access setting.
  ```

- `RestrictPublicBucketsRestrictPublicBuckets`Pengaturan membatasi akses ke titik akses atau bucket dengan kebijakan publik hanya untuk Layanan AWS prinsipal dan pengguna yang berwenang dalam akun pemilik bucket dan akun pemilik jalur akses. Pengaturan ini memblokir semua akses lintas akun ke titik akses atau bucket (kecuali oleh Layanan AWS prinsipal), sambil tetap memungkinkan pengguna di dalam akun untuk mengelola titik akses atau bucket. Pengaturan ini juga menolak semua panggilan anonim (atau tidak ditandatangani).

  Setiap penolakan yang dihasilkan dari `RestrictPublicBuckets` pengaturan adalah eksplisit. Misalnya, jika `RestrictPublicBuckets` menolak `GetObject` permintaan karena kebijakan bucket publik atau jalur akses, Anda menerima pesan berikut:

  ```
  User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
  s3:GetObject on resource: "arn:aws:s3:::amzn-s3-demo-bucket1/object-name" with
  an explicit deny in a resource-based policy
  ```

Untuk informasi selengkapnya tentang pengaturan ini, lihat the section called “Pengaturan blokir akses publik”. Untuk meninjau dan memperbarui pengaturan ini, lihatthe section called “Mengonfigurasi blokir akses publik”.

Kebijakan bucket dan kebijakan IAM

Operasi tingkat bucket

Jika tidak ada kebijakan bucket, maka bucket secara implisit mengizinkan permintaan dari identitas AWS Identity and Access Management (IAM) apa pun di akun pemilik ember. Bucket juga secara implisit menolak permintaan dari identitas IAM lainnya dari akun lain, serta menolak permintaan anonim (tidak ditandatangani). Namun, jika tidak ada kebijakan pengguna IAM, pemohon (kecuali mereka adalah pengguna Akun AWS root) secara implisit ditolak untuk membuat permintaan apa pun. Untuk informasi selengkapnya tentang logika evaluasi ini, lihat Menentukan apakah permintaan ditolak atau diizinkan dalam akun di Panduan Pengguna IAM.

Operasi tingkat objek

Jika objek dimiliki oleh akun pemilik bucket, kebijakan bucket dan kebijakan pengguna IAM akan berfungsi dengan cara yang sama untuk operasi tingkat objek seperti yang mereka lakukan untuk operasi tingkat bucket. Misalnya, jika tidak ada kebijakan bucket, bucket secara implisit mengizinkan permintaan objek dari identitas IAM apa pun di akun pemilik ember. Bucket juga secara implisit menolak permintaan objek dari identitas IAM lainnya dari akun lain, serta menolak permintaan anonim (tidak ditandatangani). Namun, jika tidak ada kebijakan pengguna IAM, pemohon (kecuali mereka adalah pengguna Akun AWS root) secara implisit ditolak untuk membuat permintaan objek apa pun.

Jika objek dimiliki oleh akun eksternal, maka akses ke objek hanya dapat diberikan melalui daftar kontrol akses objek (ACLs). Kebijakan bucket dan kebijakan pengguna IAM masih dapat digunakan untuk menolak permintaan objek.

Oleh karena itu, untuk memastikan kebijakan bucket atau kebijakan pengguna IAM Anda tidak menyebabkan kesalahan Access Denied (403 Forbidden), pastikan persyaratan berikut terpenuhi:

- Untuk akses akun yang sama, tidak boleh ada pernyataan `Deny` secara eksplisit terhadap peminta yang ingin Anda berikan izin, baik dalam kebijakan bucket maupun kebijakan pengguna IAM. Jika Anda ingin memberikan izin hanya dengan menggunakan kebijakan bucket dan kebijakan pengguna IAM, setidaknya harus ada satu pernyataan `Allow` eksplisit di salah satu kebijakan ini.
- Untuk akses lintas akun, tidak boleh ada `Deny` pernyataan eksplisit terhadap pemohon yang Anda coba berikan izin, baik dalam kebijakan bucket atau kebijakan pengguna IAM. Untuk memberikan

izin lintas akun hanya dengan menggunakan kebijakan bucket dan kebijakan pengguna IAM, pastikan bahwa kebijakan bucket dan kebijakan pengguna IAM dari pemohon menyertakan pernyataan eksplisit. `Allow`

> **Note**
>
> Pernyataan `Allow` dalam kebijakan bucket hanya berlaku untuk objek yang dimiliki oleh akun pemilik bucket yang sama. Namun, pernyataan `Deny` dalam kebijakan bucket berlaku untuk semua objek terlepas dari kepemilikan objek.

Untuk meninjau atau mengedit kebijakan bucket

> **Note**
>
> Untuk melihat atau mengedit kebijakan bucket, Anda harus memiliki izin `s3:GetBucketPolicy`.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Dari daftar Bucket, pilih nama bucket yang ingin Anda lihat atau edit kebijakan bucket-nya.
4. Pilih tab Izin.
5. Di Bawah Kebijakan bucket, pilih Edit. Halaman Edit kebijakan bucket akan muncul.

Untuk meninjau atau mengedit kebijakan bucket Anda menggunakan AWS Command Line Interface (AWS CLI), gunakan get-bucket-policyperintah.

> **Note**
>
> Jika Anda terkunci dari bucket karena kebijakan bucket yang salah, masuk ke AWS Management Console dengan menggunakan kredensi pengguna Akun AWS root Anda. Untuk mendapatkan kembali akses ke bucket Anda, pastikan untuk menghapus kebijakan bucket yang salah dengan menggunakan kredensi pengguna Akun AWS root Anda.

Kiat untuk memeriksa izin

Untuk memeriksa apakah peminta memiliki izin yang tepat dalam melakukan operasi Amazon S3, coba hal berikut ini:

- Identifikasi peminta. Jika permintaan tidak ditandatangani, berarti permintaan tersebut merupakan permintaan anonim tanpa kebijakan pengguna IAM. Jika itu adalah permintaan yang menggunakan URL yang telah ditetapkan sebelumnya, maka kebijakan pengguna sama dengan yang digunakan untuk pengguna IAM atau peran yang menandatangani permintaan tersebut.
- Pastikan Anda menggunakan peran atau pengguna IAM yang benar. Anda dapat memverifikasi peran atau pengguna IAM Anda dengan memeriksa sudut kanan atas AWS Management Console atau dengan menggunakan perintah aws sts get-caller-identity.
- Periksa kebijakan IAM yang terkait dengan peran atau pengguna IAM. Gunakan salah satu metode berikut:
  - Uji kebijakan IAM dengan simulator kebijakan IAM.
  - Tinjau berbagai jenis kebijakan IAM yang berbeda.
- Jika perlu, edit kebijakan pengguna IAM Anda.
- Tinjau contoh kebijakan berikut yang secara eksplisit menolak atau mengizinkan akses:
  - Kebijakan mengizinkan pengguna IAM eksplisit: IAM: Mengizinkan dan menolak akses ke beberapa layanan secara terprogram dan di konsol
  - Kebijakan izin bucket eksplisit: Memberikan izin ke beberapa akun untuk mengunggah objek atau menyetel objek ACLs untuk akses publik
  - Tolak kebijakan pengguna IAM secara eksplisit AWS:: Menolak akses AWS berdasarkan permintaan Wilayah AWS
  - Kebijakan bucket menolak eksplisit: Memerlukan SSE-KMS untuk semua objek yang ditulis ke bucket

Pengaturan Amazon S3 ACL

Saat memeriksa pengaturan ACL Anda, pertama-tama tinjau setelan Kepemilikan Objek Anda untuk memeriksa ACLs apakah diaktifkan di bucket. Ketahuilah bahwa izin ACL hanya dapat digunakan untuk memberikan izin dan tidak dapat digunakan untuk menolak permintaan. ACLs juga tidak dapat digunakan untuk memberikan akses ke pemohon yang ditolak oleh penolakan eksplisit dalam kebijakan bucket atau kebijakan pengguna IAM.

Pengaturan Kepemilikan Objek disetel ke pemilik Bucket diberlakukan

Jika pengaturan yang diberlakukan pemilik Bucket diaktifkan, setelan ACL tidak akan menyebabkan kesalahan Access Denied (403 Forbidden) karena setelan ini menonaktifkan semua ACLs yang berlaku untuk bucket dan objek. Pemilik bucket diberlakukan adalah setelan default (dan disarankan) untuk bucket Amazon S3.

Pengaturan Kepemilikan Objek diatur ke pilihan pemilik Bucket atau penulis Object

Izin ACL masih berlaku dengan setelan pilihan pemilik Bucket atau setelan penulis Object. Ada dua jenis ACLs: ember ACLs dan objek ACLs. Untuk perbedaan antara kedua jenis ini ACLs, lihat Pemetaan izin ACL dan izin kebijakan akses.

Bergantung pada tindakan permintaan yang ditolak, periksa izin ACL untuk bucket atau objek:

- Jika Amazon S3 menolak `LIST`, objek `PUT`, `GetBucketAcl`, atau permintaan `PutBucketAcl`, tinjau izin ACL untuk bucket Anda.

  > **Note**
  >
  > Anda tidak dapat memberikan izin `GET` objek dengan setelan ACL bucket.

- Jika Amazon S3 menolak permintaan `GET` pada objek S3, atau permintaan PutObjectAcl, maka tinjau izin ACL untuk objek tersebut.

  > **Important**
  >
  > Jika akun pemilik objek berbeda dari akun pemilik bucket, maka akses ke objek tidak dikontrol oleh kebijakan bucket.

Memecahkan masalah kesalahan Akses Ditolak (403 Forbidden) dari permintaan objek **`GET`** selama kepemilikan objek lintas akun

Tinjau pengaturan Kepemilikan Objek bucket untuk menentukan pemilik objek. Jika Anda memiliki akses ke objek ACLs, maka Anda juga dapat memeriksa akun pemilik objek. (Untuk melihat akun pemilik objek, tinjau pengaturan ACL objek di konsol Amazon S3.) Atau, Anda juga dapat membuat permintaan `GetObjectAcl` untuk menemukan ID kanonik pemilik objek untuk memverifikasi akun pemilik objek. Secara default, ACLs berikan izin izin eksplisit untuk `GET` permintaan ke akun pemilik objek.

Setelah Anda mengonfirmasi bahwa pemilik objek berbeda dari pemilik bucket, lalu tergantung pada kasus penggunaan dan tingkat akses Anda, pilih salah satu metode berikut untuk membantu mengatasi kesalahan Akses Ditolak (403 Forbidden):

- Nonaktifkan ACLs (disarankan) — Metode ini akan berlaku untuk semua objek dan dapat dilakukan oleh pemilik bucket. Metode ini secara otomatis memberikan pemilik bucket kepemilikan dan kontrol penuh atas setiap objek di bucket. Sebelum Anda menerapkan metode ini, periksa prasyarat untuk menonaktifkan. ACLs Untuk informasi tentang cara menyetel bucket ke mode diberlakukan (disarankan) pemilik Bucket, lihat Menyetel Kepemilikan Objek pada bucket yang ada.

  > **Important**
  >
  > Untuk mencegah error Access Denied (403 Forbidden), pastikan untuk memigrasikan izin ACL ke kebijakan bucket sebelum Anda menonaktifkannya. ACLs Untuk informasi selengkapnya, lihat Contoh kebijakan bucket untuk migrasi dari izin ACL.

- Ubah pemilik objek menjadi pemilik bucket–Metode ini dapat diterapkan pada objek individual, tetapi hanya pemilik objek (atau pengguna dengan izin yang sesuai) yang dapat mengubah kepemilikan objek. PUTBiaya tambahan mungkin berlaku. (Untuk informasi selengkapnya, lihat harga Amazon S3.) Metode ini memberikan pemilik bucket kepemilikan penuh atas objek, memungkinkan pemilik bucket untuk mengontrol akses ke objek melalui kebijakan bucket.

  Untuk mengubah kepemilikan objek, lakukan salah satu hal berikut:

  - Anda (pemilik bucket) dapat menyalin objek kembali ke bucket.
  - Anda dapat mengubah setelan Kepemilikan Objek bucket menjadi pilihan pemilik Bucket. Jika Penentuan Versi dinonaktifkan, objek dalam bucket akan ditimpa. Jika Penentuan Versi diaktifkan, versi duplikat dari objek yang sama akan muncul di bucket, dan pemilik bucket dapat menetapkan aturan siklus hidup yang akan diakhiri. Untuk petunjuk tentang cara mengubah pengaturan Kepemilikan Objek Anda, lihatMenyetel Kepemilikan Objek pada bucket yang ada.

    > **Note**
    >
    > Saat Anda memperbarui setelan Kepemilikan Objek ke pemilik Bucket yang diinginkan, pengaturan hanya diterapkan ke objek baru yang diunggah ke bucket.

  - Anda dapat meminta pemilik objek mengunggah kembali objek dengan objek ACL terekam `bucket-owner-full-control`.

> **Note**
>
> Untuk unggahan lintas akun, Anda juga dapat meminta ACL objek terekam `bucket-owner-full-control` dalam kebijakan bucket Anda. Untuk contoh kebijakan bucket, lihat Memberikan izin lintas akun untuk mengunggah objek sekaligus memastikan bahwa pemilik bucket memiliki kontrol penuh.

- Simpan penulis objek sebagai pemilik objek-Metode ini tidak mengubah pemilik objek, tetapi memungkinkan Anda untuk memberikan akses ke objek satu per satu. Untuk memberikan akses ke objek, Anda harus memiliki izin `PutObjectAcl` untuk objek tersebut. Kemudian, untuk memperbaiki kesalahan Access Denied (403 Forbidden), tambahkan pemohon sebagai penerima hibah untuk mengakses objek di objek. ACLs Untuk informasi selengkapnya, lihat Mengkonfigurasi ACLs.

Pengaturan S3 Block Public Access

Jika permintaan yang gagal melibatkan akses publik atau kebijakan publik, periksa pengaturan S3 Blokir Akses Publik di akun, bucket, atau titik akses Anda. Untuk informasi selengkapnya tentang pemecahan masalah akses ditolak kesalahan yang terkait dengan setelan Akses Publik Blok S3, lihat. the section called “Akses ditolak karena pengaturan Blokir Akses Publik”

Pengaturan enkripsi Amazon S3

Amazon S3 mendukung enkripsi di sisi server di bucket Anda. Enkripsi di sisi server adalah enkripsi data di tempat tujuan oleh aplikasi atau layanan yang menerimanya. Amazon S3 mengenkripsi data Anda pada tingkat objek saat menulisnya ke disk di pusat AWS data dan mendekripsi untuk Anda saat Anda mengaksesnya.

Amazon S3 sudah menerapkan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3) sebagai tingkat dasar enkripsi bagi setiap bucket di Amazon S3. Amazon S3 juga memungkinkan Anda menentukan metode enkripsi di sisi server saat mengunggah objek.

Untuk meninjau status enkripsi di sisi server dan setelan enkripsi bucket

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Dari daftar Bucket, pilih bucket yang ingin Anda periksa pengaturan enkripsinya.

4. Pilih tab Properti.
5. Gulir ke bawah ke bagian Enkripsi default lalu lihat pengaturan Jenis enkripsi.

Untuk memeriksa pengaturan enkripsi Anda dengan menggunakan AWS CLI, gunakan get-bucket-encryptionperintah.

Untuk memeriksa status enkripsi objek

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Di daftar Bucket, pilih nama bucket yang berisi objek.
4. Dari daftar Objek, pilih nama objek yang ingin ditambahkan atau ubah enkripsinya.

   Halaman detail objek akan muncul.
5. Gulir ke bawah ke bagian Pengaturan enkripsi sisi server untuk melihat pengaturan enkripsi di sisi server objek.

Untuk memeriksa status enkripsi objek Anda dengan menggunakan AWS CLI, gunakan head-objectperintah.

Persyaratan enkripsi dan izin

Amazon S3 mendukung tiga jenis enkripsi di sisi server:

- Enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3)
- Enkripsi sisi server dengan kunci AWS Key Management Service (AWS KMS) (SSE-KMS)
- Enkripsi di sisi server dengan kunci yang disediakan pelanggan (SSE-C)

Berdasarkan pengaturan enkripsi Anda, pastikan bahwa persyaratan izin berikut terpenuhi:

- SSE-S3-Tidak diperlukan izin tambahan.
- SSE-KMS (dengan CMK)-Untuk mengunggah objek, izin `kms:GenerateDataKey` pada AWS KMS key diperlukan. Untuk mengunduh objek dan melakukan unggahan objek multibagian, izin `kms:Decrypt` pada kunci KMS diperlukan.

- SSE-KMS (dengan Kunci yang dikelola AWS) — Pemohon harus dari akun yang sama yang memiliki kunci KMS. `aws/s3` Peminta juga harus memiliki izin Amazon S3 yang benar untuk mengakses objek.
- SSE-C (dengan kunci yang disediakan pelanggan)-Tidak diperlukan izin tambahan. Anda dapat mengonfigurasi kebijakan bucket agar memerlukan dan membatasi enkripsi di sisi server dengan kunci enkripsi yang disediakan pelanggan untuk objek di bucket.

Jika objek dienkripsi dengan CMK, pastikan bahwa kebijakan kunci KMS memungkinkan Anda untuk melakukan `kms:GenerateDataKey` atau tindakan `kms:Decrypt`. Untuk petunjuk tentang memeriksa kebijakan kunci KMS Anda, lihat Melihat kebijakan kunci di Panduan Pengembang AWS Key Management Service .

Pengaturan Kunci Objek S3

Jika bucket Anda mengaktifkan Kunci Objek S3 dan objek tersebut dilindungi oleh periode retensi atau penahanan hukum, Amazon S3 akan menampilkan kesalahan Akses Ditolak (403 Forbidden) saat Anda mencoba menghapus objek.

Untuk memeriksa apakah bucket telah mengaktifkan Kunci Objek

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Dari daftar Bucket, pilih nama bucket yang ingin Anda tinjau.
4. Pilih tab Properti.
5. Gulir ke bawah hingga bagian Kunci Objek. Verifikasi apakah pengaturan Kunci Objek Diaktifkan atau Dinonaktifkan.

Untuk menentukan apakah objek dilindungi oleh periode retensi atau penahanan hukum, lihat informasi kunci untuk objek Anda.

Jika objek dilindungi oleh periode retensi atau penahanan hukum, periksa hal-hal berikut:

- Jika versi objek dilindungi oleh mode retensi kepatuhan, tidak ada cara untuk menghapusnya secara permanen. `DELETE`Permintaan permanen dari pemohon, termasuk pengguna Akun AWS root, akan menghasilkan kesalahan Akses Ditolak (403 Terlarang). Selain itu, perlu diketahui

bahwa saat Anda mengirimkan permintaan `DELETE` untuk objek yang dilindungi oleh mode retensi kepatuhan, Amazon S3 membuat penanda hapus untuk objek tersebut.

- Jika versi objek dilindungi dengan mode retensi tata kelola dan Anda memiliki izin `s3:BypassGovernanceRetention`, Anda dapat melewati perlindungan dan menghapus versi secara permanen. Untuk informasi selengkapnya, lihat Memintas mode tata kelola.
- Jika versi objek dilindungi oleh penahanan hukum, maka permintaan `DELETE` permanen dapat mengakibatkan kesalahan Akses Ditolak (403 Forbidden). Untuk menghapus versi objek secara permanen, Anda harus menghapus penahanan hukum pada versi objek. Untuk menghapus penahanan hukum, Anda harus memiliki izin `s3:PutObjectLegalHold`. Untuk informasi selengkapnya tentang menghapus penahanan hukum, lihat Mengonfigurasi Kunci Objek S3.

Kebijakan VPC endpoint

Jika Anda mengakses Amazon S3 menggunakan titik akhir virtual private cloud (VPC), pastikan kebijakan titik akhir VPC tidak menghalangi Anda mengakses sumber daya Amazon S3 Anda. Secara default, kebijakan titik akhir VPC mengizinkan semua permintaan ke Amazon S3. Anda juga dapat mengonfigurasi kebijakan titik akhir VPC untuk membatasi permintaan tertentu. Untuk informasi tentang cara memeriksa kebijakan titik akhir VPC Anda, lihat sumber daya berikut:

- the section called “Akses ditolak karena kebijakan titik akhir VPC — penolakan implisit”
- the section called “Akses ditolak karena kebijakan titik akhir VPC — penolakan eksplisit”
- Kontrol akses ke titik akhir VPC dengan menggunakan kebijakan titik akhir dalam Panduan AWS PrivateLink

AWS Organizations kebijakan

Jika Anda Akun AWS milik organisasi, AWS Organizations kebijakan dapat memblokir Anda mengakses sumber daya Amazon S3. Secara default, AWS Organizations kebijakan tidak memblokir permintaan apa pun ke Amazon S3. Namun, pastikan AWS Organizations kebijakan Anda belum dikonfigurasi untuk memblokir akses ke bucket S3. Untuk petunjuk tentang cara memeriksa AWS Organizations kebijakan Anda, lihat sumber daya berikut:

- the section called “Akses ditolak karena Kebijakan Kontrol Layanan — penolakan implisit”
- the section called “Akses ditolak karena Kebijakan Kontrol Layanan — penolakan eksplisit”
- the section called “Akses ditolak karena kebijakan kontrol sumber daya — penolakan eksplisit”
- Cantumkan semua kebijakan dalam Panduan AWS Organizations Pengguna

Selain itu, jika Anda salah mengonfigurasi kebijakan bucket untuk akun anggota untuk menolak akses semua pengguna ke bucket S3 Anda, Anda dapat membuka kunci bucket dengan meluncurkan sesi istimewa untuk akun anggota di IAM. Setelah meluncurkan sesi istimewa, Anda dapat menghapus kebijakan bucket yang salah konfigurasi untuk mendapatkan kembali akses ke bucket. Untuk informasi selengkapnya, lihat Melakukan tugas istimewa pada akun AWS Organizations anggota di Panduan AWS Identity and Access Management Pengguna.

CloudFront akses distribusi

Jika Anda menerima kesalahan Akses Ditolak (403 Terlarang) saat mencoba mengakses situs web statis S3 Anda CloudFront, periksa masalah umum berikut:

- Apakah Anda memiliki format nama domain asal yang benar?
  - Pastikan Anda menggunakan format titik akhir situs web S3 (bucket-name.s3-website-region.amazonaws.com) daripada titik akhir REST API
  - Verifikasi bahwa hosting situs web statis diaktifkan di bucket Anda
- Apakah kebijakan bucket Anda mengizinkan CloudFront akses?
  - Pastikan kebijakan bucket Anda menyertakan izin untuk Origin Access Identity (OAI) atau Origin Access Control (OAC) CloudFront distribusi Anda
  - Verifikasi kebijakan menyertakan izin s3: GetObject yang diperlukan

Untuk langkah dan konfigurasi pemecahan masalah tambahan, termasuk pengaturan halaman kesalahan dan pengaturan protokol, lihat Mengapa saya mendapatkan kesalahan “akses 403 ditolak” saat menggunakan titik akhir situs web Amazon S3 sebagai asal distribusi saya? CloudFront di pusat AWS re:Post pengetahuan.

> **Note**
>
> Kesalahan ini berbeda dari 403 kesalahan yang mungkin Anda terima saat mengakses S3 secara langsung. Untuk masalah CloudFront spesifik, pastikan untuk memeriksa pengaturan CloudFront distribusi dan konfigurasi S3 Anda.

Pengaturan titik akses

Jika Anda menerima kesalahan Akses Ditolak (403 Forbidden) saat membuat permintaan melalui Titik Akses Amazon S3, Anda mungkin perlu memeriksa hal-hal berikut:

- Konfigurasi untuk titik akses Anda
- Kebijakan pengguna IAM yang digunakan untuk titik akses Anda
- Kebijakan bucket yang digunakan untuk mengelola atau mengonfigurasi titik akses lintas akun

Konfigurasi dan kebijakan titik akses

- Saat Anda membuat titik akses, Anda dapat memilih untuk menunjuk Internet atau VPC sebagai asal jaringan. Jika asal jaringan disetel ke VPC saja, Amazon S3 akan menolak permintaan apa pun yang dibuat ke titik akses yang tidak berasal dari VPC yang ditentukan. Untuk memeriksa asal jaringan titik akses Anda, lihatMembuat titik akses terbatas pada cloud privat virtual.
- Dengan titik akses, Anda juga dapat mengonfigurasi pengaturan Blok Akses Publik kustom, yang bekerja sama dengan pengaturan Blok Akses Publik di tingkat bucket atau akun. Untuk memeriksa pengaturan Blok Akses Publik kustom Anda, lihat Mengelola akses publik ke titik akses untuk ember tujuan umum.
- Untuk membuat permintaan yang berhasil ke Amazon S3 dengan menggunakan titik akses, pastikan bahwa peminta memiliki izin IAM yang diperlukan. Untuk informasi selengkapnya, lihat Mengonfigurasi kebijakan IAM untuk menggunakan titik akses.
- Jika permintaan melibatkan titik akses lintas akun, pastikan pemilik bucket telah memperbarui kebijakan bucket untuk mengotorisasi permintaan dari titik akses. Untuk informasi selengkapnya, lihat Memberikan izin untuk titik akses lintas akun.

Jika kesalahan Akses Ditolak (403 Terlarang) masih berlanjut setelah memeriksa semua item dalam topik ini, ambil ID permintaan Amazon S3 Anda dan hubungi untuk panduan tambahan. Dukungan

Sumber daya tambahan

Untuk panduan lebih lanjut tentang kesalahan Akses Ditolak (403 Terlarang), Anda dapat memeriksa sumber daya berikut:

- Bagaimana cara memecahkan masalah 403 Access Denied Error dari Amazon S3? di pusat AWS re:Post pengetahuan.
- Mengapa saya mendapatkan kesalahan 403 Forbidden ketika saya mencoba mengakses bucket atau objek Amazon S3? di pusat AWS re:Post pengetahuan.
- Mengapa saya mendapatkan kesalahan Akses Ditolak ketika saya mencoba mengakses sumber daya Amazon S3 yang sama? Akun AWS di pusat AWS re:Post pengetahuan.

- Mengapa saya mendapatkan kesalahan Akses Ditolak ketika saya mencoba mengakses bucket Amazon S3 dengan akses baca publik? di pusat AWS re:Post pengetahuan.
- Mengapa saya mendapatkan kesalahan “ketidakcocokan tanda tangan” ketika saya mencoba menggunakan URL yang telah ditentukan sebelumnya untuk mengunggah objek ke Amazon S3? di pusat AWS re:Post pengetahuan.
- Mengapa saya mendapatkan kesalahan Access Denied untuk ListObjects V2 ketika saya menjalankan perintah sinkronisasi di bucket Amazon S3 saya? di pusat AWS re:Post pengetahuan.
- Mengapa saya mendapatkan kesalahan “akses 403 ditolak” ketika saya menggunakan titik akhir situs web Amazon S3 sebagai asal distribusi saya? CloudFront di pusat AWS re:Post pengetahuan.

AWS kebijakan terkelola untuk Amazon S3

Kebijakan AWS terkelola adalah kebijakan mandiri yang dibuat dan dikelola oleh AWS. AWS Kebijakan terkelola dirancang untuk memberikan izin bagi banyak kasus penggunaan umum sehingga Anda dapat mulai menetapkan izin kepada pengguna, grup, dan peran.

Perlu diingat bahwa kebijakan AWS terkelola mungkin tidak memberikan izin hak istimewa paling sedikit untuk kasus penggunaan spesifik Anda karena tersedia untuk digunakan semua pelanggan. AWS Kami menyarankan Anda untuk mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola pelanggan yang khusus untuk kasus penggunaan Anda.

Anda tidak dapat mengubah izin yang ditentukan dalam kebijakan AWS terkelola. Jika AWS memperbarui izin yang ditentukan dalam kebijakan AWS terkelola, pembaruan akan memengaruhi semua identitas utama (pengguna, grup, dan peran) yang dilampirkan kebijakan tersebut. AWS kemungkinan besar akan memperbarui kebijakan AWS terkelola saat baru Layanan AWS diluncurkan atau operasi API baru tersedia untuk layanan yang ada.

Untuk informasi selengkapnya, lihat Kebijakan terkelola AWS dalam Panduan Pengguna IAM.

AWS kebijakan terkelola: AmazonS3FullAccess

Anda dapat melampirkan kebijakan `AmazonS3FullAccess` ke identitas IAM Anda. Kebijakan ini akan memberikan izin yang mengizinkan akses penuh ke Amazon S3.

Untuk menampilkan izin untuk kebijakan ini, lihat AmazonS3FullAccess di AWS Management Console.

AWS kebijakan terkelola: AmazonS3ReadOnlyAccess

Anda dapat melampirkan kebijakan `AmazonS3ReadOnlyAccess` ke identitas IAM Anda. Kebijakan ini memberikan izin yang memungkinkan akses hanya-baca ke Amazon S3.

Untuk menampilkan izin untuk kebijakan ini, lihat AmazonS3ReadOnlyAccess di AWS Management Console.

Kebijakan terkelola AWS : AmazonS3ObjectLambdaExecutionRolePolicy

Menyediakan AWS Lambda fungsi izin yang diperlukan untuk mengirim data ke Lambda Objek S3 saat permintaan dibuat ke titik akses Lambda Objek S3. Juga memberikan izin Lambda untuk menulis ke log Amazon. CloudWatch

Untuk menampilkan izin untuk kebijakan ini, lihat AmazonS3ObjectLambdaExecutionRolePolicy di AWS Management Console.

Kebijakan terkelola AWS : S3UnlockBucketPolicy

Jika Anda salah mengonfigurasi kebijakan bucket untuk akun anggota untuk menolak akses semua pengguna ke bucket S3, Anda dapat menggunakan kebijakan AWS terkelola (`S3UnlockBucketPolicy`) ini untuk membuka kunci bucket. Untuk informasi selengkapnya tentang cara menghapus kebijakan bucket yang salah konfigurasi yang menolak semua prinsipal mengakses bucket Amazon S3, lihat Melakukan tugas istimewa pada akun anggota di Panduan Pengguna. AWS Organizations AWS Identity and Access Management

Pembaruan Amazon S3 ke AWS kebijakan terkelola

Lihat detail tentang pembaruan kebijakan AWS terkelola untuk Amazon S3 sejak layanan ini mulai melacak perubahan ini.

Perubahan	Deskripsi	Tanggal
Amazon S3 menambahkan `S3UnlockBucketPolicy`	Amazon S3 menambahkan kebijakan AWS terkelola baru yang dipanggil `S3UnlockBucketPolicy` untuk membuka kunci bucket dan menghapus kebijakan bucket	November 1, 2024

Perubahan	Deskripsi	Tanggal
	yang salah konfigurasi yang menolak semua prinsipal mengakses bucket Amazon S3.	
Amazon S3 menambahkan izin Jelaskan ke `AmazonS3ReadOnlyAccess`	Amazon S3 menambahkan izin `s3:Describe*` ke `AmazonS3ReadOnlyAccess` .	11 Agustus 2023
Amazon S3 menambahkan izin S3 Lambda Objek ke `AmazonS3FullAccess` dan `AmazonS3ReadOnlyAccess`	Amazon S3 memperbarui kebijakan `AmazonS3FullAccess` dan `AmazonS3ReadOnlyAccess` untuk menyertakan izin untuk S3 Lambda Objek.	27 September 2021
Amazon S3 menambahkan `AmazonS3ObjectLambdaExecutionRolePolicy`	Amazon S3 menambahkan kebijakan AWS terkelola baru yang disebut yang `AmazonS3ObjectLambdaExecutionRolePolicy` menyediakan izin fungsi Lambda untuk berinteraksi dengan Lambda Objek S3 dan menulis ke log. CloudWatch	18 Agustus 2021
Amazon S3 mulai melakukan pelacakan perubahan	Amazon S3 mulai melacak perubahan untuk kebijakan yang AWS dikelola.	18 Agustus 2021

Mengelola akses ke kumpulan data bersama dengan titik akses

Jalur akses Amazon S3 menyederhanakan akses data untuk AWS layanan atau aplikasi pelanggan apa pun yang menyimpan data di S3. Titik akses diberi nama titik akhir jaringan yang dilampirkan

ke sumber data seperti bucket atau Amazon FSx untuk volume OpenZFS. Untuk informasi tentang bekerja dengan bucket, lihat Ikhtisar ember tujuan umum. Untuk informasi tentang bekerja dengan FSx OpenZFS, lihat Apa itu Amazon FSx untuk OpenZFS di Panduan Pengguna OpenZFS.

Anda dapat menggunakan titik akses untuk melakukan operasi objek S3, seperti `GetObject` dan`PutObject`. Setiap titik akses memiliki izin dan kontrol jaringan berbeda yang diterapkan S3 untuk setiap permintaan yang dibuat melalui titik akses tersebut. Setiap titik akhir memberlakukan kebijakan jalur akses khusus yang memungkinkan Anda mengontrol penggunaan berdasarkan sumber daya, pengguna, atau kondisi lainnya. Jika jalur akses Anda dilampirkan ke bucket, kebijakan titik akses berfungsi bersama dengan kebijakan bucket yang mendasarinya. Anda dapat mengonfigurasi titik akses apa pun untuk menerima permintaan hanya dari cloud privat virtual (VPC) untuk membatasi akses data Amazon S3 ke jaringan pribadi. Anda juga dapat mengonfigurasi pengaturan blok akses publik khusus untuk setiap titik akses.

> **Note**
>
> Anda hanya dapat menggunakan titik akses untuk melakukan pengoperasian pada objek. Anda tidak dapat menggunakan titik akses untuk melakukan operasi Amazon S3 lainnya, seperti menghapus bucket atau membuat konfigurasi Replikasi S3. Untuk mengetahui daftar lengkap pengoperasian S3 yang mendukung titik akses, lihat Kompatibilitas titik akses.

Topik di bagian ini menjelaskan cara bekerja dengan jalur akses Amazon S3. Untuk topik tentang penggunaan titik akses dengan bucket direktori lihat,Mengelola akses ke kumpulan data bersama dalam bucket direktori dengan titik akses.

Topik

- Aturan penamaan, batasan, dan batasan penamaan titik akses
- Mereferensikan titik akses dengan ARNs, alias titik akses, atau gaya yang dihoskan secara virtual URIs
- Kompatibilitas titik akses
- Mengonfigurasi kebijakan IAM untuk menggunakan titik akses
- Pemantauan dan pencatatan titik akses
- Membuat titik akses
- Mengelola jalur akses Amazon S3 Anda untuk bucket tujuan umum
- Menggunakan jalur akses Amazon S3 untuk bucket tujuan umum

Aturan penamaan, batasan, dan batasan penamaan titik akses

Titik akses diberi nama titik akhir jaringan yang dilampirkan ke bucket atau volume pada sistem FSx file Amazon yang menyederhanakan pengelolaan data. Saat Anda membuat titik akses, Anda memilih nama dan Wilayah AWS untuk membuatnya. Topik berikut memberikan informasi tentang aturan, batasan, dan batasan penamaan titik akses.

Topik

- Aturan penamaan untuk titik akses
- Pembatasan dan batasan untuk titik akses
- Pembatasan dan batasan untuk titik akses yang dilampirkan ke volume pada sistem FSx file Amazon

Aturan penamaan untuk titik akses

Saat Anda membuat titik akses, Anda memilih namanya dan Wilayah AWS untuk membuatnya. Tidak seperti bucket tujuan umum nama titik akses tidak perlu unik di seluruh Akun AWS atau Wilayah AWS. Hal yang sama Akun AWS dapat membuat titik akses dengan nama yang sama di berbeda Wilayah AWS atau dua yang berbeda Akun AWS dapat menggunakan nama titik akses yang sama. Namun, dalam satu Wilayah AWS Akun AWS mungkin tidak memiliki dua titik akses bernama identik.

> **Note**
>
> Jika Anda memilih untuk mempublikasikan nama jalur akses Anda, hindari memasukkan informasi sensitif dalam nama titik akses. Nama titik akses diterbitkan dalam basis data yang dapat diakses publik yang dikenal sebagai Sistem Nama Domain (DNS).

Nama titik akses harus sesuai dengan DNS dan harus memenuhi ketentuan berikut:

- Harus unik dalam satu Akun AWS dan Wilayah AWS
- Harus dimulai dengan angka atau huruf kecil
- Panjang nama harus berkisar antara 3 sampai 50 karakter
- Tidak dapat diawali atau diakhiri dengan tanda hubung (-)
- Tidak dapat berisi garis bawah (_), huruf besar, spasi, atau titik (.)

- Tidak dapat diakhiri dengan akhiran `-s3alias` atau`-ext-s3alias`. Sufiks ini dicadangkan untuk nama alias titik akses. Untuk informasi selengkapnya, lihat Alias titik akses.

Pembatasan dan batasan untuk titik akses

Titik akses Amazon S3 memiliki pembatasan dan batasan berikut:

- Setiap titik akses dikaitkan dengan tepat satu bucket atau FSx untuk volume OpenZFS. Anda harus menentukan ini ketika Anda membuat titik akses. Setelah Anda membuat titik akses, Anda tidak dapat mengaitkannya dengan bucket yang berbeda atau FSx untuk volume OpenZFS. Namun, Anda dapat menghapus titik akses, dan kemudian membuat yang lain dengan nama yang sama.
- Setelah Anda membuat titik akses, Anda tidak dapat mengubah konfigurasi cloud privat virtual (VPC).
- Ukuran kebijakan titik akses dibatasi hingga 20 KB.
- Anda dapat membuat maksimum 10.000 titik akses per Akun AWS per Wilayah AWS. Jika Anda memerlukan lebih dari 10.000 titik akses untuk satu akun di satu Wilayah, Anda dapat meminta peningkatan kuota layanan. Untuk informasi lebih lanjut tentang kuota layanan dan permintaan peningkatan, lihat AWS Service Quotas dalam Referensi Umum AWS.
- Anda tidak dapat menggunakan titik akses sebagai tujuan untuk Replikasi S3. Untuk informasi selengkapnya tentang replikasi, lihat Mereplikasi objek di dalam dan di seluruh Wilayah.
- Anda tidak dapat menggunakan alias titik akses S3 sebagai sumber atau tujuan untuk operasi Pindah di konsol Amazon S3.
- Anda dapat mengatasi titik akses hanya dengan menggunakan virtual-host-style URLs. Untuk informasi lebih lanjut tentang virtual-host-style pengalamatan, lihatMengakses bucket tujuan umum Amazon S3.
- Operasi API yang mengontrol fungsionalitas titik akses (misalnya, `PutAccessPoint` dan `GetAccessPointPolicy`) tidak mendukung panggilan akun silang.
- Anda harus menggunakan AWS Signature Version 4 saat membuat permintaan ke titik akses dengan menggunakan REST APIs. Untuk informasi selengkapnya tentang mengautentikasi permintaan, lihat Mengautentikasi Permintaan (Versi AWS Tanda Tangan 4) di Referensi API Amazon Simple Storage Service.
- Titik akses hanya mendukung permintaan melalui HTTPS. Amazon S3 akan secara otomatis merespons dengan pengalihan HTTP untuk setiap permintaan yang dibuat melalui HTTP, untuk meningkatkan permintaan ke HTTPS.

- Titik akses tidak mendukung akses anonim.
- Setelah Anda membuat titik akses, Anda tidak dapat mengubah pengaturan blokir akses publik.
- Titik akses lintas akun tidak memberikan Anda akses ke data sampai Anda diberikan izin dari pemilik bucket. Pemilik bucket selalu memegang kendali penuh atas akses ke data, dan harus memperbarui kebijakan bucket untuk mengotorisasi permintaan dari titik akses lintas akun. Untuk melihat contoh kebijakan bucket, lihat Mengonfigurasi kebijakan IAM untuk menggunakan titik akses.
- Di Wilayah AWS mana Anda memiliki lebih dari 1.000 titik akses, Anda tidak dapat mencari titik akses berdasarkan nama di konsol Amazon S3.
- Saat Anda melihat titik akses lintas akun di konsol Amazon S3, kolom Akses menampilkan Tidak Diketahui. Konsol Amazon S3 tidak dapat menentukan apakah akses publik diberikan untuk bucket dan objek terkait. Kecuali Anda memerlukan konfigurasi publik untuk kasus penggunaan tertentu, sebaiknya Anda dan pemilik bucket memblokir semua akses publik ke titik akses dan bucket. Untuk informasi selengkapnya, lihat Melakukan blok akses publik ke penyimpanan Amazon S3 Anda.

Pembatasan dan batasan untuk titik akses yang dilampirkan ke volume pada sistem FSx file Amazon

Berikut ini adalah batasan khusus saat menggunakan titik akses yang dilampirkan ke volume pada sistem FSx file Amazon:

- Saat membuat titik akses Anda hanya dapat melampirkan titik akses ke volume pada sistem FSx file Amazon yang Anda miliki. Anda tidak dapat melampirkan ke volume yang dimiliki oleh orang lain Akun AWS.
- Anda tidak dapat menggunakan `CreateAccessPoint` API saat membuat dan melampirkan titik akses ke volume pada sistem FSx file Amazon. Anda harus menggunakan `CreateAndAttachS3AccessPoint` API.
- Anda tidak dapat mematikan pengaturan akses publik blok apa pun saat membuat atau menggunakan titik akses yang dilampirkan ke volume pada sistem FSx file Amazon.
- Anda tidak dapat mencantumkan objek atau menggunakan operasi Salin atau Pindahkan di konsol S3 dengan titik akses yang dilampirkan ke volume pada sistem FSx file Amazon.
- `CopyObject`didukung untuk titik akses yang dilampirkan ke volume FSx untuk OpenZFS hanya jika sumber dan tujuan adalah titik akses yang sama. Untuk informasi selengkapnya, tentang kompatibilitas titik akses, lihatKompatibilitas titik akses.
- Unggahan multipart dibatasi hingga 5GB.

- FSx untuk jenis penyebaran OpenZFS dan dukungan kelas penyimpanan bervariasi menurut. Wilayah AWS Untuk informasi selengkapnya, lihat Ketersediaan oleh Wilayah AWS di Panduan Pengguna OpenZFS.

Mereferensikan titik akses dengan ARNs, alias titik akses, atau gaya yang dihoskan secara virtual URIs

Setelah Anda membuat titik akses, Anda dapat menggunakan titik akhir ini untuk membentuk sejumlah operasi. Saat merujuk ke titik akses, Anda dapat menggunakan Amazon Resource Names (ARNs), alias access point, atau URI gaya host virtual.

Topik

- Titik akses ARNs
- Alias titik akses
- URI bergaya host virtual

Titik akses ARNs

Titik akses memiliki Nama Sumber Daya Amazon (ARNs). Access point ARNs mirip dengan bucket ARNs, tetapi mereka secara eksplisit diketik dan menyandikan titik akses Wilayah AWS dan Akun AWS ID pemilik jalur akses. Untuk informasi selengkapnya ARNs, lihat Mengidentifikasi AWS sumber daya dengan Amazon Resource Names (ARNs) di Panduan Pengguna IAM.

Jalur akses ARNs menggunakan format berikut:

```
arn:aws:s3:region:account-id:accesspoint/resource
```

- `arn:aws:s3:us-west-2:123456789012:accesspoint/test`mewakili titik akses bernama`test`, yang dimiliki oleh akun `123456789012` di Wilayah`us-west-2`.
- `arn:aws:s3:us-west-2:123456789012:accesspoint/*`mewakili semua titik akses di bawah akun `123456789012` di Wilayah`us-west-2`.

ARNs untuk objek yang diakses melalui jalur akses gunakan format berikut:

```
arn:aws:s3:region:account-id:accesspoint/access-point-name/object/resource
```

- `arn:aws:s3:us-west-2:123456789012:accesspoint/test/object/unit-01`mewakili objek`unit-01`, diakses melalui titik akses bernama`test`, dimiliki oleh akun `123456789012` di Wilayah`us-west-2`.
- `arn:aws:s3:us-west-2:123456789012:accesspoint/test/object/*`mewakili semua objek untuk titik akses bernama`test`, dalam akun `123456789012` di Wilayah`us-west-2`.
- `arn:aws:s3:us-west-2:123456789012:accesspoint/test/object/unit-01/finance/*`mewakili semua objek di bawah awalan `unit-01/finance/` untuk titik akses bernama`test`, dalam akun `123456789012` di Wilayah`us-west-2`.

Alias titik akses

Saat Anda membuat titik akses, Amazon S3 secara otomatis membuat alias yang dapat Anda gunakan alih-alih nama bucket Amazon S3 untuk akses data. Anda dapat menggunakan alias titik akses ini alih-alih Amazon Resource Name (ARN) untuk operasi bidang data titik akses. Untuk mengetahui daftar operasi ini, lihat Kompatibilitas titik akses.

Nama alias titik akses dibuat dalam namespace yang sama dengan bucket Amazon S3. Nama alias ini dibuat secara otomatis, dan tidak dapat diubah. Nama alias titik akses memenuhi semua persyaratan nama bucket Amazon S3 yang valid dan terdiri dari bagian-bagian berikut:

ACCESS POINT NAME`-METADATA-s3alias`(untuk titik akses yang dilampirkan ke bucket Amazon S3)

ACCESS POINT NAME`-METADATA-ext-s3alias`(untuk titik akses yang dilampirkan ke sumber data bucket non-S3)

> **Note**
>
> `-ext-s3alias`Sufiks `-s3alias` dan disediakan untuk nama alias titik akses dan tidak dapat digunakan untuk nama bucket atau titik akses. Untuk informasi selengkapnya tentang penamaan bucket di Amazon S3, lihat Aturan penamaan bucket tujuan umum.

Alias titik akses menggunakan kasus dan batasan

Saat mengadopsi titik akses, Anda dapat menggunakan alias titik akses tanpa memerlukan perubahan kode yang ekstensif.

Saat Anda membuat titik akses, Amazon S3 secara otomatis menghasilkan nama alias titik akses, seperti yang ditunjukkan dalam contoh berikut. Untuk menjalankan perintah ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3control create-access-point --bucket amzn-s3-demo-bucket1 --name my-access-point
 --account-id 111122223333
{
    "AccessPointArn": "arn:aws:s3:region:111122223333:accesspoint/my-access-point",
    "Alias": "my-access-point-aqfqprnstn7aefdfbarligizwgyfouse1a-s3alias"
}
```

Anda dapat menggunakan nama alias titik akses ini alih-alih nama bucket Amazon S3 dalam operasi bidang data apa pun. Untuk mengetahui daftar operasi ini, lihat Kompatibilitas titik akses.

AWS CLI Contoh berikut untuk `get-object` perintah menggunakan alias access point bucket untuk mengembalikan informasi tentang objek yang ditentukan. Untuk menjalankan perintah ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3api get-object --bucket my-access-point-aqfqprnstn7aefdfbarligizwgyfouse1a-
s3alias --key dir/my_data.rtf my_data.rtf

{
    "AcceptRanges": "bytes",
    "LastModified": "2020-01-08T22:16:28+00:00",
    "ContentLength": 910,
    "ETag": "\"00751974dc146b76404bb7290f8f51bb\"",
    "VersionId": "null",
    "ContentType": "text/rtf",
    "Metadata": {}
}
```

Keterbatasan alias titik akses

- Alias tidak dapat dikonfigurasi oleh pelanggan.
- Alias tidak dapat dihapus atau dimodifikasi atau dinonaktifkan pada titik akses.
- Anda dapat menggunakan nama alias titik akses ini alih-alih nama bucket Amazon S3 dalam beberapa operasi bidang data. Untuk mengetahui daftar operasi ini, lihat Kompatibilitas titik akses dengan operasi S3.

- Anda tidak dapat menggunakan nama alias titik akses untuk operasi bidang kontrol Amazon S3. Untuk daftar operasi bidang kontrol Amazon S3, lihat Kontrol Amazon S3 di Referensi API Amazon Simple Storage Service.
- Anda tidak dapat menggunakan alias titik akses S3 sebagai sumber atau tujuan untuk operasi Pindah di konsol Amazon S3.
- Alias tidak dapat digunakan dalam kebijakan AWS Identity and Access Management (IAM).
- Alias tidak dapat digunakan sebagai tujuan pencatatan untuk log akses server S3.
- Alias tidak dapat digunakan sebagai tujuan logging untuk log AWS CloudTrail .
- Amazon SageMaker AI GroundTruth tidak mendukung alias titik akses.

URI bergaya host virtual

Titik akses hanya mendukung virtual-host-style pengalamatan. Dalam URI gaya host virtual, nama titik akses Akun AWS, dan Wilayah AWS merupakan bagian dari nama domain di URL. Untuk informasi lebih lanjut tentang hosting virtual, lihatHosting virtual dari ember tujuan umum.

URI bergaya tuan rumah virtual untuk titik akses menggunakan format berikut:

```
https://access-point-name-account-id.s3-accesspoint.region.amazonaws.com
```

> **Note**
>
> - Jika nama titik akses Anda mencakup karakter tanda pisah (-), sertakan tanda pisah dalam URL dan masukkan tanda pisah lainnya sebelum ID akun. Misalnya, untuk menggunakan titik akses bernama *finance-docs* dimiliki oleh akun *123456789012* di Wilayah*us-west-2*, URL yang sesuai adalah`https://finance-docs-123456789012.s3-accesspoint.us-west-2.amazonaws.com`.
> - Jalur akses S3 tidak mendukung akses melalui HTTP. Jalur akses hanya mendukung akses aman melalui HTTPS.

Kompatibilitas titik akses

Anda dapat menggunakan titik akses untuk mengakses objek menggunakan subset Amazon APIs S3 berikut. Semua operasi yang tercantum di bawah ini dapat menerima alias titik akses ARNs atau titik akses.

Untuk contoh menggunakan titik akses untuk melakukan operasi pada objek, lihatMenggunakan jalur akses Amazon S3 untuk bucket tujuan umum.

Kompatibilitas titik akses dengan operasi S3

Tabel berikut adalah sebagian daftar operasi Amazon S3 dan jika kompatibel dengan titik akses. Semua operasi di bawah ini didukung oleh titik akses menggunakan bucket S3 sebagai sumber datanya, sementara hanya beberapa operasi yang didukung oleh titik akses menggunakan volume FSx untuk OpenZFS sebagai sumber data.

Untuk informasi selengkapnya lihat, kompatibilitas titik akses di Panduan Pengguna FSx untuk OpenZFS.

Operasi S3	Titik akses yang terpasang pada bucket S3	Titik akses yang dilampirkan ke volume FSx untuk OpenZFS
AbortMultipartUpload	Didukung	Didukung
CompleteMultipartUpload	Didukung	Didukung
CopyObject (Hanya salinan wilayah yang sama)	Didukung	Didukung, jika sumber dan tujuan adalah titik akses yang sama
CreateMultipartUpload	Didukung	Didukung
DeleteObject	Didukung	Didukung
DeleteObjects	Didukung	Didukung
DeleteObjectTagging	Didukung	Didukung
GetBucketAcl	Didukung	Tidak didukung
GetBucketCors	Didukung	Tidak didukung
GetBucketLocation	Didukung	Didukung

Operasi S3	Titik akses yang terpasang pada bucket S3	Titik akses yang dilampirkan ke volume FSx untuk OpenZFS
GetBucketNotificationConfiguration	Didukung	Tidak didukung
GetBucketPolicy	Didukung	Tidak didukung
GetObject	Didukung	Didukung
GetObjectAcl	Didukung	Tidak didukung
GetObjectAttributes	Didukung	Didukung
GetObjectLegalHold	Didukung	Tidak didukung
GetObjectRetention	Didukung	Tidak didukung
GetObjectTagging	Didukung	Didukung
HeadBucket	Didukung	Didukung
HeadObject	Didukung	Didukung
ListMultipartUploads	Didukung	Didukung
ListObjects	Didukung	Didukung
ListObjectsV2	Didukung	Didukung
ListObjectVersions	Didukung	Tidak didukung
ListParts	Didukung	Didukung
Presign	Didukung	Didukung
PutObject	Didukung	Didukung
PutObjectAcl	Didukung	Tidak didukung

Operasi S3	Titik akses yang terpasang pada bucket S3	Titik akses yang dilampirkan ke volume FSx untuk OpenZFS
PutObjectLegalHold	Didukung	Tidak didukung
PutObjectRetention	Didukung	Tidak didukung
PutObjectTagging	Didukung	Didukung
RestoreObject	Didukung	Tidak didukung
UploadPart	Didukung	Didukung
UploadPartCopy (Hanya salinan wilayah yang sama)	Didukung	Didukung, jika sumber dan tujuan adalah titik akses yang sama

Mengonfigurasi kebijakan IAM untuk menggunakan titik akses

Kebijakan sumber daya dukungan jalur akses Amazon S3 AWS Identity and Access Management (IAM) yang memungkinkan Anda mengontrol penggunaan titik akses berdasarkan sumber daya, pengguna, atau kondisi lainnya. Agar aplikasi atau pengguna dapat mengakses objek melalui titik akses, baik titik akses dan bucket atau sistem FSx file Amazon yang mendasarinya harus mengizinkan permintaan tersebut.

> **Important**
>
> Pembatasan yang Anda sertakan dalam kebijakan titik akses hanya berlaku untuk permintaan yang dibuat melalui titik akses tersebut. Melampirkan titik akses ke bucket tidak mengubah perilaku sumber daya yang mendasarinya. Semua operasi yang ada terhadap bucket yang tidak dilakukan melalui jalur akses Anda akan terus berfungsi seperti sebelumnya.

Saat Anda menggunakan kebijakan sumber daya IAM, pastikan untuk menyelesaikan peringatan keamanan, kesalahan, peringatan umum, dan saran AWS Identity and Access Management Access Analyzer sebelum Anda menyimpan kebijakan Anda. Penganalisis Akses IAM menjalankan

pemeriksaan kebijakan untuk memvalidasi kebijakan Anda terhadap tata bahasa kebijakan IAM dan praktik terbaik. Pemeriksaan ini menghasilkan temuan dan memberikan rekomendasi untuk membantu Anda membuat kebijakan yang berfungsi dan sesuai dengan praktik terbaik keamanan.

Untuk mempelajari validasi kebijakan menggunakan Penganalisis Akses IAM lebih lanjut, lihat Memvalidasi kebijakan Penganalisis Akses IAM di Panduan Pengguna IAM. Untuk melihat daftar peringatan, kesalahan, dan saran yang ditampilkan oleh Penganalisis Akses IAM, lihat referensi pemeriksaan kebijakan Penganalisis Akses IAM.

Contoh kebijakan untuk titik akses

Contoh berikut menunjukkan cara membuat kebijakan IAM untuk mengontrol permintaan yang dilakukan melalui titik akses.

> **Note**
>
> Izin yang diberikan dalam kebijakan titik akses hanya berlaku jika bucket yang mendasarinya juga mengizinkan akses yang sama. Anda dapat melakukan ini menggunakan dua cara:
>
> 1. (Disarankan) Delegasikan kontrol akses dari bucket ke titik akses seperti yang dijelaskan dalam Mendelegasikan kontrol akses ke titik akses.
> 2. Tambahkan izin yang sama yang tercantum dalam kebijakan titik akses ke kebijakan bucket dasar. Contoh kebijakan titik akses Contoh 1 mendemonstrasikan cara memodifikasi kebijakan bucket dasar agar akses yang diperlukan dapat dilakukan.

Example 1–Pemberian kebijakan titik akses

Kebijakan titik akses berikut memberikan izin kepada pengguna IAM *Jane* pada akun *123456789012* atas objek GET dan PUT dengan prefiks *Jane/* melalui titik akses *my-access-point* pada akun *123456789012*.

```
{
    "Version":"2012-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::123456789012:user/Jane"
```

```
        },
        "Action": ["s3:GetObject", "s3:PutObject"],
        "Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/my-access-point/
object/Jane/*"
    }]
}
```

> **Note**
>
> Agar kebijakan titik akses secara efektif memberikan akses ke *Jane*, bucket yang mendasari juga harus mengizinkan akses yang sama ke *Jane*. Anda dapat mendelegasikan kontrol akses dari bucket ke titik akses seperti yang dijelaskan dalam Mendelegasikan kontrol akses ke titik akses. Atau, Anda dapat menambahkan kebijakan berikut ke bucket yang mendasarinya untuk memberikan izin yang diperlukan kepada Jane. Perhatikan bahwa entri `Resource` berbeda antara titik akses dan kebijakan bucket.
>
> ```
> {
>     "Version": "2012-10-17",
>     "Statement": [
>     {
>         "Effect": "Allow",
>         "Principal": {
>             "AWS": "arn:aws:iam::123456789012:user/Jane"
>         },
>         "Action": ["s3:GetObject", "s3:PutObject"],
>         "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/Jane/*"
>     }]
> }
> ```

Example 2–Kebijakan titik akses dengan kondisi tag

Kebijakan titik akses berikut memberikan izin *Mateo* kepada pengguna IAM dalam akun *123456789012* untuk objek `GET` melalui titik akses *my-access-point* dalam akun *123456789012* yang kunci tag *data* diatur dengan nilai *finance*.

```
{
    "Version":"2012-10-17",
    "Statement": [
    {
```

```
        "Effect":"Allow",
        "Principal" : {
            "AWS": "arn:aws:iam::123456789012:user/Mateo"
        },
        "Action":"s3:GetObject",
        "Resource" : "arn:aws:s3:us-west-2:123456789012:accesspoint/my-access-point/
object/*",
        "Condition" : {
            "StringEquals": {
                "s3:ExistingObjectTag/data": "finance"
            }
        }
    }]
}
```

Example 3–Kebijakan titik akses yang mengizinkan daftar bucket

Kebijakan titik akses berikut memungkinkan pengguna IAM `Arnav` di akun *123456789012* untuk melihat objek yang ada dalam bucket yang mendasari titik akses *my-access-point* pada akun *123456789012*.

```
{
    "Version":"2012-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::123456789012:user/Arnav"
        },
        "Action": "s3:ListBucket",
        "Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/my-access-point"
    }]
}
```

Example 4–Kebijakan kontrol layanan

Kebijakan kontrol layanan berikut mewajibkan semua titik akses baru untuk dibuat dengan asal jaringan cloud privat virtual (VPC). Dengan kebijakan ini, pengguna di dalam organisasi Anda tidak dapat membuat titik akses baru yang dapat diakses dari internet.

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
    {
        "Effect": "Deny",
        "Action": "s3:CreateAccessPoint",
        "Resource": "*",
        "Condition": {
            "StringNotEquals": {
                "s3:AccessPointNetworkOrigin": "VPC"
            }
        }
    }]
}
```

Example 5–Kebijakan bucket untuk membatasi operasi S3 ke asal jaringan VPC

Kebijakan bucket berikut membatasi akses ke semua operasi S3 object untuk bucket *amzn-s3-demo-bucket* ke titik akses dengan asal jaringan VPC.

> **Important**
>
> Sebelum menggunakan pernyataan seperti yang ditunjukkan pada contoh ini, pastikan Anda tidak perlu menggunakan fitur yang tidak didukung oleh titik akses, seperti Cross-Region Replication.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Principal": "*",
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:BypassGovernanceRetention",
                "s3:DeleteObject",
                "s3:DeleteObjectTagging",
                "s3:DeleteObjectVersion",
                "s3:DeleteObjectVersionTagging",
                "s3:GetObject",
                "s3:GetObjectAcl",
                "s3:GetObjectLegalHold",
                "s3:GetObjectRetention",
```

```
                "s3:GetObjectTagging",
                "s3:GetObjectVersion",
                "s3:GetObjectVersionAcl",
                "s3:GetObjectVersionTagging",
                "s3:ListMultipartUploadParts",
                "s3:PutObject",
                "s3:PutObjectAcl",
                "s3:PutObjectLegalHold",
                "s3:PutObjectRetention",
                "s3:PutObjectTagging",
                "s3:PutObjectVersionAcl",
                "s3:PutObjectVersionTagging",
                "s3:RestoreObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Condition": {
                "StringNotEquals": {
                    "s3:AccessPointNetworkOrigin": "VPC"
                }
            }
        }
    ]
}
```

Kunci syarat

Titik akses S3 memiliki kunci kondisi yang dapat digunakan dalam kebijakan IAM untuk mengontrol akses ke sumber daya Anda. Kunci kondisi berikut hanya mewakili bagian dari kebijakan IAM. Untuk contoh kebijakan lengkap, lihat Contoh kebijakan untuk titik akses, the section called “Mendelegasikan kontrol akses ke titik akses”, dan the section called “Memberikan izin untuk titik akses lintas akun”.

`s3:DataAccessPointArn`

Contoh ini menunjukkan string yang dapat Anda gunakan untuk mencocokkan ARN titik akses. Contoh berikut cocok dengan semua titik akses untuk Akun AWS *123456789012* di Wilayah*us-west-2*:

```
"Condition" : {
    "StringLike": {
        "s3:DataAccessPointArn": "arn:aws:s3:us-west-2:123456789012:accesspoint/*"
    }
```

```
}
```

`s3:DataAccessPointAccount`

Contoh ini menunjukkan operator string yang dapat Anda gunakan untuk mencocokkan ID akun pemilik titik akses. Contoh berikut cocok dengan semua titik akses yang dimiliki oleh Akun AWS *123456789012*.

```
"Condition" : {
    "StringEquals": {
        "s3:DataAccessPointAccount": "123456789012"
    }
}
```

`s3:AccessPointNetworkOrigin`

Contoh ini menunjukkan operator string yang dapat Anda gunakan untuk mencocokkan asal jaringan, baik `Internet` atau VPC. Contoh berikut hanya mencocokkan titik akses dengan asal VPC.

```
"Condition" : {
    "StringEquals": {
        "s3:AccessPointNetworkOrigin": "VPC"
    }
}
```

Untuk informasi selengkapnya tentang menggunakan kunci kondisi dengan Amazon S3, lihat Tindakan, sumber daya, dan kunci kondisi untuk Amazon S3 di Referensi Otorisasi Layanan.

Untuk informasi selengkapnya tentang izin operasi S3 API menurut jenis sumber daya S3, lihat. Izin yang diperlukan untuk operasi API Amazon S3

Mendelegasikan kontrol akses ke titik akses

Anda dapat mendelegasikan kontrol akses untuk sebuah bucket ke titik akses bucket. Kebijakan ini memungkinkan akses penuh ke semua titik akses yang dimiliki oleh akun pemilik bucket. Oleh karena itu, semua akses ke bucket ini dikendalikan oleh kebijakan yang melekat pada titik aksesnya. Kami menyarankan Anda untuk mengonfigurasi bucket dengan cara ini untuk semua kasus penggunaan yang tidak memerlukan akses langsung ke bucket.

Example 6–Kebijakan bucket yang mendelegasikan kontrol akses ke titik akses

```
{
    "Version": "2012-10-17",
    "Statement" : [
    {
        "Effect": "Allow",
        "Principal" : { "AWS": "*" },
        "Action" : "*",
        "Resource" : [ "Bucket ARN", "Bucket ARN/*"],
        "Condition": {
            "StringEquals" : { "s3:DataAccessPointAccount" : "Bucket owner's account
 ID" }
        }
    }]
}
```

Memberikan izin untuk titik akses lintas akun

Untuk membuat titik akses ke bucket yang dimiliki oleh akun lain, Anda harus terlebih dahulu membuat titik akses dengan menentukan nama bucket dan ID pemilik akun. Kemudian, pemilik bucket harus memperbarui kebijakan bucket untuk mengotorisasi permintaan dari titik akses. Membuat titik akses mirip dengan membuat DNS CNAME di mana titik akses tidak menyediakan akses ke konten bucket. Semua akses bucket dikendalikan oleh kebijakan bucket. Contoh kebijakan bucket berikut mengizinkan `GET` dan `LIST` meminta pada bucket dari titik akses yang dimiliki oleh Akun AWS yang tepercaya.

Ganti *`Bucket ARN`* dengan ARN ember.

Example 7 — Kebijakan bucket mendelegasikan izin ke yang lain Akun AWS

```
{
    "Version": "2012-10-17",
    "Statement" : [
    {
        "Effect": "Allow",
        "Principal" : { "AWS": "*" },
        "Action" : ["s3:GetObject","s3:ListBucket"],
        "Resource" : [ "Bucket ARN", "Bucket ARN/*"],
        "Condition": {
            "StringEquals" : { "s3:DataAccessPointAccount" : "Access point owner's
 account ID" }
```

```
        }
    }]
}
```

> **Note**
>
> Jalur akses lintas akun hanya tersedia untuk titik akses yang terpasang pada bucket S3. Anda tidak dapat melampirkan titik akses ke volume pada sistem FSx file Amazon yang dimiliki oleh orang lain Akun AWS.

Pemantauan dan pencatatan titik akses

Amazon S3 mencatat permintaan yang dibuat melalui titik akses dan permintaan yang dibuat untuk operasi API yang mengelola titik akses, seperti `CreateAccessPoint` dan. `GetAccessPointPolicy` Untuk memantau dan mengelola pola penggunaan, Anda juga dapat mengonfigurasi metrik permintaan CloudWatch Log Amazon untuk titik akses.

Topik

- CloudWatch metrik permintaan
- AWS CloudTrail log untuk permintaan yang dibuat melalui titik akses

CloudWatch metrik permintaan

Untuk memahami dan meningkatkan kinerja aplikasi yang menggunakan titik akses, Anda dapat menggunakan CloudWatch metrik permintaan Amazon S3. Metrik permintaan membantu Anda memantau permintaan Amazon S3 untuk mengidentifikasi dan bertindak terhadap masalah operasional dengan cepat.

Secara default, metrik permintaan tersedia di tingkat bucket. Namun, Anda dapat menentukan filter untuk metrik permintaan menggunakan prefiks bersama, tag objek, atau titik akses. Saat Anda membuat filter titik akses, konfigurasi metrik permintaan menyertakan permintaan ke titik akses yang Anda tentukan. Anda dapat menerima metrik, mengatur alarm, dan mengakses dasbor untuk melihat operasi waktu nyata yang dilakukan melalui titik akses ini.

Anda harus memilih untuk meminta metrik dengan mengonfigurasinya di dalam konsol, atau menggunakan API Amazon S3. Metrik permintaan tersedia dalam interval 1 menit setelah beberapa

latensi untuk diproses. Metrik permintaan ditagih dengan tarif yang sama dengan metrik CloudWatch kustom. Untuk informasi lebih lanjut, lihat harga Amazon CloudWatch .

Untuk membuat konfigurasi metrik permintaan yang memfilter berdasarkan titik akses, lihat Membuat konfigurasi metrik yang melakukan filter berdasarkan prefiks, tag objek, atau titik akses.

AWS CloudTrail log untuk permintaan yang dibuat melalui titik akses

Anda dapat mencatat permintaan yang dibuat melalui titik akses dan permintaan yang dibuat untuk APIs yang mengelola titik akses, seperti `CreateAccessPoint` dan `GetAccessPointPolicy`, dengan menggunakan log akses server dan AWS CloudTrail.

CloudTrail entri log untuk permintaan yang dibuat melalui titik akses termasuk titik akses ARN di `resources` bagian log.

Sebagai contoh, anggap Anda memiliki konfigurasi berikut:

- Sebuah bucket bernama *amzn-s3-demo-bucket1* di Wilayah *us-west-2* yang berisi objek *my-image.jpg*
- Titik akses bernama *my-bucket-ap* yang terkait dengan *amzn-s3-demo-bucket1*
- Akun AWS ID dari *123456789012*

Contoh berikut menunjukkan `resources` bagian dari entri CloudTrail log untuk konfigurasi sebelumnya:

```
"resources": [
        {"type": "AWS::S3::Object",
            "ARN": "arn:aws:s3:::amzn-s3-demo-bucket1/my-image.jpg"
        },
        {"accountId": "123456789012",
            "type": "AWS::S3::Bucket",
            "ARN": "arn:aws:s3:::amzn-s3-demo-bucket1"
        },
        {"accountId": "123456789012",
            "type": "AWS::S3::AccessPoint",
            "ARN": "arn:aws:s3:us-west-2:123456789012:accesspoint/my-bucket-ap"
        }
    ]
```

Jika Anda menggunakan titik akses yang dilampirkan ke volume pada sistem FSx file Amazon, `resources` bagian entri CloudTrail log akan terlihat berbeda. Misalnya:

```
"resources": [
        {
            "accountId": "123456789012",
            "type": "AWS::FSx::Volume",
            "ARN": "arn:aws:fsx:us-east-1:123456789012:volume/fs-0123456789abcdef9/
fsvol-01234567891112223"
        }
    ]
```

Untuk informasi selengkapnya tentang Log Akses Server S3, lihat Pencatatan permintaan dengan pencatatan akses server. Untuk informasi lebih lanjut tentang AWS CloudTrail, lihat Apa itu AWS CloudTrail? dalam AWS CloudTrail User Guide.

Membuat titik akses

Anda dapat membuat titik akses S3 dengan menggunakan REST AWS Management Console API, AWS Command Line Interface (AWS CLI) AWS SDKs, atau Amazon S3. Titik akses diberi nama titik akhir jaringan yang dilampirkan ke sumber data seperti bucket atau Amazon FSx untuk volume OpenZFS.

Secara default, Anda dapat membuat hingga 10.000 titik akses per Wilayah untuk setiap Akun AWS Anda. Jika Anda memerlukan lebih dari 10.000 titik akses untuk satu akun di satu Wilayah, Anda dapat meminta peningkatan kuota layanan. Untuk informasi lebih lanjut tentang kuota layanan dan permintaan peningkatan, lihat AWS Service Quotas dalam Referensi Umum AWS.

Topik

- Membuat titik akses dengan bucket S3
- Membuat titik akses dengan Amazon FSx
- Membuat titik akses terbatas pada cloud privat virtual
- Mengelola akses publik ke titik akses untuk ember tujuan umum

Membuat titik akses dengan bucket S3

Titik akses dikaitkan dengan tepat satu bucket tujuan umum Amazon S3. Jika Anda ingin menggunakan ember di dalam Anda Akun AWS, Anda harus terlebih dahulu membuat ember. Untuk

informasi tentang membuat bucket, lihat Membuat, mengonfigurasi, dan bekerja dengan bucket tujuan umum Amazon S3.

Anda juga dapat membuat titik akses lintas akun yang terkait dengan bucket di bucket lain Akun AWS, selama Anda mengetahui nama bucket dan ID akun pemilik bucket. Namun, membuat titik akses lintas akun tidak memberikan Anda akses ke data di bucket sampai Anda diberikan izin dari pemilik bucket. Pemilik bucket harus memberikan akses akun pemilik titik akses (akun Anda) ke bucket melalui kebijakan bucket. Untuk informasi selengkapnya, lihat Memberikan izin untuk titik akses lintas akun.

Menggunakan konsol S3

Untuk membuat titik akses

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi di bagian atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah tempat Anda ingin membuat titik akses. Titik akses harus dibuat di Wilayah yang sama dengan bucket terkait.
3. Pada panel navigasi di kiri, pilih Titik Akses.
4. Pada halaman Titik Akses, pilih Buat titik akses.
5. Di dalam bidang Nama titik akses, masukkan nama untuk titik akses tersebut. Untuk informasi lebih lanjut tentang penamaan titik akses, lihat Aturan penamaan untuk titik akses.
6. Untuk Sumber data, tentukan bucket S3 yang ingin Anda gunakan dengan titik akses.

   Untuk menggunakan bucket di dalam akun Anda, pilih Pilih bucket di akun ini, dan masukkan atau telusuri nama bucket.

   Untuk menggunakan bucket di tempat lain Akun AWS, pilih Tentukan bucket di akun lain, lalu masukkan Akun AWS ID dan nama bucket. Jika Anda menggunakan bucket di bucket lain Akun AWS, pemilik bucket harus memperbarui kebijakan bucket untuk mengotorisasi permintaan dari titik akses. Untuk contoh kebijakan bucket, lihat Memberikan izin untuk titik akses lintas akun.

   > **Note**
   >
   > Untuk informasi tentang menggunakan volume FSx untuk OpenZFS sebagai sumber data, lihat. Membuat titik akses dengan Amazon FSx

7. Pilih Network Origin, baik Internet atau Virtual Private Cloud (VPC). Jika Anda memilih virtual private cloud (VPC), masukkan ID VPC yang ingin Anda gunakan dengan titik akses.

   Untuk informasi lebih lanjut tentang asal jaringan untuk titik akses, lihat Membuat titik akses terbatas pada cloud privat virtual.

8. Di bagian bawah Pengaturan Blokir Titik Akses untuk Titik Akses ini, pilih pengaturan blokir akses publik yang ingin Anda terapkan ke titik akses tersebut. Semua pengaturan blok akses publik diaktifkan secara default untuk titik akses baru. Kami menyarankan Anda tetap mengaktifkan semua pengaturan kecuali Anda tahu bahwa Anda memiliki kebutuhan khusus untuk menonaktifkan salah satu pengaturan tersebut.

   > **Note**
   >
   > Setelah Anda membuat titik akses, Anda tidak dapat mengubah pengaturan blokir akses publik.

   Untuk informasi selengkapnya tentang menggunakan Blokir Akses Publik Amazon S3 dengan titik akses, lihat Mengelola akses publik ke titik akses untuk ember tujuan umum.

9. (Opsional) Di bawah Kebijakan Titik Akses-opsional, tentukan kebijakan titik akses. Sebelum Anda menyimpan kebijakan Anda, pastikan untuk mengatasi peringatan keamanan, kesalahan, peringatan umum, dan saran apa pun. Untuk informasi selengkapnya tentang penentuan kebijakan titik akses, lihat Contoh kebijakan untuk titik akses.

10. Pilih Buat titik akses.

Menggunakan AWS CLI

Contoh berikut membuat Titik Akses Lambda Objek yang diberi nama *example-ap* untuk bucket *amzn-s3-demo-bucket* di akun *111122223333*. Untuk membuat titik akses, Anda mengirim permintaan ke Amazon S3 yang menentukan hal berikut:

- Nama titik akses. Untuk informasi tentang aturan penamaan, lihat the section called “Aturan penamaan untuk titik akses”.
- Nama bucket yang ingin Anda asosiasikan dengan titik akses.
- ID akun untuk Akun AWS yang memiliki jalur akses.

```
aws s3control create-access-point --name example-ap --account-id 111122223333 --
bucket amzn-s3-demo-bucket
```

Saat Anda membuat titik akses dengan menggunakan bucket di tempat lain Akun AWS, sertakan `--bucket-account-id` parameternya. Contoh perintah berikut membuat titik akses di Akun AWS *111122223333*, menggunakan bucket *amzn-s3-demo-bucket2*, yang ada di Akun AWS *444455556666*.

```
aws s3control create-access-point --name example-ap --account-id 111122223333 --
bucket amzn-s3-demo-bucket --bucket-account-id 444455556666
```

Penggunaan API REST

Anda dapat menggunakan REST API untuk membuat titik akses. Untuk informasi selengkapnya, lihat CreateAccessPoint dalam Referensi API Amazon Simple Storage Service.

Membuat titik akses dengan Amazon FSx

Anda dapat membuat dan melampirkan titik akses ke volume FSx untuk OpenZFS menggunakan FSx konsol Amazon, AWS CLI, atau API. Setelah terpasang, Anda dapat menggunakan objek S3 APIs untuk mengakses data file Anda. Data Anda terus berada di sistem FSx file Amazon dan terus dapat diakses secara langsung untuk beban kerja Anda yang ada. Anda terus mengelola penyimpanan Anda menggunakan semua kemampuan manajemen penyimpanan FSx untuk OpenZFS, termasuk backup, snapshot, kuota pengguna dan grup, dan kompresi.

Untuk petunjuk tentang membuat titik akses dan melampirkannya ke volume FSx untuk OpenZFS lihat, Membuat titik akses di Panduan Pengguna FSx untuk OpenZFS.

Membuat titik akses terbatas pada cloud privat virtual

Ketika Anda membuat titik akses, Anda dapat memilih untuk membuat jalur akses dapat diakses dari internet, atau Anda dapat menentukan bahwa semua permintaan yang dibuat melalui titik akses tersebut harus berasal dari virtual private cloud (VPC) tertentu. Titik akses yang dapat diakses dari internet dikatakan memiliki asal jaringan dari `Internet`. Ini dapat digunakan dari mana saja di internet, tunduk pada pembatasan akses lain yang berlaku untuk titik akses, sumber data yang mendasari, dan sumber daya terkait, seperti objek yang diminta. Titik akses yang hanya dapat diakses dari VPC yang ditentukan memiliki asal jaringan `VPC`, dan Amazon S3 akan menolak permintaan apa pun yang diajukan ke titik akses yang tidak berasal dari VPC tersebut.

> **Important**
>
> Anda hanya dapat menentukan asal jaringan titik akses saat membuat titik akses. Setelah membuat titik akses, Anda tidak dapat mengubah asal jaringannya.

Untuk membatasi titik akses ke akses kustom VPC, Anda harus menyertakan parameter `VpcConfiguration` dengan permintaan untuk membuat titik akses. Dalam parameter `VpcConfiguration`, Anda menentukan ID VPC yang ingin Anda bisa gunakan titik aksesnya. Jika permintaan dibuat melalui titik akses, permintaan harus berasal dari VPC atau Amazon S3 akan menolaknya.

Anda dapat mengambil asal jaringan titik akses menggunakan AWS CLI, AWS SDKs, atau REST APIs. Jika titik akses memiliki konfigurasi VPC yang ditentukan, asal jaringannya adalah `VPC`. Jika tidak, asal jaringan titik aksesnya adalah `Internet`.

Contoh: Buat dan batasi jalur akses ke ID VPC

Contoh berikut membuat titik akses yang bernama `example-vpc-ap` untuk bucket *`amzn-s3-demo-bucket`* dalam akun `123456789012` yang mengizinkan akses hanya dari VPC `vpc-1a2b3c`. Contoh tersebut kemudian memverifikasi bahwa titik akses yang baru memiliki asal jaringan `VPC`.

AWS CLI

```
aws s3control create-access-point --name example-vpc-ap --account-id 123456789012 --
bucket amzn-s3-demo-bucket --vpc-configuration VpcId=vpc-1a2b3c
```

```
aws s3control get-access-point --name example-vpc-ap --account-id 123456789012

{
    "Name": "example-vpc-ap",
    "Bucket": "amzn-s3-demo-bucket",
    "NetworkOrigin": "VPC",
    "VpcConfiguration": {
        "VpcId": "vpc-1a2b3c"
    },
    "PublicAccessBlockConfiguration": {
        "BlockPublicAcls": true,
        "IgnorePublicAcls": true,
        "BlockPublicPolicy": true,
```

```
        "RestrictPublicBuckets": true
    },
    "CreationDate": "2019-11-27T00:00:00Z"
}
```

Untuk menggunakan titik akses dengan VPC, Anda harus mengubah kebijakan akses untuk titik akhir VPC Anda. Titik akhir VPC memungkinkan lalu lintas mengalir dari VPC Anda ke Amazon S3. Kebijakan kontrol akses yang mengontrol bagaimana sumber daya dalam VPC diizinkan untuk berinteraksi dengan Amazon S3. Permintaan dari VPC ke Amazon S3 hanya berhasil melalui titik akses jika kebijakan titik akhir VPC memberikan akses ke titik akses dan bucket yang mendasarinya.

> **Note**
>
> Agar sumber daya hanya dapat diakses dalam VPC, pastikan untuk membuat zona yang di-hosting pribadi untuk titik akhir VPC Anda. Untuk menggunakan zona yang di-hosting pribadi, modifikasi pengaturan VPC Anda sehingga atribut jaringan VPC `enableDnsHostnames` dan `enableDnsSupport` diatur ke `true`.

Contoh pernyataan kebijakan berikut ini mengonfigurasi titik akhir VPC untuk mengizinkan panggilan ke `GetObject` untuk bucket bernama `awsexamplebucket1` dan titik akses bernama `example-vpc-ap`.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Principal": "*",
        "Action": [
            "s3:GetObject"
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:s3:::awsexamplebucket1/*",
            "arn:aws:s3:us-west-2:123456789012:accesspoint/example-vpc-ap/object/*"
        ]
    }]
}
```

> **Note**
>
> Pernyataan `"Resource"` di dalam contoh ini menggunakan Amazon Resource Name (ARN) untuk menentukan titik akses. Untuk informasi selengkapnya tentang titik akses ARNs, lihatMereferensikan titik akses dengan ARNs, alias titik akses, atau gaya yang dihoskan secara virtual URIs.

Untuk informasi selengkapnya tentang kebijakan titik akhir VPC, lihat Menggunakan kebijakan titik akhir untuk Amazon S3 dalam Panduan Pengguna Amazon VPC.

Untuk tutorial tentang membuat titik akses dengan titik akhir VPC, lihat Mengelola akses Amazon S3 dengan titik akhir dan titik akses VPC.

Contoh: Membuat dan membatasi titik akses yang dilampirkan ke volume FSx untuk OpenZFS ke ID VPC

Anda dapat membuat titik akses yang dilampirkan ke volume FSx untuk OpenZFS menggunakan FSx konsol Amazon, AWS CLI, atau API. Setelah terpasang, Anda dapat menggunakan objek S3 APIs untuk mengakses data file Anda dari VPC tertentu.

Untuk petunjuk tentang membuat dan membatasi titik akses yang dilampirkan ke volume FSx untuk OpenZFS lihat, Membuat titik akses terbatas pada virtual private cloud (VPC) di for OpenZFS User Guide. FSx

Mengelola akses publik ke titik akses untuk ember tujuan umum

Titik akses Amazon S3 mendukung pengaturan independen blokir akses publik untuk setiap titik akses. Saat membuat titik akses, Anda dapat menentukan pengaturan blokir akses publik yang berlaku pada titik akses tersebut. Untuk setiap permintaan yang dibuat melalui titik akses, Amazon S3 mengevaluasi pengaturan blokir akses publik untuk titik akses tersebut, bucket yang mendasarinya, dan akun pemilik bucket. Jika salah satu pengaturan ini menunjukkan bahwa permintaan tersebut harus diblokir, Amazon S3 menolak permintaan tersebut.

Untuk informasi lebih lanjut tentang fitur Blokir Akses Publik S3, lihat Melakukan blok akses publik ke penyimpanan Amazon S3 Anda.

Important

- Semua pengaturan blokir akses publik diaktifkan secara default untuk titik akses baru. Anda harus secara eksplisit menonaktifkan pengaturan apa pun yang tidak Anda inginkan selama pembuatan titik akses.
- Anda tidak dapat mematikan pengaturan akses publik blok apa pun saat membuat atau menggunakan titik akses yang dilampirkan ke sistem FSx file Amazon.
- Setelah Anda membuat titik akses, Anda tidak dapat mengubah pengaturan blokir akses publik.

Example

Contoh: Membuat titik akses dengan Pengaturan Blok Akses Publik Khusus

Contoh ini membuat titik akses bernama `example-ap` untuk bucket *`amzn-s3-demo-bucket`* dalam akun `123456789012` dengan pengaturan Blokir Akses Publik non-default. Contoh ini kemudian mengambil konfigurasi titik akses baru untuk memverifikasi pengaturan Blokir Akses Publik miliknya.

AWS CLI

```
aws s3control create-access-point --name example-ap --account-id
 123456789012 --bucket amzn-s3-demo-bucket--public-access-block-configuration
 BlockPublicAcls=false,IgnorePublicAcls=false,BlockPublicPolicy=true,RestrictPublicBuckets=t
```

```
aws s3control get-access-point --name example-ap --account-id 123456789012

{
    "Name": "example-ap",
    "Bucket": "amzn-s3-demo-bucket",
    "NetworkOrigin": "Internet",
    "PublicAccessBlockConfiguration": {
        "BlockPublicAcls": false,
        "IgnorePublicAcls": false,
        "BlockPublicPolicy": true,
        "RestrictPublicBuckets": true
    },
    "CreationDate": "2019-11-27T00:00:00Z"
}
```

Mengelola jalur akses Amazon S3 Anda untuk bucket tujuan umum

Bagian ini menjelaskan cara mengelola jalur akses Amazon S3 Anda untuk bucket tujuan umum menggunakan AWS Management Console, AWS Command Line Interface, atau REST API. Untuk informasi tentang mengelola titik akses yang dilampirkan ke volume FSx untuk OpenZFS, lihat Mengelola titik akses Amazon S3 Anda di Panduan Pengguna untuk FSx OpenZFS.

> **Note**
>
> Anda hanya dapat menggunakan titik akses untuk melakukan pengoperasian pada objek. Anda tidak dapat menggunakan titik akses untuk melakukan operasi Amazon S3 lainnya, seperti menghapus bucket atau membuat konfigurasi Replikasi S3. Untuk mengetahui daftar lengkap pengoperasian S3 yang mendukung titik akses, lihat Kompatibilitas titik akses.

Topik

- Buat daftar titik akses Anda untuk ember tujuan umum
- Lihat detail untuk titik akses Anda untuk bucket tujuan umum
- Hapus titik akses Anda untuk bucket tujuan umum

Buat daftar titik akses Anda untuk ember tujuan umum

Bagian ini menjelaskan cara membuat daftar titik akses Anda untuk bucket tujuan umum menggunakan AWS Management Console, AWS Command Line Interface, atau REST API.

Menggunakan konsol S3

Untuk membuat daftar titik akses di Akun AWS

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi di bagian atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah yang ingin Anda daftarkan titik akses.
3. Di panel navigasi di sisi kiri konsol, pilih Access Points.
4. (Opsional) Cari titik akses berdasarkan nama. Hanya titik akses yang Anda pilih yang Wilayah AWS akan muncul di sini.

5. Pilih nama titik akses yang ingin Anda kelola atau gunakan.

Menggunakan AWS CLI

`list-access-points`Contoh perintah berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk daftar titik akses Anda.

Perintah berikut mencantumkan titik akses untuk Akun AWS *111122223333*.

```
aws s3control list-access-points --account-id 111122223333
```

Perintah berikut mencantumkan titik akses Akun AWS *111122223333* yang dilampirkan ke bucket*amzn-s3-demo-bucket*.

```
aws s3control list-access-points --account-id 111122223333 --bucket amzn-s3-demo-bucket
```

Untuk informasi dan contoh selengkapnya, lihat list-access-points di AWS CLI Referensi Perintah.

Penggunaan API REST

Anda dapat menggunakan REST API untuk membuat daftar titik akses Anda. Untuk informasi selengkapnya, lihat ListAccessPoints dalam Referensi API Amazon Simple Storage Service.

Lihat detail untuk titik akses Anda untuk bucket tujuan umum

Bagian ini menjelaskan cara melihat detail titik akses Anda untuk bucket tujuan umum menggunakan AWS Management Console, AWS Command Line Interface, atau REST API.

Menggunakan konsol S3

Untuk melihat detail titik akses Anda di Akun AWS

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi di bagian atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah yang ingin Anda daftarkan titik akses.
3. Di panel navigasi di sisi kiri konsol, pilih Access Points.

4. (Opsional) Cari titik akses berdasarkan nama. Hanya titik akses yang Anda pilih yang Wilayah AWS akan muncul di sini.
5. Pilih nama titik akses yang ingin Anda kelola atau gunakan.
6. Pilih tab Properties untuk melihat sumber data titik akses, ID akun, tanggal pembuatan Wilayah AWS, asal jaringan, URI S3, ARN, dan alias titik akses untuk titik akses yang dipilih.
7. Pilih tab Izin untuk melihat pengaturan blokir akses publik dan kebijakan titik akses untuk titik akses yang dipilih.

> **Note**
>
> Anda tidak dapat mengubah pengaturan akses publik blok apa pun untuk titik akses setelah titik akses dibuat.

Menggunakan AWS CLI

`get-access-point`Contoh perintah berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk melihat rincian untuk titik akses Anda.

Perintah berikut mencantumkan detail untuk titik akses *my-access-point* untuk Akun AWS *111122223333* dilampirkan ke bucket *amzn-s3-demo-bucket* S3.

```
aws s3control get-access-point --name my-access-point --account-id 111122223333
```

Contoh output:

```
{
    "Name": "my-access-point",
    "Bucket": "amzn-s3-demo-bucket",
    "NetworkOrigin": "Internet",
    "PublicAccessBlockConfiguration": {
        "BlockPublicAcls": true,
        "IgnorePublicAcls": true,
        "BlockPublicPolicy": true,
        "RestrictPublicBuckets": true
    },
    "CreationDate": "2016-08-29T22:57:52Z",
    "Alias": "my-access-point-u1ny6bhm7moymqx8cuon8o1g4mwikuse2a-s3alias",
```

```
    "AccessPointArn": "arn:aws:s3:Wilayah AWS:111122223333:accesspoint/my-access-
point",
    "Endpoints": {
        "ipv4": "s3-accesspoint.Wilayah AWS.amazonaws.com",
        "fips": "s3-accesspoint-fips.Wilayah AWS.amazonaws.com",
        "fips_dualstack": "s3-accesspoint-fips.dualstack.Wilayah AWS.amazonaws.com",
        "dualstack": "s3-accesspoint.dualstack.Wilayah AWS.amazonaws.com"
    },
    "BucketAccountId": "111122223333"
}
```

Perintah berikut mencantumkan rincian untuk titik akses *example-fsx-ap* untuk Akun AWS *444455556666*. Jalur akses ini dilampirkan ke sistem FSx file Amazon.

```
aws s3control get-access-point --name example-fsx-ap --account-id 444455556666
```

Contoh output:

```
{
    "Name": "example-fsx-ap",
    "Bucket": "",
    "NetworkOrigin": "Internet",
    "PublicAccessBlockConfiguration": {
        "BlockPublicAcls": true,
        "IgnorePublicAcls": true,
        "BlockPublicPolicy": true,
        "RestrictPublicBuckets": true
    },
    "CreationDate": "2025-01-19T14:16:12Z",
    "Alias": "example-fsx-ap-qrqbyebjtsxorhhaa5exx6r3q7-ext-s3alias",
    "AccessPointArn": "arn:aws:s3:Wilayah AWS:444455556666:accesspoint/example-fsx-ap",
    "Endpoints": {
        "ipv4": "s3-accesspoint.Wilayah AWS.amazonaws.com",
        "fips": "s3-accesspoint-fips.Wilayah AWS.amazonaws.com",
        "fips_dualstack": "s3-accesspoint-fips.dualstack.Wilayah AWS.amazonaws.com",
        "dualstack": "s3-accesspoint.dualstack.Wilayah AWS.amazonaws.com"
    },
    "DataSourceId": "arn:aws::fsx:Wilayah AWS:444455556666:file-system/
fs-5432106789abcdef0/volume/vol-0123456789abcdef0",
    "DataSourceType": "FSX_OPENZFS"
}
```

Untuk informasi dan contoh selengkapnya, lihat get-access-point di AWS CLI Referensi Perintah.

Penggunaan API REST

Anda dapat menggunakan REST API untuk melihat detail titik akses Anda. Untuk informasi selengkapnya, lihat GetAccessPoint dalam Referensi API Amazon Simple Storage Service.

Hapus titik akses Anda untuk bucket tujuan umum

Bagian ini menjelaskan cara menghapus titik akses Anda untuk bucket tujuan umum menggunakan AWS Management Console, AWS Command Line Interface, atau REST API.

Menggunakan konsol S3

Untuk menghapus titik akses Anda di Akun AWS

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi di bagian atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah yang ingin Anda daftarkan titik akses.
3. Di panel navigasi di sisi kiri konsol, pilih Access Points.
4. (Opsional) Cari titik akses berdasarkan nama. Hanya titik akses yang Anda pilih yang Wilayah AWS akan muncul di sini.
5. Pilih nama titik akses yang ingin Anda kelola atau gunakan.
6. Dari halaman Access Point, pilih Hapus untuk menghapus titik akses yang telah Anda pilih.
7. Untuk mengonfirmasi penghapusan, ketik nama titik akses dan pilih Hapus.

Menggunakan AWS CLI

`delete-access-point`Contoh perintah berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk menghapus titik akses Anda.

Perintah berikut menghapus titik akses *`my-access-point`* untuk Akun AWS *`111122223333`*.

```
aws s3control delete-access-point --name my-access-point --account-id 111122223333
```

Untuk informasi dan contoh selengkapnya, lihat delete-access-point di AWS CLI Referensi Perintah.

Penggunaan API REST

Anda dapat menggunakan REST API untuk melihat detail titik akses Anda. Untuk informasi selengkapnya, lihat DeleteAccessPoint dalam Referensi API Amazon Simple Storage Service.

Menggunakan jalur akses Amazon S3 untuk bucket tujuan umum

Contoh berikut menunjukkan cara menggunakan titik akses untuk bucket tujuan umum dengan operasi yang kompatibel di Amazon S3.

> **Note**
>
> S3 secara otomatis menghasilkan alias titik akses untuk semua titik akses dan alias ini dapat digunakan di mana saja nama bucket digunakan untuk melakukan operasi tingkat objek. Untuk informasi selengkapnya, lihat Alias titik akses.

Anda hanya dapat menggunakan titik akses untuk bucket tujuan umum untuk melakukan operasi pada objek. Anda tidak dapat menggunakan titik akses untuk melakukan operasi Amazon S3 lainnya, seperti mengubah atau menghapus bucket. Untuk mengetahui daftar lengkap pengoperasian S3 yang mendukung titik akses, lihat Kompatibilitas titik akses.

Topik

- Daftar objek melalui titik akses untuk bucket tujuan umum
- Unduh objek melalui titik akses untuk bucket tujuan umum
- Konfigurasikan daftar kontrol akses (ACLs) melalui titik akses untuk bucket tujuan umum
- Unggah objek melalui titik akses untuk bucket tujuan umum
- Menambahkan tag-set melalui titik akses untuk bucket tujuan umum
- Menghapus objek melalui titik akses untuk bucket tujuan umum

Daftar objek melalui titik akses untuk bucket tujuan umum

Bagian ini menjelaskan cara membuat daftar objek Anda melalui titik akses untuk bucket tujuan umum menggunakan AWS Management Console, AWS Command Line Interface, atau REST API.

Menggunakan konsol S3

Untuk membuat daftar objek Anda melalui titik akses di Akun AWS

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi di bagian atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah yang ingin Anda daftarkan titik akses.
3. Di panel navigasi di sisi kiri konsol, pilih Access Points.
4. (Opsional) Cari titik akses berdasarkan nama. Hanya titik akses yang Anda pilih yang Wilayah AWS akan muncul di sini.
5. Pilih nama titik akses yang ingin Anda kelola atau gunakan.
6. Di bawah tab Objects, Anda dapat melihat nama objek yang ingin Anda akses melalui titik akses. Saat Anda menggunakan titik akses, Anda hanya dapat melakukan operasi objek yang diizinkan oleh izin titik akses.

> **Note**
>
> - Tampilan konsol selalu menampilkan semua objek dalam bucket. Menggunakan titik akses seperti yang dijelaskan dalam prosedur ini membatasi operasi yang dapat Anda lakukan pada objek tersebut, namun tidak membatasi apakah Anda dapat melihat keberadaan objek tersebut di dalam bucket.
> - AWS Management Console Tidak mendukung penggunaan titik akses virtual private cloud (VPC) untuk mengakses sumber daya bucket. Untuk mengakses sumber daya bucket dari titik akses VPC, gunakan REST Amazon S3 AWS CLI AWS SDKs, atau Amazon S3. APIs

Menggunakan AWS CLI

`list-objects-v2`Contoh perintah berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk daftar objek Anda melalui titik akses.

Perintah berikut mencantumkan objek untuk Akun AWS *`111122223333`* menggunakan titik akses*`my-access-point`*.

```
aws s3api list-objects-v2 --bucket arn:aws:s3:Wilayah AWS:111122223333:accesspoint/my-access-point
```

> **Note**
>
> S3 secara otomatis menghasilkan alias titik akses untuk semua titik akses dan alias ini dapat digunakan di mana saja nama bucket digunakan untuk melakukan operasi tingkat objek. Untuk informasi selengkapnya, lihat Alias titik akses.

Untuk informasi dan contoh selengkapnya, lihat list-access-points di AWS CLI Referensi Perintah.

Penggunaan API REST

Anda dapat menggunakan REST API untuk membuat daftar titik akses Anda. Untuk informasi selengkapnya, lihat ListObjectsV2 dalam Referensi API Amazon Simple Storage Service.

Unduh objek melalui titik akses untuk bucket tujuan umum

Bagian ini menjelaskan cara mengunduh objek melalui titik akses untuk bucket tujuan umum menggunakan AWS Management Console, AWS Command Line Interface, atau REST API.

Menggunakan konsol S3

Untuk mengunduh objek melalui titik akses di Akun AWS

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi di bagian atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah yang ingin Anda daftarkan titik akses.
3. Di panel navigasi di sisi kiri konsol, pilih Access Points.
4. (Opsional) Cari titik akses berdasarkan nama. Hanya titik akses yang Anda pilih yang Wilayah AWS akan muncul di sini.
5. Pilih nama titik akses yang ingin Anda kelola atau gunakan.
6. Di bawah tab Objek, pilih nama objek yang ingin Anda unduh.
7. Pilih Unduh.

Menggunakan AWS CLI

`get-object`Contoh perintah berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk men-download objek melalui titik akses.

Perintah berikut mengunduh objek `puppy.jpg` untuk Akun AWS *111122223333* menggunakan titik akses*my-access-point*. Anda harus menyertakan`outfile`, yang merupakan nama file untuk objek yang diunduh, seperti*my_downloaded_image.jpg*.

```
aws s3api get-object --bucket arn:aws:s3:Wilayah AWS:111122223333:accesspoint/my-
access-point --key puppy.jpg my_downloaded_image.jpg
```

> **Note**
>
> S3 secara otomatis menghasilkan alias titik akses untuk semua titik akses dan alias ini dapat digunakan di mana saja nama bucket digunakan untuk melakukan operasi tingkat objek. Untuk informasi selengkapnya, lihat Alias titik akses.

Untuk informasi dan contoh selengkapnya, lihat get-object di AWS CLI Referensi Perintah.

Penggunaan API REST

Anda dapat menggunakan REST API untuk mengunduh objek melalui titik akses. Untuk informasi selengkapnya, lihat GetObject dalam Referensi API Amazon Simple Storage Service.

Menggunakan AWS SDKs

Anda dapat menggunakan AWS SDK untuk Python untuk mengunduh objek melalui titik akses.

Python

Dalam contoh berikut, file bernama *hello.txt* diunduh untuk AWS akun *111122223333* menggunakan titik akses bernama*my-access-point*.

```
import boto3
s3 = boto3.client('s3')
s3.download_file('arn:aws:s3:us-east-1:111122223333:accesspoint/my-access-point',
 'hello.txt', '/tmp/hello.txt')
```

Konfigurasikan daftar kontrol akses (ACLs) melalui titik akses untuk bucket tujuan umum

Bagian ini menjelaskan cara mengonfigurasi ACLs melalui titik akses untuk bucket tujuan umum menggunakan AWS Management Console, AWS Command Line Interface, atau REST API. Untuk informasi lebih lanjut tentang ACLs, lihatGambaran umum daftar kontrol akses (ACL).

Menggunakan konsol S3

Untuk mengkonfigurasi ACLs melalui titik akses di Akun AWS

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi di bagian atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah yang ingin Anda daftarkan titik akses.
3. Di panel navigasi di sisi kiri konsol, pilih Access Points.
4. (Opsional) Cari titik akses berdasarkan nama. Hanya titik akses yang Anda pilih yang Wilayah AWS akan muncul di sini.
5. Pilih nama titik akses yang ingin Anda kelola atau gunakan.
6. Di bawah tab Objects, pilih nama objek yang ingin Anda konfigurasikan ACL.
7. Di bawah tab Izin, pilih Edit untuk mengkonfigurasi objek ACL.

> **Note**
>
> Saat ini, Amazon S3 tidak mendukung perubahan pengaturan akses publik blokir titik akses setelah titik akses dibuat.

Menggunakan AWS CLI

`put-object-acl`Contoh perintah berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk mengkonfigurasi izin akses melalui titik akses menggunakan ACL.

Perintah berikut menerapkan ACL ke objek yang ada `puppy.jpg` melalui titik akses yang dimiliki oleh Akun AWS *`111122223333`*.

```
aws s3api put-object-acl --bucket arn:aws:s3:Wilayah AWS:111122223333:accesspoint/my-access-point --key puppy.jpg --acl private
```

> **Note**
>
> S3 secara otomatis menghasilkan alias titik akses untuk semua titik akses dan alias ini dapat digunakan di mana saja nama bucket digunakan untuk melakukan operasi tingkat objek. Untuk informasi selengkapnya, lihat Alias titik akses.

Untuk informasi dan contoh selengkapnya, lihat put-object-acl di AWS CLI Referensi Perintah.

Penggunaan API REST

Anda dapat menggunakan REST API untuk mengonfigurasi izin akses melalui titik akses menggunakan ACL. Untuk informasi selengkapnya, lihat PutObjectAcl dalam Referensi API Amazon Simple Storage Service.

Unggah objek melalui titik akses untuk bucket tujuan umum

Bagian ini menjelaskan cara mengunggah objek melalui titik akses untuk bucket tujuan umum menggunakan AWS Management Console, AWS Command Line Interface, atau REST API.

Menggunakan konsol S3

Untuk mengunggah objek melalui titik akses di Akun AWS

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi di bagian atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah yang ingin Anda daftarkan titik akses.
3. Di panel navigasi di sisi kiri konsol, pilih Access Points.
4. (Opsional) Cari titik akses berdasarkan nama. Hanya titik akses yang Anda pilih yang Wilayah AWS akan muncul di sini.
5. Pilih nama titik akses yang ingin Anda kelola atau gunakan.
6. Di bawah tab Objek, pilih Unggah.
7. Seret dan lepas file dan folder yang ingin Anda unggah di sini, atau pilih Tambahkan file atau Tambahkan folder.

> **Note**
>
> Ukuran maksimum file yang dapat Anda unggah menggunakan konsol Amazon S3 adalah 160 GB. Untuk mengunggah file yang lebih besar dari 160 GB, gunakan AWS Command Line Interface (AWS CLI) AWS SDKs, atau Amazon S3 REST API.

8. Untuk mengubah izin daftar kontrol akses, pilih Izin.
9. Di bagian bawah Daftar kontrol akses (ACL), edit izinnya.

   Untuk informasi selengkapnya tentang izin akses objek, lihat Menggunakan konsol S3 untuk mengatur izin ACL untuk objek. Anda dapat memberikan akses baca ke objek Anda kepada publik (semua orang di dunia) untuk semua file yang Anda unggah. Namun, kami merekomendasikan untuk tidak mengubah pengaturan default untuk mengakses baca publik. Pemberian akses baca publik berlaku untuk sebagian kecil kasus penggunaan, seperti saat bucket digunakan untuk situs web. Anda dapat selalu mengubah izin objek setelah Anda mengunggah objek.
10. Untuk mengonfigurasi properti tambahan lainnya, pilih Properti.
11. Di bagian bawah Kelas penyimpanan pilih kelas penyimpanan untuk file yang Anda unggah.

    Untuk informasi selengkapnya tentang kelas penyimpanan, lihat Memahami dan mengelola kelas penyimpanan Amazon S3.
12. Untuk memperbarui pengaturan enkripsi untuk objek Anda, di bagian bawah Pengaturan enkripsi di sisi server, lakukan hal berikut.

    a. Pilih Tentukan kunci enkripsi.

    b. Di bawah Pengaturan enkripsi, pilih Gunakan pengaturan bucket untuk enkripsi default, atau Ganti pengaturan bucket untuk enkripsi default.

    c. Jika Anda memilih Ganti pengaturan bucket untuk enkripsi default, Anda harus mengonfigurasi pengaturan enkripsi berikut.

    - Untuk mengenkripsi file yang diunggah dengan menggunakan kunci yang dikelola oleh Amazon S3, pilih Kunci yang dikelola Amazon S3 (SSE-S3).

      Untuk informasi selengkapnya, lihat Menggunakan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3).

- Untuk mengenkripsi file yang diunggah dengan menggunakan kunci yang disimpan di AWS Key Management Service (AWS KMS), pilih AWS Key Management Service kunci (SSE-KMS). Lalu pilih salah satu opsi berikut ini untuk kunci AWS KMS :
  - Untuk memilih dari daftar kunci KMS yang tersedia, pilih Pilih dari AWS KMS keys Anda, dan pilih Kunci KMS Anda dari daftar kunci yang tersedia.

    Kunci Kunci yang dikelola AWS (`aws/s3`) dan kunci terkelola pelanggan Anda muncul dalam daftar ini. Untuk informasi selengkapnya tentang CMK, lihat Kunci pelanggan dan AWS kunci di AWS Key Management Service Panduan Pengembang.
  - Untuk memasukkan ARN kunci KMS, pilih Masukkan AWS KMS key ARN, lalu masukkan ARN kunci KMS Anda di bidang yang muncul.
  - Untuk membuat kunci terkelola pelanggan baru di AWS KMS konsol, pilih Buat kunci KMS.

    Untuk informasi selengkapnya tentang membuat AWS KMS key, lihat Membuat kunci di Panduan AWS Key Management Service Pengembang.

    > **Important**
    >
    > Anda hanya dapat menggunakan tombol KMS yang tersedia Wilayah AWS sama dengan bucket. Konsol Amazon S3 hanya mencantumkan kunci 100 KMS pertama di Wilayah yang sama dengan bucket. Untuk menggunakan kunci KMS yang tidak terdaftar, Anda harus memasukkan ARN kunci KMS Anda. Jika Anda ingin menggunakan kunci KMS yang dimiliki oleh akun lain, Anda harus terlebih dahulu memiliki izin untuk menggunakan kunci tersebut, dan kemudian Anda harus memasukkan ARN kunci KMS.
    > Amazon S3 hanya mendukung kunci KMS enkripsi simetris, dan tidak mendukung kunci KMS asimetris. Untuk informasi selengkapnya, lihat Mengidentifikasi tombol KMS simetris dan asimetris dalam Panduan Pengembang AWS Key Management Service .

13. Untuk menggunakan checksum tambahan, pilih Aktif. Kemudian untuk Fungsi checksum, pilih fungsi yang ingin Anda gunakan. Amazon S3 menghitung dan menyimpan nilai checksum setelah menerima objek secara keseluruhan. Anda dapat menggunakan kotak Nilai prakalkulasi untuk memberikan nilai yang telah dihitung sebelumnya. Jika sudah, Amazon S3 membandingkan nilai yang Anda berikan dengan nilai yang dihitung. Jika kedua nilai tidak cocok, Amazon S3 menghasilkan kesalahan.

Checksum tambahan memungkinkan Anda menentukan algoritma checksum yang ingin Anda gunakan untuk memverifikasi data Anda. Untuk informasi selengkapnya tentang checksum tambahan, lihat Memeriksa integritas objek di Amazon S3.

14. Untuk menambahkan tanda ke semua objek yang Anda unggah, pilih Tambahkan tag. Masukkan nama tag di bidang Kunci. Masukkan nilai untuk tanda.

    Penandaan objek memberi Anda cara untuk mengategorikan penyimpanan. Setiap tag adalah pasangan nilai kunci. Kunci dan nilai tag peka huruf besar dan kecil. Anda dapat membuat hingga 10 tanda per objek. Kunci tanda dapat terdiri dari hingga 128 karakter Unicode, dan nilai tanda dapat terdiri dari hingga 256 karakter Unicode. Untuk informasi selengkapnya tentang tag objek, lihat Mengategorikan penyimpanan Anda menggunakan tag.

15. Untuk menambahkan metadata, pilih Tambahkan metadata.

    a. Di bawah Jenis, pilih Ditentukan sistem atau Ditentukan pengguna.

       Untuk metadata yang ditentukan sistem, Anda dapat memilih header HTTP umum, seperti Jenis Konten dan Konten-Disposisi. Untuk daftar metadata yang ditentukan sistem, serta informasi tentang apakah Anda dapat menambahkan nilai, lihat Metadata objek yang ditentukan sistem. Metadata apa pun yang dimulai dengan prefiks `x-amz-meta-` diperlakukan sebagai metadata yang ditentukan pengguna. Metadata yang ditentukan pengguna disimpan bersama objek, dan dikembalikan saat Anda mengunduh objek tersebut. Baik kunci maupun nilainya harus sesuai dengan standar US-ASCII. Metadata yang ditentukan pengguna dapat berukuran sebesar 2 KB. Untuk informasi selengkapnya tentang metadata yang ditentukan sistem dan yang ditentukan pengguna, lihat Bekerja dengan metadata objek.

    b. Untuk Kunci, pilih kuncinya.

    c. Ketikkan nilai untuk kunci tersebut.

16. Untuk mengunggah objek Anda, pilih Unggah.

    Amazon S3 mengunggah objek Anda. Setelah unggahannya selesai, Anda dapat melihat pesan sukses di halaman Unggah: status.

Menggunakan AWS CLI

`put-object`Contoh perintah berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk meng-upload objek melalui titik akses.

Perintah berikut mengunggah objek `puppy.jpg` untuk Akun AWS *111122223333* menggunakan titik *my-access-point* akses.

```
aws s3api put-object --bucket arn:aws:s3:Wilayah AWS:111122223333:accesspoint/my-access-point --key puppy.jpg --body puppy.jpg
```

> **Note**
>
> S3 secara otomatis menghasilkan alias titik akses untuk semua titik akses dan alias titik akses dapat digunakan di mana saja nama bucket digunakan untuk melakukan operasi tingkat objek. Untuk informasi selengkapnya, lihat Alias titik akses.

Untuk informasi dan contoh selengkapnya, lihat put-object di AWS CLI Referensi Perintah.

Penggunaan API REST

Anda dapat menggunakan REST API untuk mengunggah objek melalui titik akses. Untuk informasi selengkapnya, lihat PutObject dalam Referensi API Amazon Simple Storage Service.

Menggunakan AWS SDKs

Anda dapat menggunakan AWS SDK untuk Python untuk mengunggah objek melalui titik akses.

Python

Dalam contoh berikut, file bernama *hello.txt* diunggah untuk AWS akun *111122223333* menggunakan titik akses bernama*my-access-point*.

```
import boto3
s3 = boto3.client('s3')
s3.upload_file('/tmp/hello.txt', 'arn:aws:s3:us-east-1:111122223333:accesspoint/my-access-point', 'hello.txt')
```

Menambahkan tag-set melalui titik akses untuk bucket tujuan umum

Bagian ini menjelaskan cara menambahkan kumpulan tag melalui titik akses untuk bucket tujuan umum menggunakan AWS Management Console, AWS Command Line Interface, atau REST API. Untuk informasi selengkapnya, lihat Mengategorikan penyimpanan Anda menggunakan tag.

Menggunakan konsol S3

Untuk menambahkan tag-set melalui titik akses di Akun AWS

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi di bagian atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah yang ingin Anda daftarkan titik akses.
3. Di panel navigasi di sisi kiri konsol, pilih Access Points.
4. (Opsional) Cari titik akses berdasarkan nama. Hanya titik akses yang Anda pilih yang Wilayah AWS akan muncul di sini.
5. Pilih nama titik akses yang ingin Anda kelola atau gunakan.
6. Di bawah tab Objects, pilih nama objek yang ingin Anda tambahkan tag-set.
7. Di bawah tab Properties, temukan sub-header Tag dan pilih Edit.
8. Tinjau objek yang terdaftar, dan pilih Tambah tag.
9. Setiap tag objek adalah pasangan nilai kunci. Masukkan Kunci dan Nilai. Untuk menambahkan tag lainnya, pilih Tambahkan tag.

   Anda dapat memasukkan hingga 10 tag untuk objek.
10. Pilih Simpan perubahan.

Menggunakan AWS CLI

`put-object-tagging`Contoh perintah berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk menambahkan tag-set melalui titik akses.

Perintah berikut menambahkan tag-set untuk objek yang ada `puppy.jpg` menggunakan titik akses. *`my-access-point`*

```
aws s3api put-object-tagging --bucket arn:aws:s3:Wilayah
 AWS:111122223333:accesspoint/my-access-point --key puppy.jpg --tagging
 TagSet=[{Key="animal",Value="true"}]
```

> **Note**
>
> S3 secara otomatis menghasilkan alias titik akses untuk semua titik akses dan alias titik akses dapat digunakan di mana saja nama bucket digunakan untuk melakukan operasi tingkat objek. Untuk informasi selengkapnya, lihat Alias titik akses.

Untuk informasi dan contoh selengkapnya, lihat put-object-tagging di AWS CLI Referensi Perintah.

Penggunaan API REST

Anda dapat menggunakan REST API untuk menambahkan tag-set ke objek melalui titik akses. Untuk informasi selengkapnya, lihat PutObjectTagging dalam Referensi API Amazon Simple Storage Service.

Menghapus objek melalui titik akses untuk bucket tujuan umum

Bagian ini menjelaskan cara menghapus objek melalui titik akses untuk bucket tujuan umum menggunakan AWS Management Console, AWS Command Line Interface, atau REST API.

Menggunakan konsol S3

Untuk menghapus objek atau objek melalui titik akses di Akun AWS

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi di bagian atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah yang ingin Anda daftarkan titik akses.
3. Di panel navigasi di sisi kiri konsol, pilih Access Points.
4. (Opsional) Cari titik akses berdasarkan nama. Hanya titik akses yang Anda pilih yang Wilayah AWS akan muncul di sini.
5. Pilih nama titik akses yang ingin Anda kelola atau gunakan.
6. Di bawah tab Objek, pilih nama objek atau objek yang ingin Anda hapus.
7. Tinjau objek yang terdaftar untuk dihapus, dan ketik hapus di kotak konfirmasi.
8. Pilih Hapus objek.

Menggunakan AWS CLI

`delete-object`Contoh perintah berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk menghapus objek melalui titik akses.

Perintah berikut menghapus objek yang ada `puppy.jpg` menggunakan titik *`my-access-point`* akses.

```
aws s3api delete-object --bucket arn:aws:s3:Wilayah AWS:111122223333:accesspoint/my-access-point --key puppy.jpg
```

> **Note**
>
> S3 secara otomatis menghasilkan alias titik akses untuk semua titik akses dan alias titik akses dapat digunakan di mana saja nama bucket digunakan untuk melakukan operasi tingkat objek. Untuk informasi selengkapnya, lihat Alias titik akses.

Untuk informasi dan contoh selengkapnya, lihat delete-object di AWS CLI Referensi Perintah.

Penggunaan API REST

Anda dapat menggunakan REST API untuk menghapus objek melalui titik akses. Untuk informasi selengkapnya, lihat DeleteObject dalam Referensi API Amazon Simple Storage Service.

Mengelola akses dengan S3 Access Grants

Untuk mematuhi prinsip hak akses paling rendah, Anda menentukan akses terperinci ke data Amazon S3 Anda berdasarkan aplikasi, persona, grup, atau unit organisasi. Anda dapat menggunakan berbagai pendekatan untuk mencapai akses terperinci ke data Anda di Amazon S3, tergantung pada skala dan kompleksitas pola akses.

Pendekatan paling sederhana untuk mengelola akses ke small-to-medium jumlah kumpulan data di Amazon S3 AWS Identity and Access Management by (IAM) prinsipal adalah dengan menentukan kebijakan izin IAM dan kebijakan bucket S3. Strategi ini berhasil, selama kebijakan yang diperlukan sesuai dengan batas ukuran kebijakan bucket S3 (20 KB) dan kebijakan IAM (5 KB), dan dalam jumlah pengguna utama IAM yang diizinkan per akun.

Karena jumlah set data dan kasus penggunaan Anda meningkat, Anda mungkin memerlukan lebih banyak ruang kebijakan. Pendekatan yang menawarkan lebih banyak ruang secara signifikan

untuk pernyataan kebijakan adalah dengan menggunakan Titik Akses S3 sebagai titik akhir tambahan untuk bucket S3, karena setiap titik akses dapat memiliki kebijakannya sendiri. Anda dapat menentukan pola kontrol akses yang cukup terperinci, karena Anda dapat memiliki ribuan titik akses Wilayah AWS per akun, dengan kebijakan hingga 20 KB untuk setiap titik akses. Meskipun S3 Access Points meningkatkan jumlah ruang kebijakan yang tersedia, diperlukan mekanisme bagi klien untuk menemukan titik akses yang tepat untuk set data yang tepat.

Pendekatan ketiga adalah menerapkan pola broker sesi IAM, di mana Anda menerapkan logika keputusan akses dan secara dinamis menghasilkan kredensial sesi IAM jangka pendek untuk setiap sesi akses. Sementara pendekatan broker sesi IAM mendukung pola dan skala izin dinamis yang sewenang-wenang secara efektif, Anda harus membangun logika pola akses.

Alih-alih menggunakan pendekatan ini, Anda dapat menggunakan S3 Access Grants untuk mengelola akses ke data Amazon S3. S3 Access Grants menyediakan model yang disederhanakan untuk menentukan izin akses ke data di Amazon S3 berdasarkan prefiks, bucket, atau objek. Selain itu, Anda dapat menggunakan S3 Access Grants untuk memberikan akses ke kedua pengguna utama IAM dan langsung ke pengguna atau grup dari direktori perusahaan Anda.

Anda biasanya menentukan izin untuk data di Amazon S3 dengan memetakan pengguna dan grup ke set data. Anda dapat menggunakan S3 Access Grants untuk menentukan pemetaan akses langsung prefiks S3 ke pengguna dan peran dalam bucket dan objek Amazon S3. Dengan skema akses yang disederhanakan di S3 Access Grants, Anda dapat memberikan akses read-only, write-only, atau read-write berdasarkan prefiks per-S3 ke kedua pengguna utama IAM dan langsung ke pengguna atau grup dari direktori perusahaan. Dengan kemampuan S3 Access Grants ini, aplikasi dapat meminta data dari Amazon S3 atas nama pengguna aplikasi yang diautentikasi saat ini.

Saat Anda mengintegrasikan S3 Access Grants dengan fitur propagasi identitas tepercaya AWS IAM Identity Center, aplikasi Anda dapat mengajukan permintaan ke Layanan AWS (termasuk S3 Access Grants) secara langsung atas nama pengguna direktori perusahaan yang diautentikasi. Aplikasi Anda tidak perlu lagi memetakan pengguna terlebih dahulu ke pengguna utama IAM. Selain itu, karena identitas pengguna akhir disebarkan sampai ke Amazon S3, mengaudit pengguna mana yang mengakses S3 Object mana yang disederhanakan. Anda tidak perlu lagi merekonstruksi hubungan antara pengguna yang berbeda dan sesi IAM. Saat Anda menggunakan S3 Access Grants dengan propagasi identitas tepercaya IAM Identity Center, setiap AWS CloudTrailperistiwa data untuk Amazon S3 berisi referensi langsung ke pengguna akhir atas nama siapa data tersebut diakses.

Propagasi identitas tepercaya adalah AWS IAM Identity Center fitur yang Layanan AWS dapat digunakan oleh administrator terhubung untuk memberikan dan mengaudit akses ke data layanan. Akses ke data ini didasarkan pada atribut pengguna seperti asosiasi grup. Menyiapkan propagasi

identitas tepercaya memerlukan kolaborasi antara administrator yang terhubung Layanan AWS dan administrator Pusat Identitas IAM. Untuk informasi lebih lanjut, lihat Prasyarat dan pertimbangan.

Untuk informasi selengkapnya tentang S3 Access Grants, lihat topik-topik berikut.

Topik

- Konsep S3 Access Grants
- S3 Access Grants dan identitas direktori perusahaan
- Memulai S3 Access Grants
- Bekerja dengan instans S3 Access Grants
- Bekerja dengan lokasi S3 Access Grants
- Bekerja dengan hibah di S3 Access Grants
- Mendapatkan data S3 menggunakan hibah akses
- Akses S3 memberikan akses lintas akun
- Menggunakan AWS tag dengan S3 Access Grants
- Batasan S3 Access Grants
- Integrasi S3 Access Grants

Konsep S3 Access Grants

Alur kerja Hibah Akses S3

Alur kerja S3 Access Grants adalah:

1. Buat instance S3 Access Grants. Lihat Bekerja dengan instans S3 Access Grants.
2. Dalam instans S3 Access Grants Anda, daftarkan lokasi di data Amazon S3 Anda, dan petakan lokasi ini AWS Identity and Access Management ke peran (IAM). Lihat Mendaftarkan lokasi.
3. Buat hibah untuk penerima hibah, yang memberi penerima hibah akses ke sumber daya S3 Anda. Lihat Bekerja dengan hibah di S3 Access Grants.
4. Penerima hibah meminta kredensil sementara dari S3 Access Grants. Lihat Minta akses ke data Amazon S3 melalui S3 Access Grants.
5. Penerima hibah mengakses data S3 menggunakan kredensil sementara tersebut. Lihat Mengakses data S3 menggunakan kredensil yang dijual oleh S3 Access Grants.

Untuk informasi selengkapnya, lihat Memulai S3 Access Grants.

Instans S3 Access Grants

Instance S3 Access Grants adalah wadah logis untuk hibah individu. Saat Anda membuat instance S3 Access Grants, Anda harus menentukan. Wilayah AWS Masing-masing Wilayah AWS di dalam Anda Akun AWS dapat memiliki satu instance S3 Access Grants. Untuk informasi selengkapnya, lihat Bekerja dengan instans S3 Access Grants.

Jika Anda ingin menggunakan S3 Access Grants untuk memberikan akses ke identitas pengguna dan grup dari direktori perusahaan Anda, Anda juga harus mengaitkan instance S3 Access Grants Anda dengan sebuah instance. AWS IAM Identity Center Untuk informasi selengkapnya, lihat S3 Access Grants dan identitas direktori perusahaan.

Instance S3 Access Grants yang baru dibuat kosong. Anda harus mendaftarkan lokasi dalam instance, yang dapat berupa jalur default S3 (`s3://`), bucket, atau awalan dalam bucket. Setelah Anda mendaftarkan setidaknya satu lokasi, Anda dapat membuat hibah akses yang memberikan akses ke data di lokasi terdaftar ini.

Lokasi

Akses S3 Memberikan lokasi memetakan bucket atau awalan ke peran (IAM). AWS Identity and Access Management S3 Access Grants mengasumsikan peran IAM ini untuk menjual kredensil sementara kepada penerima hibah yang mengakses lokasi tertentu. Anda harus terlebih dahulu mendaftarkan setidaknya satu lokasi di instans S3 Access Grants sebelum Anda dapat membuat hibah akses.

Kami menyarankan Anda mendaftarkan lokasi default (`s3://`) dan memetakannya ke peran IAM. Lokasi di jalur S3 default (`s3://`) mencakup akses ke semua bucket S3 Anda di akun Anda Wilayah AWS . Saat membuat hibah akses, Anda dapat mempersempit cakupan hibah menjadi bucket, awalan, atau objek di dalam lokasi default.

Kasus penggunaan manajemen akses yang lebih kompleks mungkin mengharuskan Anda untuk mendaftar lebih dari lokasi default. Beberapa contoh kasus penggunaan tersebut adalah:

- Misalkan lokasi terdaftar di instance S3 Access Grants Anda dengan peran IAM yang dipetakan ke sana, tetapi peran IAM ini ditolak akses ke awalan tertentu di dalam bucket. *amzn-s3-demo-bucket* Dalam hal ini, Anda dapat mendaftarkan awalan yang tidak dapat diakses oleh peran IAM sebagai lokasi terpisah dan memetakan lokasi tersebut ke peran IAM yang berbeda dengan akses yang diperlukan.
- Misalkan Anda ingin membuat hibah yang membatasi akses hanya ke pengguna dalam titik akhir virtual private cloud (VPC). Dalam hal ini, Anda dapat mendaftarkan lokasi untuk bucket

di mana peran IAM membatasi akses ke titik akhir VPC. Kemudian, ketika penerima hibah meminta kredensil S3 Access Grants, S3 Access Grants mengasumsikan peran IAM lokasi untuk menjual kredenal sementara. Kredensi ini akan menolak akses ke bucket tertentu kecuali penelepon berada dalam titik akhir VPC. Izin penolakan ini diterapkan selain izin BACA, TULIS, atau READWRITE reguler yang ditentukan dalam hibah.

Jika kasus penggunaan Anda mengharuskan Anda mendaftarkan beberapa lokasi di instans S3 Access Grants, Anda dapat mendaftarkan salah satu dari berikut ini:

- Lokasi S3 default () `s3://`
- Ember (misalnya,*`amzn-s3-demo-bucket`*) atau beberapa ember
- Sebuah ember dan awalan (misalnya,*`amzn-s3-demo-bucket`*`/prefix*`) atau beberapa awalan

Untuk jumlah maksimum lokasi yang dapat Anda daftarkan di instans S3 Access Grants, lihat. Batasan S3 Access Grants Untuk informasi selengkapnya tentang mendaftarkan lokasi S3 Access Grants, lihat. Mendaftarkan lokasi

Setelah Anda mendaftarkan lokasi pertama di instans S3 Access Grants, instans Anda masih tidak memiliki hibah akses individual di dalamnya. Jadi, belum ada akses yang diberikan ke data S3 Anda. Anda sekarang dapat membuat hibah akses untuk memberikan akses. Untuk informasi selengkapnya tentang membuat hibah, lihatBekerja dengan hibah di S3 Access Grants.

Izin

Hibah individu dalam instans S3 Access Grants memungkinkan identitas tertentu—prinsipal IAM, atau pengguna atau grup di direktori perusahaan—untuk mendapatkan akses di dalam lokasi yang terdaftar di instans S3 Access Grants Anda.

Saat Anda membuat hibah, Anda tidak perlu memberikan akses ke seluruh lokasi yang terdaftar. Anda dapat mempersempit ruang lingkup akses hibah di dalam suatu lokasi. Jika lokasi terdaftar adalah jalur S3 default (`s3://`), Anda harus mempersempit cakupan hibah ke bucket, awalan dalam bucket, atau objek tertentu. Jika lokasi hibah yang terdaftar adalah bucket atau awalan, maka Anda dapat memberikan akses ke seluruh bucket atau awalan, atau Anda dapat secara opsional mempersempit cakupan hibah ke awalan, subprefiks, atau objek.

Dalam hibah, Anda juga mengatur tingkat akses hibah ke READ, WRITE, atau READWRITE. Misalkan Anda memiliki hibah yang memberi grup direktori perusahaan `01234567-89ab-cdef-0123-456789abcdef` READ akses ke buckets3://*`amzn-s3-demo-bucket`*`/projects/items/*`. Pengguna dalam grup ini dapat memiliki akses BACA ke setiap objek

yang memiliki nama kunci objek yang dimulai dengan awalan `projects/items/` di bucket bernama*`amzn-s3-demo-bucket`*.

Untuk jumlah maksimum hibah yang dapat Anda buat di instance S3 Access Grants, lihat. Batasan S3 Access Grants Untuk informasi selengkapnya tentang membuat hibah, lihatMembuat pemberian.

Akses S3 memberikan kredensial sementara

Setelah Anda membuat hibah, aplikasi resmi yang menggunakan identitas yang ditentukan dalam hibah dapat meminta kredensi just-in-time akses. Untuk melakukan ini, aplikasi memanggil operasi API GetDataAccessS3. Penerima hibah dapat menggunakan operasi API ini untuk meminta akses ke data S3 yang telah Anda bagikan dengan mereka.

Instans S3 Access Grants mengevaluasi permintaan `GetDataAccess` terhadap pemberian yang dimilikinya. Jika ada hibah yang cocok untuk pemohon, S3 Access Grants mengasumsikan peran IAM yang terkait dengan lokasi terdaftar dari hibah yang cocok. S3 Access Grants mencakup izin kredenal sementara untuk hanya mengakses bucket, awalan, atau objek S3 yang ditentukan oleh cakupan hibah.

Waktu kedaluwarsa kredenal akses sementara default menjadi 1 jam, tetapi Anda dapat mengaturnya ke nilai apa pun dari 15 menit hingga 12 jam. Lihat sesi durasi maksimum di referensi AssumeRoleAPI.

Cara kerjanya

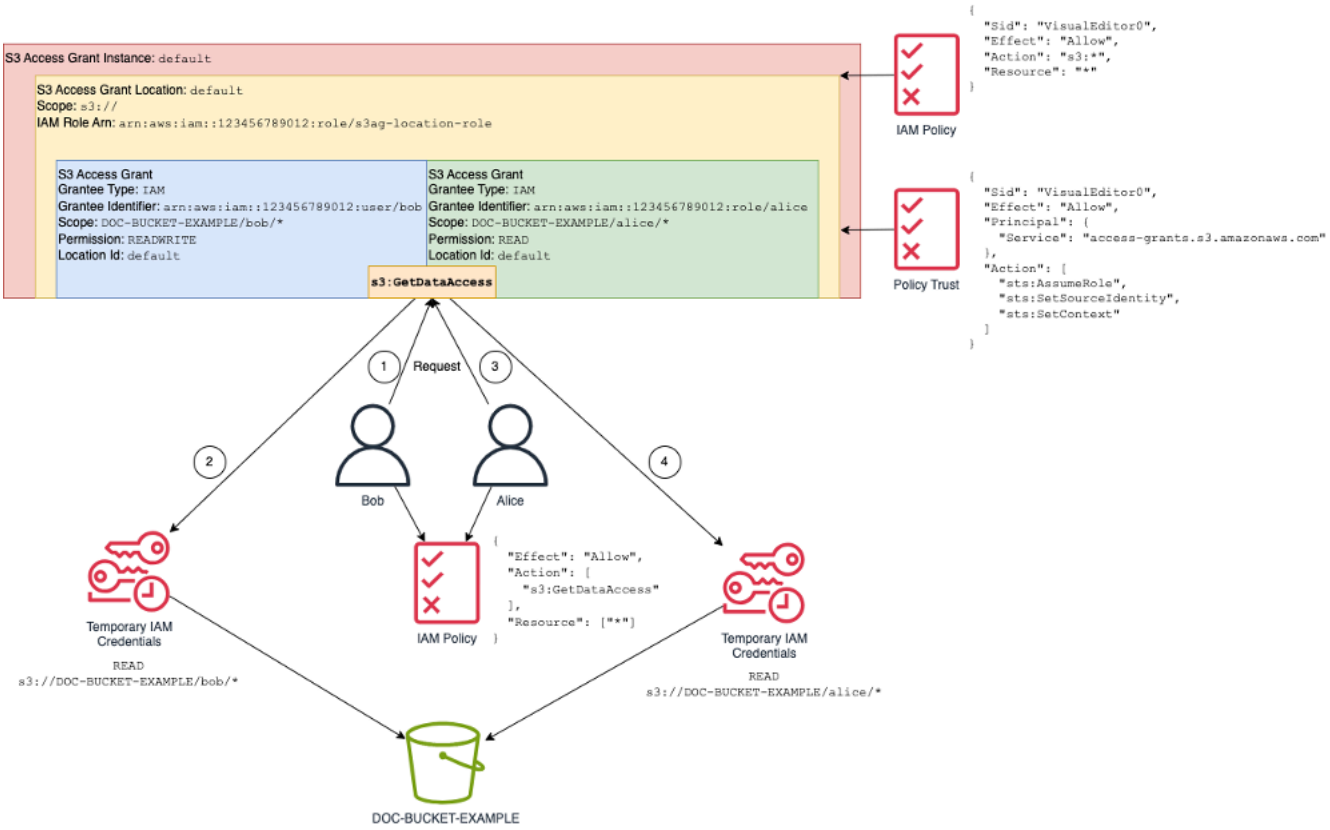
Dalam diagram berikut, lokasi Amazon S3 default dengan cakupan `s3://` terdaftar dengan peran IAM `s3ag-location-role`. Peran IAM ini memiliki izin untuk melakukan tindakan Amazon S3 dalam akun saat kredensialnya diperoleh melalui S3 Access Grants.

Di lokasi ini, dua pemberian akses individu dibuat untuk dua pengguna IAM. Pengguna IAM Bob diberikan keduanya `READ` dan `WRITE` akses pada `bob/` prefiks di bucket `DOC-BUCKET-EXAMPLE`. Peran IAM lainnya, Alice, hanya diberikan `READ` akses pada `alice/` awalan di ember. `DOC-BUCKET-EXAMPLE` Hibah, berwarna biru, didefinisikan untuk Bob untuk mengakses prefiks `bob/` di `DOC-BUCKET-EXAMPLE` bucket. Hibah, berwarna hijau, didefinisikan untuk Alice untuk mengakses prefiks `alice/` di `DOC-BUCKET-EXAMPLE` bucket.

Saat tiba waktunya bagi Bob untuk melakukan `READ` data, peran IAM yang terkait dengan lokasi hibahnya memanggil operasi S3 Access Grants API GetDataAccess. Jika Bob mencoba `READ` prefiks

atau S3 Object yang dimulai dengan`s3://DOC-BUCKET-EXAMPLE/bob/*`, `GetDataAccess` permintaan mengembalikan satu set kredensial sesi IAM sementara dengan izin untuk `s3://DOC-BUCKET-EXAMPLE/bob/*`. Demikian pula, Bob dapat `WRITE` ke prefiks atau S3 Object apa pun yang dimulai dengan `s3://DOC-BUCKET-EXAMPLE/bob/*`, karena pemberian juga memungkinkan itu.

Demikian pula, Alice bisa `READ` apa saja yang dimulai dengan `s3://DOC-BUCKET-EXAMPLE/alice/`. Namun, jika dia mencoba `WRITE` apa pun ke bucket, prefiks, atau objek apa pun `s3://`, dia akan mendapatkan kesalahan Access Denied (403 Forbidden), karena tidak ada pemberian yang memberikannya `WRITE` akses ke data apa pun. Selain itu, jika Alice meminta tingkat akses apa pun (`READ` atau `WRITE`) ke data di luar `s3://DOC-BUCKET-EXAMPLE/alice/`, dia akan kembali menerima kesalahan Akses Ditolak.



Pola ini menskalakan ke sejumlah besar pengguna dan bucket dan menyederhanakan pengelolaan izin tersebut. Daripada mengedit kebijakan bucket S3 yang berpotensi besar setiap kali Anda ingin menambahkan atau menghapus hubungan akses prefiks pengguna individual, Anda dapat menambahkan dan menghapus masing-masing pemberian terpisah.

S3 Access Grants dan identitas direktori perusahaan

Anda dapat menggunakan Amazon S3 Access Grants untuk memberikan akses ke prinsipal AWS Identity and Access Management (IAM) (pengguna atau peran), baik dalam hal yang sama maupun yang lain. Akun AWS Namun, dalam banyak kasus, entitas yang mengakses data adalah pengguna akhir dari direktori perusahaan Anda. Alih-alih memberikan akses ke pengguna utama IAM, Anda dapat menggunakan S3 Access Grants untuk memberikan akses langsung ke pengguna dan grup perusahaan Anda. Dengan S3 Access Grants, Anda tidak perlu lagi memetakan identitas perusahaan Anda ke pengguna utama IAM menengah untuk mengakses data S3 Anda melalui aplikasi perusahaan Anda.

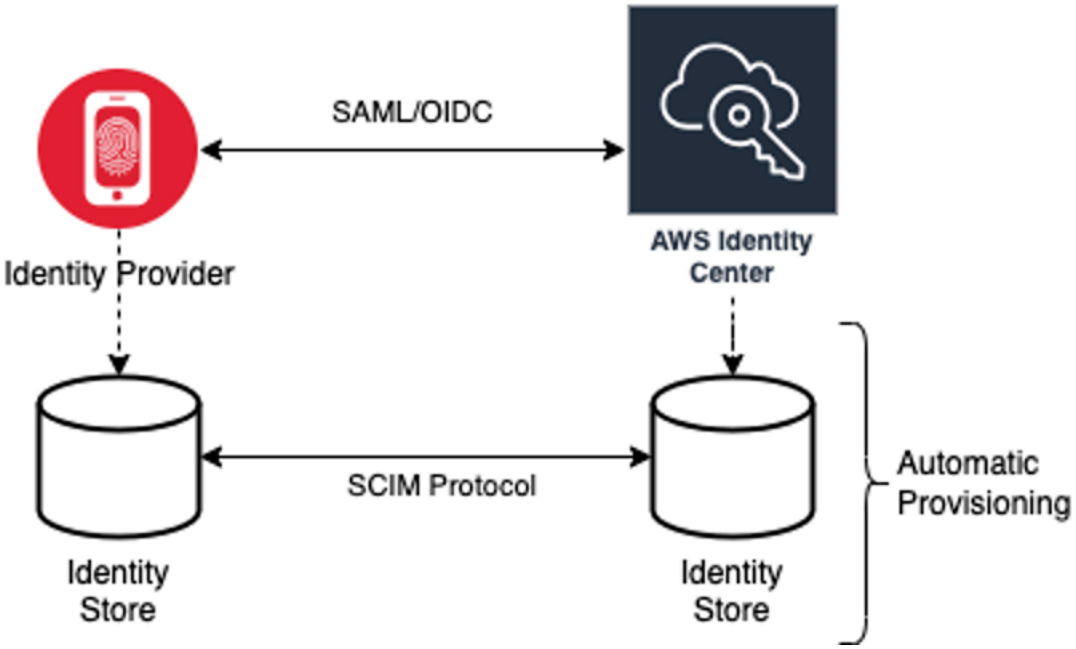
Fungsionalitas baru ini—dukungan untuk menggunakan akses identitas pengguna akhir ke data—disediakan dengan mengaitkan instance S3 Access Grants Anda dengan sebuah instance. AWS IAM Identity Center IAM Identity Center mendukung penyedia identitas berbasis standar dan merupakan hub AWS untuk layanan atau fitur apa pun, termasuk S3 Access Grants, yang mendukung identitas pengguna akhir. IAM Identity Center menyediakan dukungan otentikasi untuk identitas perusahaan melalui fitur propagasi identitas tepercaya. Untuk informasi selengkapnya, lihat Propagasi identitas terpercaya di seluruh aplikasi.

Untuk memulai dengan dukungan identitas tenaga kerja di S3 Access Grants, sebagai prasyarat, Anda mulai di IAM Identity Center dengan mengonfigurasi penyediaan identitas antara penyedia identitas perusahaan Anda dan IAM Identity Center. IAM Identity Center mendukung penyedia identitas perusahaan seperti Okta, Microsoft Entra ID (sebelumnya Azure Active Directory), atau penyedia identitas (IdP) eksternal lainnya yang mendukung protokol System for Cross-domain Identity Management (SCIM). Saat Anda menghubungkan Pusat Identitas IAM ke IdP dan mengaktifkan penyediaan otomatis, pengguna dan grup dari IdP Anda disinkronkan ke dalam penyimpanan identitas di Pusat Identitas IAM. Setelah langkah ini, IAM Identity Center memiliki pandangan sendiri tentang pengguna dan grup Anda, sehingga Anda dapat merujuk kepada mereka dengan menggunakan fitur lain Layanan AWS dan fitur, seperti S3 Access Grants. Untuk informasi selengkapnya tentang mengonfigurasi penyediaan otomatis Pusat Identitas IAM, lihat Penyediaan otomatis di Panduan Pengguna AWS IAM Identity Center .

IAM Identity Center terintegrasi AWS Organizations sehingga Anda dapat mengelola izin secara terpusat di beberapa Akun AWS tanpa mengonfigurasi setiap akun secara manual. Dalam organisasi tipikal, administrator identitas Anda mengonfigurasi satu instans Pusat Identitas IAM untuk seluruh organisasi, sebagai satu titik sinkronisasi identitas. Instance IAM Identity Center ini biasanya berjalan di dedicated Akun AWS di organisasi Anda. Dalam konfigurasi umum ini, Anda dapat merujuk ke identitas pengguna dan grup di S3 Access Grants dari manapun Akun AWS di organisasi.

Namun, jika AWS Organizations administrator belum mengonfigurasi instans Pusat Identitas IAM pusat, Anda dapat membuat instance lokal di akun yang sama dengan instans S3 Access Grants Anda. Konfigurasi seperti itu lebih umum untuk proof-of-concept atau kasus penggunaan pengembangan lokal. Dalam semua kasus, instance IAM Identity Center harus Wilayah AWS sama dengan instance S3 Access Grants yang akan dikaitkan.

Dalam diagram berikut konfigurasi Pusat Identitas IAM dengan IdP eksternal, IdP dikonfigurasi dengan SCIM untuk menyinkronkan penyimpanan identitas dari IdP ke penyimpanan identitas di IAM Identity Center.



Untuk menggunakan identitas direktori perusahaan Anda dengan S3 Access Grants, lakukan hal berikut ini:

- Siapkan penyediaan Otomatis di Pusat Identitas IAM untuk menyinkronkan informasi pengguna dan grup dari IdP Anda ke Pusat Identitas IAM.
- Konfigurasikan sumber identitas eksternal Anda dalam IAM Identity Center sebagai penerbit token tepercaya. Untuk informasi selengkapnya, lihat Propagasi identitas tepercaya di seluruh aplikasi di Panduan AWS IAM Identity Center Pengguna.
- Kaitkan instans S3 Access Grants Anda dengan instans Pusat Identitas IAM Anda. Anda dapat melakukan ini saat membuat instans S3 Access Grants Anda. Jika Anda telah membuat instance S3 Access Grants, lihat. Kaitkan atau pisahkan instans Pusat Identitas IAM Anda

Bagaimana identitas direktori dapat mengakses data S3

Misalkan Anda memiliki pengguna direktori perusahaan yang perlu mengakses data S3 Anda melalui aplikasi perusahaan, misalnya, aplikasi penampil dokumen, yang terintegrasi dengan IdP eksternal Anda (misalnya, Okta) untuk mengautentikasi pengguna. Otentikasi pengguna dalam aplikasi ini biasanya dilakukan melalui pengalihan di browser web pengguna. Karena pengguna dalam direktori bukan pengguna utama IAM, aplikasi Anda memerlukan kredensial IAM yang dapat memanggil operasi S3 Access Grants `GetDataAccess` API untuk mendapatkan kredensial akses ke data S3 atas nama pengguna. Tidak seperti pengguna IAM dan peran yang mendapatkan kredensialnya sendiri, aplikasi Anda memerlukan cara untuk mewakili pengguna direktori, yang tidak dipetakan ke peran IAM, sehingga pengguna bisa mendapatkan akses data melalui S3 Access Grants.

Transisi ini, dari pengguna direktori yang diautentikasi ke penelepon IAM yang dapat membuat permintaan ke S3 Access Grants atas nama pengguna direktori, dilakukan oleh aplikasi melalui fitur penerbit token tepercaya IAM Identity Center. Aplikasi, setelah mengautentikasi pengguna direktori, memiliki token identitas dari IDP (misalnya, Okta) yang mewakili pengguna direktori menurut Okta. Konfigurasi penerbit token tepercaya di IAM Identity Center memungkinkan aplikasi untuk menukar Okta token ini (Oktapenyewa dikonfigurasi sebagai “penerbit tepercaya”) untuk token identitas yang berbeda dari IAM Identity Center yang akan mewakili pengguna direktori dengan aman Layanan AWS. Aplikasi data kemudian akan mengambil peran IAM, menyediakan token pengguna direktori dari IAM Identity Center sebagai konteks tambahan. Aplikasi dapat menggunakan sesi IAM yang dihasilkan untuk memanggil S3 Access Grants. Token mewakili identitas aplikasi (pengguna utama IAM itu sendiri) serta identitas pengguna direktori.

Langkah utama dari transisi ini adalah pertukaran token. Aplikasi melakukan pertukaran token ini dengan memanggil operasi `CreateTokenWithIAM` API di IAM Identity Center. Tentu saja, itu juga merupakan panggilan AWS API dan membutuhkan kepala sekolah IAM untuk menandatanganinya. Pengguna utama IAM yang membuat permintaan ini biasanya merupakan peran IAM yang terkait dengan aplikasi. Misalnya, jika aplikasi berjalan di Amazon EC2, `CreateTokenWithIAM` permintaan biasanya dilakukan oleh peran IAM yang terkait dengan EC2 instance tempat aplikasi berjalan. Hasil dari `CreateTokenWithIAM` panggilan yang berhasil adalah token identitas baru, yang akan dikenali di dalamnya Layanan AWS.

Langkah selanjutnya, sebelum aplikasi dapat memanggil `GetDataAccess` nama pengguna direktori, adalah agar aplikasi mendapatkan sesi IAM yang mencakup identitas pengguna direktori. Aplikasi melakukan ini dengan `AssumeRole` permintaan AWS Security Token Service (AWS STS) yang juga menyertakan token IAM Identity Center untuk pengguna direktori sebagai konteks identitas tambahan. Konteks tambahan inilah yang memungkinkan IAM Identity Center untuk menyebarkan

identitas pengguna direktori ke langkah berikutnya. Peran IAM yang diasumsikan aplikasi adalah peran yang memerlukan izin IAM untuk memanggil operasi `GetDataAccess`.

Setelah mengasumsikan peran IAM pembawa identitas dengan token IAM Identity Center untuk pengguna direktori sebagai konteks tambahan, aplikasi sekarang memiliki semua yang diperlukan untuk membuat permintaan yang ditandatangani `GetDataAccess` atas nama pengguna direktori yang diautentikasi.

Penyebaran token didasarkan pada langkah-langkah berikut:

Buat aplikasi Pusat Identitas IAM

Pertama, buat aplikasi baru di IAM Identity Center. Aplikasi ini akan menggunakan template yang memungkinkan IAM Identity Center untuk mengidentifikasi jenis pengaturan aplikasi yang dapat Anda gunakan. Perintah untuk membuat aplikasi mengharuskan Anda untuk memberikan contoh IAM Identity Center Amazon Resource Name (ARN), nama aplikasi, dan penyedia aplikasi ARN. Penyedia aplikasi adalah SAMB atau penyedia OAuth aplikasi yang akan digunakan aplikasi untuk melakukan panggilan ke IAM Identity Center.

Untuk menggunakan perintah contoh berikut, ganti *`user input placeholders`* dengan informasi Anda sendiri:

```
aws sso-admin create-application \
 --instance-arn "arn:aws:sso:::instance/ssoins-ssoins-1234567890abcdef" \
 --application-provider-arn "arn:aws:sso::aws:applicationProvider/custom" \
 --name MyDataApplication
```

Respons:

```
{
   "ApplicationArn": "arn:aws:sso::123456789012:application/ssoins-
ssoins-1234567890abcdef/apl-abcd1234a1b2c3d"
}
```

Buat penerbit token tepercaya

Sekarang setelah Anda memiliki aplikasi IAM Identity Center, langkah selanjutnya adalah mengonfigurasi penerbit token tepercaya yang akan digunakan untuk menukar nilai `IdToken` Anda dari IdP Anda dengan token IAM Identity Center. Pada langkah ini Anda perlu memberikan item berikut:

- URL penerbit penyedia identitas
- Nama penerbit token tepercaya
- Jalur atribut klaim
- Jalur atribut toko identitas
- Opsi pengambilan JSON Web Key Set (JWKS)

Jalur atribut klaim adalah atribut penyedia identitas yang akan digunakan untuk memetakan ke atribut penyimpanan identitas. Biasanya, jalur atribut klaim adalah alamat email pengguna, tetapi Anda dapat menggunakan atribut lain untuk melakukan pemetaan.

Buat file yang disebut `oidc-configuration.json` dengan informasi berikut. Untuk menggunakan file ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
{
  "OidcJwtConfiguration":
     {
      "IssuerUrl": "https://login.microsoftonline.com/a1b2c3d4-abcd-1234-b7d5-
b154440ac123/v2.0",
      "ClaimAttributePath": "preferred_username",
      "IdentityStoreAttributePath": "userName",
      "JwksRetrievalOption": "OPEN_ID_DISCOVERY"
     }
}
```

Untuk membuat penerbit token tepercaya, jalankan perintah berikut. Untuk menggunakan contoh perintah ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws sso-admin create-trusted-token-issuer \
  --instance-arn "arn:aws:sso:::instance/ssoins-1234567890abcdef" \
  --name MyEntraIDTrustedIssuer \
  --trusted-token-issuer-type OIDC_JWT \
  --trusted-token-issuer-configuration file://./oidc-configuration.json
```

Respons

```
{
  "TrustedTokenIssuerArn": "arn:aws:sso::123456789012:trustedTokenIssuer/
ssoins-1234567890abcdef/tti-43b4a822-1234-1234-1234-a1b2c3d41234"
}
```

Connect aplikasi IAM Identity Center dengan penerbit token tepercaya

Penerbit token tepercaya memerlukan beberapa pengaturan konfigurasi lagi untuk bekerja. Tetapkan audiens yang akan dipercaya oleh penerbit token tepercaya. Audiens adalah nilai di dalam `IdToken` yang diidentifikasi oleh kunci dan dapat ditemukan di pengaturan penyedia identitas. Misalnya:

```
1234973b-abcd-1234-abcd-345c5a9c1234
```

Buat file dengan nama `grant.json` dengan konten berikut ini. Untuk menggunakan file ini, ubah audiens agar sesuai dengan pengaturan penyedia identitas Anda dan berikan ARN penerbit token tepercaya yang dikembalikan oleh perintah sebelumnya.

```
{
   "JwtBearer":
     {
       "AuthorizedTokenIssuers":
         [
           {
             "TrustedTokenIssuerArn": "arn:aws:sso::123456789012:trustedTokenIssuer/
ssoins-1234567890abcdef/tti-43b4a822-1234-1234-1234-a1b2c3d41234",
               "AuthorizedAudiences":
                 [
                   "1234973b-abcd-1234-abcd-345c5a9c1234"
                 ]
            }
         ]
     }
 }
```

Jalankan perintah contoh berikut. Untuk menggunakan perintah ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws sso-admin put-application-grant \
  --application-arn "arn:aws:sso::123456789012:application/ssoins-
ssoins-1234567890abcdef/apl-abcd1234a1b2c3d" \
  --grant-type "urn:ietf:params:oauth:grant-type:jwt-bearer" \
  --grant file://./grant.json \
```

Perintah ini menetapkan penerbit token tepercaya dengan pengaturan konfigurasi untuk mempercayai audiens dalam file `grant.json` dan menautkan audiens ini dengan aplikasi yang dibuat pada langkah pertama untuk bertukar token dari jenis `jwt-bearer`. String

`urn:ietf:params:oauth:grant-type:jwt-bearer` bukan merupakan string arbitrer. Ini adalah namespace terdaftar di profil pernyataan OAuth JSON Web Token (JWT). Anda dapat menemukan informasi lebih lanjut tentang namespace ini di RFC 7523.

Selanjutnya, gunakan perintah berikut untuk mengatur cakupan mana yang akan disertakan oleh penerbit token tepercaya saat bertukar `IdToken` nilai dari penyedia identitas Anda. Untuk S3 Access Grants, nilai untuk `--scope` parameter adalah `s3:access_grants:read_write`.

```
aws sso-admin put-application-access-scope \
  --application-arn "arn:aws:sso::111122223333:application/ssoins-
ssoins-111122223333abcdef/apl-abcd1234a1b2c3d" \
  --scope "s3:access_grants:read_write"
```

Langkah terakhir adalah melampirkan kebijakan sumber daya ke aplikasi IAM Identity Center. Kebijakan ini akan memungkinkan peran IAM aplikasi Anda untuk membuat permintaan ke operasi API `sso-oauth:CreateTokenWithIAM` dan menerima `IdToken` nilai dari IAM Identity Center.

Buat file dengan nama `authentication-method.json` dengan konten berikut ini. Ganti *123456789012* dengan ID akun Anda.

```
{
   "Iam":
       {
         "ActorPolicy":
             {
                "Version": "2012-10-17",
                    "Statement":
                    [
                        {
                           "Effect": "Allow",
                            "Principal":
                            {
                              "AWS": "arn:aws:iam::123456789012:role/webapp"
                            },
                           "Action": "sso-oauth:CreateTokenWithIAM",
                            "Resource": "*"
                        }
                    ]
                }
            }
        }
```

Untuk melampirkan kebijakan ke aplikasi IAM Identity Center, jalankan perintah berikut:

```
aws sso-admin put-application-authentication-method \
   --application-arn "arn:aws:sso::123456789012:application/ssoins-
ssoins-1234567890abcdef/apl-abcd1234a1b2c3d" \
   --authentication-method-type IAM \
   --authentication-method file://./authentication-method.json
```

Ini melengkapi pengaturan konfigurasi untuk menggunakan S3 Access Grants dengan pengguna direktori melalui aplikasi web. Anda dapat menguji penyiapan ini secara langsung di aplikasi atau Anda dapat memanggil operasi `CreateTokenWithIAM` API dengan menggunakan perintah berikut dari peran IAM yang diizinkan dalam kebijakan aplikasi Pusat Identitas IAM:

```
aws sso-oidc create-token-with-iam \
   --client-id "arn:aws:sso::123456789012:application/ssoins-ssoins-1234567890abcdef/
apl-abcd1234a1b2c3d"  \
   --grant-type urn:ietf:params:oauth:grant-type:jwt-bearer \
   --assertion IdToken
```

Respons akan serupa dengan ini:

```
{
    "accessToken": "<suppressed long string to reduce space>",
    "tokenType": "Bearer",
    "expiresIn": 3600,
    "refreshToken": "<suppressed long string to reduce space>",
    "idToken": "<suppressed long string to reduce space>",
    "issuedTokenType": "urn:ietf:params:oauth:token-type:refresh_token",
    "scope": [
      "sts:identity_context",
      "s3:access_grants:read_write",
      "openid",
      "aws"
    ]
}
```

Jika Anda memecahkan kode `IdToken` nilai yang dikodekan dengan base64, Anda dapat melihat pasangan kunci-nilai dalam format JSON. Kunci `sts:identity_context` berisi nilai yang perlu dikirim aplikasi Anda dalam `sts:AssumeRole` permintaan untuk menyertakan informasi identitas pengguna direktori. Berikut adalah contoh yang `IdToken` di-decode:

```
{
    "aws:identity_store_id": "d-996773e796",
    "sts:identity_context": "AQoJb3JpZ2luX2VjEOTtl;<SUPRESSED>",
    "sub": "83d43802-00b1-7054-db02-f1d683aacba5",
    "aws:instance_account": "123456789012",
    "iss": "https://identitycenter.amazonaws.com/ssoins-1234567890abcdef",
    "sts:audit_context": "AQoJb3JpZ2luX2VjEOT<SUPRESSED>==",
    "aws:identity_store_arn": "arn:aws:identitystore::232642235904:identitystore/
d-996773e796",
    "aud": "abcd12344U0gi7n4Yyp0-WV1LWNlbnRyYWwtMQ",
    "aws:instance_arn": "arn:aws:sso:::instance/ssoins-6987d7fb04cf7a51",
    "aws:credential_id": "EXAMPLEHI5glPh40y9TpApJn8...",
    "act": {
       "sub": "arn:aws:sso::232642235904:trustedTokenIssuer/
ssoins-6987d7fb04cf7a51/43b4a822-1020-7053-3631-cb2d3e28d10e"
    },
    "auth_time": "2023-11-01T20:24:28Z",
    "exp": 1698873868,
    "iat": 1698870268
}
```

Anda bisa mendapatkan nilai dari `sts:identity_context` dan meneruskan informasi ini dalam `sts:AssumeRole` panggilan. Berikut ini adalah contoh CLI dari sintaks. Peran yang akan diasumsikan adalah peran sementara dengan izin untuk menginvokasi `s3:GetDataAccess`.

```
aws sts assume-role \
   --role-arn "arn:aws:iam::123456789012:role/temp-role" \
   --role-session-name "TempDirectoryUserRole" \
   --provided-contexts ProviderArn="arn:aws:iam::aws:contextProvider/
IdentityCenter",ContextAssertion="value from sts:identity_context"
```

Anda sekarang dapat menggunakan kredensial yang diterima dari panggilan ini untuk menginvokasi operasi `s3:GetDataAccess` API dan menerima kredensial akhir dengan akses ke sumber daya S3 Anda.

Memulai S3 Access Grants

Amazon S3 Access Grants adalah fitur Amazon S3 yang menyediakan solusi kontrol akses yang dapat diskalakan untuk data S3 Anda. S3 Access Grants adalah vendor kredensial S3, artinya Anda mendaftar dengan S3 Access Grants daftar pemberian Anda dan pada tingkat apa. Setelah itu, ketika pengguna atau klien membutuhkan akses ke data S3 Anda, mereka terlebih dahulu meminta S3

Access Grants untuk kredensial. Jika ada pemberian terkait yang mengotorisasi akses, S3 Access Grants menjual kredensial akses sementara dengan hak akses paling rendah. Pengguna atau klien kemudian dapat menggunakan kredensial penjual S3 Access Grants untuk mengakses data S3 Anda. Dengan mengingat hal itu, jika persyaratan data S3 Anda mengamanatkan konfigurasi izin yang kompleks atau besar, Anda dapat menggunakan S3 Access Grants untuk menskalakan izin data S3 untuk pengguna, grup, peran, dan aplikasi.

Untuk sebagian besar kasus penggunaan, Anda dapat mengelola kontrol akses untuk data S3 menggunakan AWS Identity and Access Management (IAM) dengan kebijakan bucket atau kebijakan berbasis identitas IAM.

Namun, jika Anda memiliki persyaratan kontrol akses S3 yang kompleks, seperti berikut ini, Anda bisa mendapatkan keuntungan besar dari menggunakan S3 Access Grants:

- Anda mengalami batas ukuran kebijakan bucket sebesar 20 KB.
- Anda memberikan identitas manusia, misalnya, Microsoft Entra ID (sebelumnya adalah Azure Active Directory), Okta atau pengguna dan grup Ping, akses ke data S3 untuk analitik dan big data.
- Anda harus menyediakan akses lintas akun tanpa sering memperbarui kebijakan IAM.
- Data Anda tidak terstruktur dan tingkat objek daripada terstruktur, dalam format baris dan kolom.

Alur kerja S3 Access Grants adalah sebagai berikut:

Langkah-langkah	Deskripsi
1	Membuat instans S3 Access Grants Untuk memulai, memulai instans S3 Access Grants yang akan berisi pemberian akses individual Anda.
2	Mendaftarkan lokasi Kedua, daftarkan lokasi data S3 (seperti default, `s3://`) dan kemudian tentukan peran IAM default yang diasumsikan oleh S3 Access Grants saat menyediakan akses ke lokasi data S3. Anda juga dapat menambahkan lokasi kustom ke bucket atau prefiks tertentu dan memetakannya ke peran IAM kustom.
3	Membuat pemberian

Langkah-langkah	Deskripsi
	Buat pemberian izin individu. Tentukan dalam izin ini memberikan lokasi S3 terdaftar, ruang lingkup akses data di dalam lokasi, identitas penerima pemberian, dan tingkat aksesnya (`READ`, `WRITE`, atau `READWRITE`).
4	Minta akses ke data S3 Ketika pengguna, aplikasi, dan Layanan AWS ingin mengakses data S3, mereka pertama kali membuat permintaan akses. S3 Access Grants menentukan apakah permintaan harus diotorisasi. Jika ada pemberian terkait yang mengotorisasi akses, S3 Access Grants menggunakan peran IAM lokasi terdaftar yang terkait dengan pemberian tersebut untuk menjual kembali kredensial sementara ke pemohon.
5	Mengakses data S3 Aplikasi menggunakan kredensial sementara yang dijual oleh S3 Access Grants untuk mengakses data S3.

Bekerja dengan instans S3 Access Grants

Untuk memulai menggunakan AmazonS3 Access Grants, pertama-tama Anda membuat instans S3 Access Grants. Anda hanya dapat membuat satu instans Hibah Akses S3 Wilayah AWS per akun. Instans S3 Access Grants berfungsi sebagai kontainer untuk sumber daya S3 Access Grants Anda, yang mencakup lokasi dan pemberian terdaftar.

Dengan S3 Access Grants, Anda dapat membuat hibah izin ke data S3 untuk pengguna dan peran AWS Identity and Access Management (IAM). Jika Anda telah menambahkan direktori identitas perusahaan AWS IAM Identity Center, Anda dapat mengaitkan instance IAM Identity Center ini dari direktori perusahaan Anda dengan instans S3 Access Grants Anda. Setelah selesai, Anda dapat membuat pemberian akses untuk pengguna dan grup perusahaan Anda. Jika Anda belum menambahkan direktori perusahaan ke Pusat Identitas IAM, Anda dapat mengaitkan instans S3 Access Grants Anda dengan instans Pusat Identitas IAM nantinya.

Topik

- Membuat instans S3 Access Grants
- Dapatkan detail instans S3 Access Grants
- Buat daftar instans S3 Access Grants Anda
- Kaitkan atau pisahkan instans Pusat Identitas IAM Anda
- Hapus instans S3 Access Grants

Membuat instans S3 Access Grants

Untuk memulai menggunakan AmazonS3 Access Grants, pertama-tama Anda membuat instans S3 Access Grants. Anda hanya dapat membuat satu instans Hibah Akses S3 Wilayah AWS per akun. Instans S3 Access Grants berfungsi sebagai kontainer untuk sumber daya S3 Access Grants Anda, yang mencakup lokasi dan pemberian terdaftar.

Dengan S3 Access Grants, Anda dapat membuat hibah izin ke data S3 untuk pengguna dan peran AWS Identity and Access Management (IAM). Jika Anda telah menambahkan direktori identitas perusahaan AWS IAM Identity Center, Anda dapat mengaitkan instance IAM Identity Center ini dari direktori perusahaan Anda dengan instans S3 Access Grants Anda. Setelah selesai, Anda dapat membuat pemberian akses untuk pengguna dan grup perusahaan Anda. Jika Anda belum menambahkan direktori perusahaan ke Pusat Identitas IAM, Anda dapat mengaitkan instans S3 Access Grants Anda dengan instans Pusat Identitas IAM nantinya.

Anda dapat membuat instance S3 Access Grants dengan menggunakan konsol Amazon S3,AWS CLI(), AWS Command Line Interface Amazon S3 REST API, dan. AWS SDKs

Menggunakan konsol S3

Sebelum Anda dapat memberikan akses ke data S3 Anda dengan S3 Access Grants, Anda harus terlebih dahulu membuat instance S3 Access Grants sama Wilayah AWS dengan data S3 Anda.

Prasyarat

Jika Anda ingin memberikan akses ke data S3 Anda dengan menggunakan identitas dari direktori perusahaan Anda, tambahkan direktori identitas perusahaan Anda ke. AWS IAM Identity Center Jika Anda belum siap untuk melakukannya, Anda dapat mengaitkan instans S3 Access Grants Anda dengan instans IAM Identity Center nanti.

Membuat instans S3 Access Grants

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah yang ingin Anda alihkan.
3. Pada panel navigasi di sebelah kiri, pilih Access Grants.
4. Pada halaman S3 Access Grants, pilih Buat S3 Access Grants instans.

   a. Pada Langkah 1 dari wizard Instans Set up Access Grants, verifikasi bahwa Anda ingin membuat instans saat ini Wilayah AWS. Pastikan bahwa ini sama di Wilayah AWS mana data S3 Anda berada. Anda dapat membuat satu instance Hibah Akses S3 Wilayah AWS per akun.

   b. (Opsional) Jika Anda telah menambahkan direktori identitas perusahaan AWS IAM Identity Center, Anda dapat mengaitkan instance IAM Identity Center ini dari direktori perusahaan Anda dengan instans S3 Access Grants Anda.

      Untuk melakukannya, pilih Add IAM Identity Center instance di ***region***. Kemudian masukkan instans Amazon Resource Name (ARN).

      Jika Anda belum menambahkan direktori perusahaan ke Pusat Identitas IAM, Anda dapat mengaitkan instans S3 Access Grants Anda dengan instans Pusat Identitas IAM nantinya.

   c. Untuk membuat instans S3 Access Grants, pilih Berikutnya. Untuk mendaftarkan lokasi, lihat Langkah 2-mendaftarkan lokasi.

5. Jika Selanjutnya atau Buat instans S3 Access Grants dinonaktifkan:

   Tidak dapat membuat instans

   - Anda mungkin sudah memiliki instans S3 Access Grants Wilayah AWS yang sama. Pada panel navigasi di sebelah kiri, pilih Access Grants. Pada halaman S3 Access Grants, gulir ke bawah ke instans S3 Access Grants di bagian akun Anda untuk menentukan apakah instans sudah ada.
   - Anda mungkin tidak memiliki `s3:CreateAccessGrantsInstance` izin yang diperlukan untuk membuat instance S3 Access Grants. Menghubungi administrator akun Anda. Untuk izin tambahan yang diperlukan jika Anda mengaitkan instans Pusat Identitas IAM, dengan instans S3 Access Grants Anda, lihat. CreateAccessGrantsInstance

Menggunakan AWS CLI

Untuk menginstal AWS CLI, lihat Menginstal AWS CLI di Panduan AWS Command Line Interface Pengguna.

Untuk menggunakan perintah contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri.

Example Membuat instans S3 Access Grants

```
aws s3control create-access-grants-instance \
--account-id 111122223333 \
--region us-east-2
```

Respons:

```
{
    "CreatedAt": "2023-05-31T17:54:07.893000+00:00",
    "AccessGrantsInstanceId": "default",
    "AccessGrantsInstanceArn": "arn:aws:s3:us-east-2:111122223333:access-grants/
default"
}
```

Penggunaan API REST

Anda dapat menggunakan API REST Amazon S3 untuk membuat instans S3 Access Grants. Untuk informasi tentang dukungan API REST untuk mengelola instans S3 Access Grants, lihat bagian berikut di Referensi API Amazon Simple Storage Service:

- AssociateAccessGrantsIdentityCenter
- CreateAccessGrantsInstance
- DeleteAccessGrantsInstance
- DissociateAccessGrantsIdentityCenter
- GetAccessGrantsInstance
- GetAccessGrantsInstanceForPrefix
- GetAccessGrantsInstanceResourcePolicy
- ListAccessGrantsInstances
- PutAccessGrantsInstanceResourcePolicy

Menggunakan AWS SDKs

Bagian ini memberikan contoh cara membuat instance S3 Access Grants dengan menggunakan. AWS SDKs

Java

Contoh ini membuat instans S3 Access Grants, yang berfungsi sebagai kontainer untuk pemberian akses individual Anda. Anda dapat memiliki satu instance S3 Access Grants per akun Wilayah AWS Anda. Respons tersebut mencakup ID instans `default` dan Amazon Resource Name (ARN) yang dihasilkan untuk instans S3 Access Grants Anda.

Example Membuat permintaan instans S3 Access Grants

```
public void createAccessGrantsInstance() {
CreateAccessGrantsInstanceRequest createRequest =
 CreateAccessGrantsInstanceRequest.builder().accountId("111122223333").build();
CreateAccessGrantsInstanceResponse createResponse =
 s3Control.createAccessGrantsInstance(createRequest);LOGGER.info("CreateAccessGrantsInstance
 " + createResponse);
}
```

Respons:

```
CreateAccessGrantsInstanceResponse(
CreatedAt=2023-06-07T01:46:20.507Z,
AccessGrantsInstanceId=default,
AccessGrantsInstanceArn=arn:aws:s3:us-east-2:111122223333:access-grants/default)
```

Dapatkan detail instans S3 Access Grants

Anda bisa mendapatkan detail instans Hibah Akses Amazon S3 Anda secara khusus. Wilayah AWS Anda bisa mendapatkan detail instans S3 Access Grants dengan menggunakan konsol Amazon S3,AWS CLI(), AWS Command Line Interface Amazon S3 REST API, dan file. AWS SDKs

Menggunakan konsol S3

Untuk mendapatkan detail instance S3 Access Grants

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

2. Pada panel navigasi di sebelah kiri, pilih Access Grants.
3. Pada halaman S3 Access Grants, pilih Region yang berisi instance S3 Access Grants yang ingin Anda gunakan.
4. Halaman S3 Access Grants mencantumkan instans S3 Access Grants Anda dan instans lintas akun yang telah dibagikan dengan akun Anda. Untuk melihat detail instance, pilih Lihat detail.

Menggunakan AWS CLI

Untuk menginstal AWS CLI, lihat Menginstal AWS CLI di Panduan AWS Command Line Interface Pengguna.

Untuk menggunakan perintah contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri.

Example — Dapatkan detail instance Hibah Akses S3

```
aws s3control get-access-grants-instance \
 --account-id 111122223333 \
 --region us-east-2
```

Respons:

```
{
    "AccessGrantsInstanceArn": "arn:aws:s3:us-east-2: 111122223333:access-grants/
default",
    "AccessGrantsInstanceId": "default",
    "CreatedAt": "2023-05-31T17:54:07.893000+00:00"
}
```

Penggunaan API REST

Untuk informasi tentang dukungan Amazon S3 REST API untuk mengelola instance S3 Access Grants, lihat bagian berikut di Referensi API Layanan Penyimpanan Sederhana Amazon:

- GetAccessGrantsInstance
- GetAccessGrantsInstanceForPrefix

Menggunakan AWS SDKs

Bagian ini memberikan contoh cara mendapatkan detail instance S3 Access Grants dengan menggunakan. AWS SDKs

Untuk menggunakan contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri.

Java

Example — Dapatkan instance Hibah Akses S3

```
public void getAccessGrantsInstance() {
GetAccessGrantsInstanceRequest getRequest = GetAccessGrantsInstanceRequest.builder()
.accountId("111122223333")
.build();
GetAccessGrantsInstanceResponse getResponse =
 s3Control.getAccessGrantsInstance(getRequest);
LOGGER.info("GetAccessGrantsInstanceResponse: " + getResponse);
}
```

Respons:

```
GetAccessGrantsInstanceResponse(
AccessGrantsInstanceArn=arn:aws:s3:us-east-2: 111122223333:access-grants/default,
CreatedAt=2023-06-07T01:46:20.507Z)
```

Buat daftar instans S3 Access Grants Anda

Anda dapat mencantumkan instans S3 Access Grants, termasuk instance yang telah dibagikan dengan Anda melalui (). AWS Resource Access Manager AWS RAM

Anda dapat mencantumkan instans S3 Access Grants dengan menggunakan konsol Amazon S3, AWS Command Line Interface ()AWS CLI, Amazon S3 REST API, dan. AWS SDKs

Menggunakan konsol S3

Untuk mencantumkan instans Hibah Akses S3 Anda

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

2. Pada panel navigasi di sebelah kiri, pilih Access Grants.
3. Pada halaman S3 Access Grants, pilih Region yang berisi instance S3 Access Grants yang ingin Anda gunakan.
4. Halaman S3 Access Grants mencantumkan instans S3 Access Grants Anda dan instans lintas akun yang telah dibagikan dengan akun Anda. Untuk melihat detail instance, pilih Lihat detail.

Menggunakan AWS CLI

Untuk menginstal AWS CLI, lihat Menginstal AWS CLI di Panduan AWS Command Line Interface Pengguna.

Untuk menggunakan perintah contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri.

Example — Daftar semua instance Hibah Akses S3 untuk akun

Tindakan ini mencantumkan instans S3 Access Grants untuk sebuah akun. Anda hanya dapat memiliki satu instance S3 Access Grants per. Wilayah AWS Tindakan ini juga mencantumkan instance Hibah Akses S3 lintas akun lain yang dapat diakses akun Anda.

```
aws s3control list-access-grants-instances \
 --account-id 111122223333 \
 --region us-east-2
```

Respons:

```
{
    "AccessGrantsInstanceArn": "arn:aws:s3:us-east-2: 111122223333:access-grants/
default",
    "AccessGrantsInstanceId": "default",
    "CreatedAt": "2023-05-31T17:54:07.893000+00:00"
}
```

Penggunaan API REST

Untuk informasi tentang dukungan Amazon S3 REST API untuk mengelola instance S3 Access Grants, lihat bagian berikut di Referensi API Layanan Penyimpanan Sederhana Amazon:

- ListAccessGrantsInstances

Menggunakan AWS SDKs

Bagian ini memberikan contoh cara mendapatkan detail instance S3 Access Grants dengan menggunakan. AWS SDKs

Untuk menggunakan contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri.

Java

Example — Daftar semua instance Hibah Akses S3 untuk akun

Tindakan ini mencantumkan instans S3 Access Grants untuk sebuah akun. Anda hanya dapat memiliki satu instans S3 Access Grants per Wilayah. Tindakan ini juga dapat mencantumkan instans S3 Access Grants lintas akun lain yang dapat diakses oleh akun Anda.

```
public void listAccessGrantsInstances() {
ListAccessGrantsInstancesRequest listRequest =
 ListAccessGrantsInstancesRequest.builder()
.accountId("111122223333")
.build();
ListAccessGrantsInstancesResponse listResponse =
 s3Control.listAccessGrantsInstances(listRequest);
LOGGER.info("ListAccessGrantsInstancesResponse: " + listResponse);
}
```

Respons:

```
ListAccessGrantsInstancesResponse(
AccessGrantsInstancesList=[
ListAccessGrantsInstanceEntry(
AccessGrantsInstanceId=default,
AccessGrantsInstanceArn=arn:aws:s3:us-east-2:111122223333:access-grants/default,
CreatedAt=2023-06-07T04:28:11.728Z
)
]
)
```

Kaitkan atau pisahkan instans Pusat Identitas IAM Anda

Di Amazon S3 Access Grants, Anda dapat mengaitkan AWS IAM Identity Center instance direktori identitas perusahaan Anda dengan instance S3 Access Grants. Setelah melakukannya, Anda dapat membuat hibah akses untuk pengguna dan grup direktori perusahaan Anda, selain pengguna dan AWS Identity and Access Management peran (IAM).

Jika Anda tidak lagi ingin membuat hibah akses untuk pengguna dan grup direktori perusahaan, Anda dapat memisahkan instans Pusat Identitas IAM Anda dari instans S3 Access Grants Anda.

Anda dapat mengaitkan atau memisahkan instans Pusat Identitas IAM dengan menggunakan konsol Amazon S3,AWS CLI(), AWS Command Line Interface Amazon S3 REST API, dan. AWS SDKs

Menggunakan konsol S3

Sebelum Anda mengaitkan instans Pusat Identitas IAM Anda dengan instans S3 Access Grants Anda, Anda harus menambahkan direktori identitas perusahaan Anda ke IAM Identity Center. Untuk informasi selengkapnya, lihat the section called “S3 Access Grants dan identitas direktori perusahaan”.

Untuk mengaitkan instans Pusat Identitas IAM dengan instans S3 Access Grants

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Pada panel navigasi di sebelah kiri, pilih Access Grants.
3. Pada halaman S3 Access Grants, pilih Region yang berisi instance S3 Access Grants yang ingin Anda gunakan.
4. Pilih Lihat detail untuk contoh.
5. Pada halaman detail, di bagian Pusat Identitas IAM, pilih untuk Menambahkan instance Pusat Identitas IAM atau Deregister instance Pusat Identitas IAM yang sudah terkait.

Menggunakan AWS CLI

Untuk menginstal AWS CLI, lihat Menginstal AWS CLI di Panduan AWS Command Line Interface Pengguna.

Untuk menggunakan perintah contoh berikut, ganti *`user input placeholders`* dengan informasi Anda sendiri.

Example — Kaitkan instans Pusat Identitas IAM dengan instance S3 Access Grants

```
aws s3control associate-access-grants-identity-center \
 --account-id 111122223333 \
 --identity-center-arn arn:aws:sso:::instance/ssoins-1234a567bb89012c \
 --profile access-grants-profile \
 --region eu-central-1

// No response body
```

Example — Lepaskan instans Pusat Identitas IAM dari instance S3 Access Grants

```
aws s3control dissociate-access-grants-identity-center \
 --account-id 111122223333 \
 --profile access-grants-profile \
 --region eu-central-1

// No response body
```

Penggunaan API REST

Untuk informasi tentang dukungan Amazon S3 REST API untuk mengelola asosiasi antara instans Pusat Identitas IAM dan instance S3 Access Grants, lihat bagian berikut di Referensi API Layanan Penyimpanan Sederhana Amazon:

- AssociateAccessGrantsIdentityCenter
- DissociateAccessGrantsIdentityCenter

Hapus instans S3 Access Grants

Anda dapat menghapus instans Amazon S3 Access Grants dari akun Wilayah AWS Anda. Namun, sebelum Anda dapat menghapus instance S3 Access Grants, Anda harus terlebih dahulu melakukan hal berikut:

- Hapus semua sumber daya dalam instance S3 Access Grants, termasuk semua hibah dan lokasi. Untuk informasi selengkapnya, lihat Menghapus hibah dan Menghapus lokasi.
- Jika Anda telah mengaitkan AWS IAM Identity Center instance dengan instans S3 Access Grants, Anda harus memisahkan instance IAM Identity Center. Untuk informasi selengkapnya, lihat Mengaitkan atau memisahkan instans Pusat Identitas IAM Anda.

> **Important**
>
> Jika Anda menghapus instance S3 Access Grants, penghapusan bersifat permanen dan tidak dapat dibatalkan. Semua penerima hibah yang diberi akses melalui hibah dalam instance Hibah Akses S3 ini akan kehilangan akses ke data S3 Anda.

Anda dapat menghapus instans S3 Access Grants dengan menggunakan konsol Amazon S3,AWS CLI(), AWS Command Line Interface Amazon S3 REST API, dan. AWS SDKs

Menggunakan konsol S3

Untuk menghapus instans S3 Access Grants

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Pada panel navigasi di sebelah kiri, pilih Access Grants.
3. Pada halaman S3 Access Grants, pilih Region yang berisi instance S3 Access Grants yang ingin Anda gunakan.
4. Pilih Lihat detail untuk contoh.
5. Pada halaman detail instance, pilih Hapus instance di sudut kanan atas.
6. Di kotak dialog yang muncul, pilih Hapus. Tindakan ini tidak dapat dibatalkan.

Menggunakan AWS CLI

Untuk menginstal AWS CLI, lihat Menginstal AWS CLI di Panduan AWS Command Line Interface Pengguna.

Untuk menggunakan perintah contoh berikut, ganti *`user input placeholders`* dengan informasi Anda sendiri.

> **Note**
>
> Sebelum Anda dapat menghapus instance S3 Access Grants, Anda harus terlebih dahulu menghapus semua hibah dan lokasi yang dibuat dalam instance S3 Access Grants. Jika Anda telah mengaitkan instans Pusat Identitas IAM dengan instans S3 Access Grants, Anda harus memutuskannya terlebih dahulu.

Example — Hapus instance Hibah Akses S3

```
aws s3control delete-access-grants-instance \
--account-id 111122223333 \
--profile access-grants-profile \
--region us-east-2 \
--endpoint-url https://s3-control.us-east-2.amazonaws.com \

 // No response body
```

Penggunaan API REST

Untuk informasi tentang dukungan Amazon S3 REST API untuk menghapus instance S3 Access Grants, lihat di DeleteAccessGrantsInstanceReferensi API Amazon Simple Storage Service.

Menggunakan AWS SDKs

Bagian ini memberikan contoh cara menghapus instance S3 Access Grants dengan menggunakan. AWS SDKs

Untuk menggunakan contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri.

Java

> **Note**
>
> Sebelum Anda dapat menghapus instance S3 Access Grants, Anda harus terlebih dahulu menghapus semua hibah dan lokasi yang dibuat dalam instance S3 Access Grants. Jika Anda telah mengaitkan instans Pusat Identitas IAM dengan instans S3 Access Grants, Anda harus memutuskannya terlebih dahulu.

Example — Hapus instance Hibah Akses S3

```
public void deleteAccessGrantsInstance() {
DeleteAccessGrantsInstanceRequest deleteRequest =
 DeleteAccessGrantsInstanceRequest.builder()
.accountId("111122223333")
.build();
```

```
DeleteAccessGrantsInstanceResponse deleteResponse =
 s3Control.deleteAccessGrantsInstance(deleteRequest);
LOGGER.info("DeleteAccessGrantsInstanceResponse: " + deleteResponse);
}
```

Bekerja dengan lokasi S3 Access Grants

Setelah membuat instans Amazon S3 Access Grants Wilayah AWS di akun Anda, Anda mendaftarkan lokasi S3 dalam instance tersebut. Lokasi S3 Access Grants memetakan lokasi default S3 (`s3://`), bucket, atau awalan ke peran AWS Identity and Access Management (IAM). S3 Access Grants mengasumsikan peran IAM ini untuk menjual kredensil sementara kepada penerima hibah yang mengakses lokasi tertentu. Anda harus terlebih dahulu mendaftarkan setidaknya satu lokasi di instans S3 Access Grants sebelum Anda dapat membuat hibah akses.

Anda dapat mendaftarkan lokasi, melihat detail lokasi, mengedit lokasi, dan menghapus lokasi.

> **Note**
>
> Setelah Anda mendaftarkan lokasi pertama di instans S3 Access Grants, instans Anda masih tidak memiliki hibah akses individual di dalamnya. Untuk membuat hibah akses, lihatMembuat pemberian.

Topik

- Mendaftarkan lokasi
- Lihat detail lokasi yang terdaftar
- Memperbarui lokasi terdaftar
- Hapus lokasi terdaftar

Mendaftarkan lokasi

Setelah membuat instans Amazon S3 Access Grants Wilayah AWS di akun Anda, Anda mendaftarkan lokasi S3 dalam instance tersebut. Lokasi S3 Access Grants memetakan lokasi default S3 (`s3://`), bucket, atau awalan ke peran AWS Identity and Access Management (IAM). S3 Access Grants mengasumsikan peran IAM ini untuk menjual kredensil sementara kepada penerima hibah yang mengakses lokasi tertentu. Anda harus terlebih dahulu mendaftarkan setidaknya satu lokasi di instans S3 Access Grants sebelum Anda dapat membuat hibah akses.

Kasus penggunaan yang disarankan

Kami menyarankan Anda mendaftarkan lokasi default (`s3://`) dan memetakannya ke peran IAM. Lokasi di jalur S3 default (`s3://`) mencakup akses ke semua bucket S3 Anda di akun Anda Wilayah AWS . Saat membuat hibah akses, Anda dapat mempersempit cakupan hibah menjadi bucket, awalan, atau objek di dalam lokasi default.

Kasus penggunaan manajemen akses yang kompleks

Kasus penggunaan manajemen akses yang lebih kompleks mungkin mengharuskan Anda untuk mendaftar lebih dari lokasi default. Beberapa contoh kasus penggunaan tersebut adalah:

- Misalkan lokasi terdaftar di instance S3 Access Grants Anda dengan peran IAM yang dipetakan ke sana, tetapi peran IAM ini ditolak akses ke awalan tertentu di dalam bucket. *amzn-s3-demo-bucket* Dalam hal ini, Anda dapat mendaftarkan awalan yang tidak dapat diakses oleh peran IAM sebagai lokasi terpisah dan memetakan lokasi tersebut ke peran IAM yang berbeda dengan akses yang diperlukan.
- Misalkan Anda ingin membuat hibah yang membatasi akses hanya ke pengguna dalam titik akhir virtual private cloud (VPC). Dalam hal ini, Anda dapat mendaftarkan lokasi untuk bucket di mana peran IAM membatasi akses ke titik akhir VPC. Kemudian, ketika penerima hibah meminta kredensil S3 Access Grants, S3 Access Grants mengasumsikan peran IAM lokasi untuk menjual kredenal sementara. Kredensi ini akan menolak akses ke bucket tertentu kecuali penelepon berada dalam titik akhir VPC. Izin penolakan ini diterapkan selain izin BACA, TULIS, atau READWRITE reguler yang ditentukan dalam hibah.

Saat mendaftarkan lokasi, Anda juga harus menentukan peran IAM yang diasumsikan oleh S3 Access Grants untuk menjual kredensi sementara dan untuk mencakup izin untuk hibah tertentu.

Jika kasus penggunaan Anda mengharuskan Anda mendaftarkan beberapa lokasi di instans S3 Access Grants, Anda dapat mendaftarkan salah satu dari berikut ini:

URI S3	Peran IAM	Deskripsi
`s3://`	*Default-IAM-role*	Lokasi default, `s3://`, mencakup semua bucket di file Wilayah AWS.
`s3://`*amzn-s3-demo-bucket1* `/`	*IAM-role-For-bucket*	Lokasi ini mencakup semua objek dalam bucket yang ditentukan.

URI S3	Peran IAM	Deskripsi
`s3://amzn-s3-demo-bucket1 /prefix-name`	*IAM-role-For-prefix*	Lokasi ini mencakup semua objek dalam bucket dengan nama kunci objek yang dimulai dengan awalan ini.

Sebelum Anda dapat mendaftarkan bucket atau awalan tertentu, pastikan Anda melakukan hal berikut:

- Buat satu atau beberapa bucket yang berisi data yang ingin Anda berikan akses. Bucket ini harus ditempatkan Wilayah AWS sama dengan instance S3 Access Grants Anda. Untuk informasi selengkapnya, lihat Membuat bucket.

  Menambahkan awalan adalah langkah opsional. Awalan adalah string di awal nama kunci objek. Anda dapat menggunakannya untuk mengatur objek di bucket Anda serta untuk manajemen akses. Untuk menambahkan prefiks ke bucket, lihat Membuat nama kunci objek.
- Buat peran IAM yang memiliki izin untuk mengakses data S3 Anda di file. Wilayah AWS Untuk informasi selengkapnya, lihat Membuat peran IAM di panduan AWS IAM Identity Center pengguna.
- Dalam kebijakan kepercayaan peran IAM, berikan akses utama layanan S3 Access Grants (`access-grants.s3.amazonaws.com`) ke peran IAM yang Anda buat. Untuk melakukannya, Anda dapat membuat file JSON yang berisi pernyataan berikut ini. Untuk menambahkan kebijakan kepercayaan ke akun Anda, lihat Membuat peran menggunakan kebijakan kepercayaan khusus.

  TestRolePolicy.json

```
{
  "Version": "2012-10-17",
    "Statement": [
    {
      "Sid": "TestRolePolicy",
      "Effect": "Allow",
      "Principal": {
        "Service":"access-grants.s3.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRole",
        "sts:SetSourceIdentity"
      ],
      "Condition": {
```

```
        "StringEquals": {
          "aws:SourceAccount":"accountId",
          "aws:SourceArn":"arn:aws:s3:region:accountId:access-grants/default"
        }
      }
    }
  ]
}
```

Atau, untuk kasus penggunaan Pusat Identitas IAM, gunakan kebijakan berikut yang mencakup pernyataan kedua:

```
{
  "Version": "2012-10-17",
    "Statement": [
    {
      "Sid": "TestRolePolicyIdentityCenter",
      "Effect": "Allow",
      "Principal": {
        "Service":"access-grants.s3.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRole",
        "sts:SetSourceIdentity"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount":"accountId",
          "aws:SourceArn":"arn:aws:s3:region:accountId:access-grants/default"
        }
      }
    },
    {
      "Sid": "Stmt1234567891012",
        "Effect": "Allow",
        "Principal": {
          "Service": "access-grants.s3.amazonaws.com"
        },
        "Action": "sts:SetContext",
        "Condition":{
          "StringEquals":{
            "aws:SourceAccount":"accountId",
            "aws:SourceArn":"arn:aws:s3:region:accountId:access-grants/default"
```

```
          },
          "ForAllValues:ArnEquals": {
            "sts:RequestContextProviders":"arn:aws:iam::aws:contextProvider/
IdentityCenter"
          }
        }
      }
  ]
}
```

- Buat kebijakan IAM untuk melampirkan izin Amazon S3 ke peran IAM yang Anda buat. Lihat file contoh `iam-policy.json` berikut ini, dan ganti *`user input placeholders`* dengan informasi Anda sendiri.

> **Note**
>
> - Jika Anda menggunakan enkripsi sisi server dengan kunci AWS Key Management Service (AWS KMS) untuk mengenkripsi data Anda, contoh berikut menyertakan AWS KMS izin yang diperlukan untuk peran IAM dalam kebijakan. Jika Anda tidak menggunakan fitur ini, Anda dapat menghapus izin ini dari kebijakan IAM Anda.
> - Anda dapat membatasi peran IAM untuk mengakses data S3 hanya jika kredensialnya dijual oleh S3 Access Grants. Contoh ini menunjukkan cara menambahkan `Condition` pernyataan untuk instance S3 Access Grants tertentu. Untuk menggunakan ini`Condition`, ganti ARN instance S3 Access Grants dalam `Condition` pernyataan dengan ARN instance S3 Access Grants Anda, yang memiliki format: `arn:aws:s3:`*`region`*`:`*`accountId`*`:access-grants/default`

iam-policy.json

```
{
   "Version":"2012-10-17",
   "Statement": [
       {
         "Sid": "ObjectLevelReadPermissions",
         "Effect":"Allow",
         "Action":[
            "s3:GetObject",
            "s3:GetObjectVersion",
```

```
            "s3:GetObjectAcl",
            "s3:GetObjectVersionAcl",
            "s3:ListMultipartUploadParts"
         ],
         "Resource":[
            "arn:aws:s3:::*"
         ],
         "Condition":{
            "StringEquals": { "aws:ResourceAccount": "accountId" },
            "ArnEquals": {
                "s3:AccessGrantsInstanceArn": ["arn:aws:s3:region:accountId:access-
grants/default"]
            }
        }
      },
      {
         "Sid": "ObjectLevelWritePermissions",
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:PutObjectAcl",
            "s3:PutObjectVersionAcl",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion",
            "s3:AbortMultipartUpload"
         ],
         "Resource":[
            "arn:aws:s3:::*"
         ],
         "Condition":{
            "StringEquals": { "aws:ResourceAccount": "accountId" },
            "ArnEquals": {
                "s3:AccessGrantsInstanceArn": ["arn:aws:s3:Wilayah
 AWS:accountId:access-grants/default"]
            }
         }
      },
      {
         "Sid": "BucketLevelReadPermissions",
         "Effect":"Allow",
         "Action":[
            "s3:ListBucket"
         ],
         "Resource":[
```

```
            "arn:aws:s3:::*"
         ],
         "Condition":{
            "StringEquals": { "aws:ResourceAccount": "accountId" },
            "ArnEquals": {
                "s3:AccessGrantsInstanceArn": ["arn:aws:s3:Wilayah
 AWS:accountId:access-grants/default"]
            }
         }
      },
   //Optionally add the following section if you use SSE-KMS encryption
      {
         "Sid": "KMSPermissions",
         "Effect":"Allow",
         "Action":[
            "kms:Decrypt",
            "kms:GenerateDataKey"
         ],
         "Resource":[
            "*"
         ]
      }
   ]
}
```

Anda dapat mendaftarkan lokasi di instans S3 Access Grants dengan menggunakan konsol Amazon S3,AWS CLI(), AWS Command Line Interface Amazon S3 REST API, atau. AWS SDKs

> **Note**
>
> Setelah Anda mendaftarkan lokasi pertama di instans S3 Access Grants, instans Anda masih tidak memiliki hibah akses individual di dalamnya. Untuk membuat hibah akses, lihatMembuat pemberian.

Menggunakan konsol S3

Sebelum Anda dapat memberikan akses ke data S3 Anda dengan S3 Access Grants, Anda harus memiliki setidaknya satu lokasi terdaftar.

Untuk mendaftarkan lokasi di instans S3 Access Grants

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Pada panel navigasi di sebelah kiri, pilih Access Grants.
3. Pada halaman S3 Access Grants, pilih Region yang berisi instance S3 Access Grants yang ingin Anda gunakan.

   Jika Anda menggunakan instans S3 Access Grants untuk pertama kalinya, pastikan Anda telah menyelesaikan Langkah 1-buat instans S3 Access Grants dan navigasikan ke Langkah 2 dari wizard instans Set up Access Grants. Jika Anda sudah memiliki instans S3 Access Grants, pilih Lihat detail, lalu dari tab Lokasi, pilih Daftar lokasi.

   a. Untuk lingkup Lokasi, pilih Jelajahi S3 atau masukkan jalur URI S3 ke lokasi yang ingin Anda daftarkan. Untuk format URI S3, lihat tabel format lokasi. Setelah memasukkan URI, Anda dapat memilih Lihat untuk menelusuri lokasi.

   b. Untuk Peran IAM, pilih salah satu hal berikut ini:

      - Pilih dari peran IAM yang ada

        Pilih peran IAM dari daftar dropdown. Setelah Anda memilih peran, pilih Lihat untuk memastikan bahwa peran ini memiliki izin yang diperlukan untuk mengelola lokasi yang Anda daftarkan. Secara kustom, pastikan bahwa peran ini memberikan S3 Access memberikan izin `sts:AssumeRole` dan `sts:SetSourceIdentity`.

      - Masukkan ARN peran IAM

        Arahkan ke Konsol IAM. Salin Nama Sumber Daya Amazon (ARN) dari peran IAM dan tempel di kotak ini.

   c. Untuk menyelesaikan, pilih Berikutnya atau Daftar lokasi.

4. Pemecahan Masalah:

   Tidak dapat mendaftarkan lokasi

   - Lokasi mungkin sudah terdaftar.

     Anda mungkin tidak memiliki `s3:CreateAccessGrantsLocation` izin untuk mendaftarkan lokasi. Menghubungi administrator akun Anda.

Menggunakan AWS CLI

Untuk menginstal AWS CLI, lihat Menginstal AWS CLI di Panduan AWS Command Line Interface Pengguna.

Anda dapat mendaftarkan lokasi default, `s3://`, atau lokasi kustom di instans S3 Access Grants Anda. Pastikan Anda terlebih dahulu membuat peran IAM dengan akses pengguna utama ke lokasi, lalu pastikan Anda memberikan izin S3 Access Grants untuk mengambil peran ini.

Untuk menggunakan perintah contoh berikut, ganti *`user input placeholders`* dengan informasi Anda sendiri.

Example Buat kebijakan sumber daya

Buat kebijakan yang memungkinkan S3 Access Grants untuk mengambil peran IAM. Untuk melakukannya, Anda dapat membuat file JSON yang berisi pernyataan berikut ini. Untuk menambahkan kebijakan sumber daya ke akun Anda, lihat Membuat dan melampirkan kebijakan yang dikelola pelanggan pertama Anda.

TestRolePolicy.json

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1234567891011",
      "Action": ["sts:AssumeRole", "sts:SetSourceIdentity"],
      "Effect": "Allow",
      "Principal": {"Service":"access-grants.s3.amazonaws.com"}
    }
  ]
}
```

Example Buat peran

Jalankan perintah IAM berikut untuk membuat peran.

```
aws iam create-role --role-name accessGrantsTestRole \
 --region us-east-2 \
 --assume-role-policy-document file://TestRolePolicy.json
```

Menjalankan `create-role` perintah mengembalikan kebijakan:

```
{
    "Role": {
        "Path": "/",
        "RoleName": "accessGrantsTestRole",
        "RoleId": "AROASRDGX4WM4GH55GIDA",
        "Arn": "arn:aws:iam::111122223333:role/accessGrantsTestRole",
        "CreateDate": "2023-05-31T18:11:06+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Sid": "Stmt1685556427189",
                    "Action": [
                        "sts:AssumeRole",
                        "sts:SetSourceIdentity"
                    ],
                    "Effect": "Allow",
                    "Principal": {
                        "Service":"access-grants.s3.amazonaws.com"
                    }
                }
            ]
        }
    }
}
```

Example

Buat kebijakan IAM untuk melampirkan izin Amazon S3 ke peran IAM. Lihat file contoh `iam-policy.json` berikut ini, dan ganti *`user input placeholders`* dengan informasi Anda sendiri.

> **Note**
>
> Jika Anda menggunakan enkripsi sisi server dengan kunci AWS Key Management Service (AWS KMS) untuk mengenkripsi data, contoh berikut akan menambahkan AWS KMS izin yang diperlukan untuk peran IAM dalam kebijakan. Jika Anda tidak menggunakan fitur ini, Anda dapat menghapus izin ini dari kebijakan IAM Anda.
> Untuk memastikan bahwa peran IAM hanya dapat digunakan untuk mengakses data di S3 jika kredensialnya dijual oleh S3 Access Grants, contoh ini menunjukkan cara menambahkan pernyataan `Condition` pernyataan yang menentukan instans S3 Access

Grants (`s3:AccessGrantsInstance:` *`InstanceArn`*) dalam kebijakan IAM Anda. Saat menggunakan contoh kebijakan berikut, ganti *`user input placeholders`* dengan informasi Anda sendiri.

iam-policy.json

```
{
   "Version":"2012-10-17",
   "Statement": [
       {
         "Sid": "ObjectLevelReadPermissions",
         "Effect":"Allow",
         "Action":[
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:GetObjectAcl",
            "s3:GetObjectVersionAcl",
            "s3:ListMultipartUploadParts"
         ],
         "Resource":[
            "arn:aws:s3:::*"
         ],
         "Condition":{
            "StringEquals": { "aws:ResourceAccount": "accountId" },
            "ArnEquals": {
                "s3:AccessGrantsInstanceArn": ["arn:aws:s3:region:accountId:access-
grants/default"]
            }
        }
      },
      {
         "Sid": "ObjectLevelWritePermissions",
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:PutObjectAcl",
            "s3:PutObjectVersionAcl",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion",
            "s3:AbortMultipartUpload"
         ],
```

```
            "Resource":[
               "arn:aws:s3:::*"
            ],
            "Condition":{
               "StringEquals": { "aws:ResourceAccount": "accountId" },
               "ArnEquals": {
                   "s3:AccessGrantsInstanceArn": ["arn:aws:s3:Wilayah
 AWS:accountId:access-grants/default"]
               }
            }
         },
         {
            "Sid": "BucketLevelReadPermissions",
            "Effect":"Allow",
            "Action":[
               "s3:ListBucket"
            ],
            "Resource":[
               "arn:aws:s3:::*"
            ],
            "Condition":{
               "StringEquals": { "aws:ResourceAccount": "accountId" },
               "ArnEquals": {
                   "s3:AccessGrantsInstanceArn": ["arn:aws:s3:Wilayah
 AWS:accountId:access-grants/default"]
               }
            }
         },
         {
            "Sid": "KMSPermissions",
            "Effect":"Allow",
            "Action":[
               "kms:Decrypt",
               "kms:GenerateDataKey"
            ],
            "Resource":[
               "*"
            ]
         }
      ]
}
```

Example

Jalankan perintah berikut:

```
aws iam put-role-policy \
--role-name accessGrantsTestRole \
--policy-name accessGrantsTestRole \
--policy-document file://iam-policy.json
```

Example Daftarkan lokasi default

```
aws s3control create-access-grants-location \
 --account-id 111122223333 \
 --location-scope s3:// \
 --iam-role-arn arn:aws:iam::111122223333:role/accessGrantsTestRole
```

Respons:

```
{"CreatedAt": "2023-05-31T18:23:48.107000+00:00",
    "AccessGrantsLocationId": "default",
    "AccessGrantsLocationArn": "arn:aws:s3:us-east-2:111122223333:access-grants/
default/location/default",
    "LocationScope": "s3://"
    "IAMRoleArn": "arn:aws:iam::111122223333:role/accessGrantsTestRole"
}
```

Example Daftarkan lokasi kustom

```
aws s3control create-access-grants-location \
 --account-id 111122223333 \
 --location-scope s3://DOC-BUCKET-EXAMPLE/ \
 --iam-role-arn arn:aws:iam::123456789012:role/accessGrantsTestRole
```

Respons:

```
{"CreatedAt": "2023-05-31T18:23:48.107000+00:00",
    "AccessGrantsLocationId": "635f1139-1af2-4e43-8131-a4de006eb456",
    "AccessGrantsLocationArn": "arn:aws:s3:us-east-2: 111122223333:access-grants/
default/location/635f1139-1af2-4e43-8131-a4de006eb888",
    "LocationScope": "s3://DOC-BUCKET-EXAMPLE/",
    "IAMRoleArn": "arn:aws:iam::111122223333:role/accessGrantsTestRole"
```

```
}
```

Penggunaan API REST

Untuk informasi tentang dukungan API REST Amazon S3 untuk mengelola instans S3 Access Grants, lihat bagian berikut di Referensi API Amazon Simple Storage Service:

- CreateAccessGrantsLocation
- DeleteAccessGrantsLocation
- GetAccessGrantsLocation
- ListAccessGrantsLocations
- UpdateAccessGrantsLocation

Menggunakan AWS SDKs

Bagian ini memberikan contoh cara mendaftarkan lokasi dengan menggunakan AWS SDKs.

Untuk menggunakan contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri.

Java

Anda dapat mendaftarkan lokasi default, `s3://`, atau lokasi kustom di instans S3 Access Grants Anda. Pastikan Anda terlebih dahulu membuat peran IAM dengan akses pengguna utama ke lokasi, lalu pastikan Anda memberikan izin S3 Access Grants untuk mengambil peran ini.

Untuk menggunakan perintah contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri.

Example Daftarkan lokasi default

Permintaan:

```
public void createAccessGrantsLocation() {
CreateAccessGrantsLocationRequest createRequest =
 CreateAccessGrantsLocationRequest.builder()
.accountId("111122223333")
.locationScope("s3://")
.iamRoleArn("arn:aws:iam::123456789012:role/accessGrantsTestRole")
.build();
```

```
CreateAccessGrantsLocationResponse createResponse =
 s3Control.createAccessGrantsLocation(createRequest);
LOGGER.info("CreateAccessGrantsLocationResponse: " + createResponse);
}
```

Respons:

```
CreateAccessGrantsLocationResponse(
CreatedAt=2023-06-07T04:35:11.027Z,
AccessGrantsLocationId=default,
AccessGrantsLocationArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
location/default,
LocationScope=s3://,
IAMRoleArn=arn:aws:iam::111122223333:role/accessGrantsTestRole
)
```

Example Daftarkan lokasi kustom

Permintaan:

```
public void createAccessGrantsLocation() {
CreateAccessGrantsLocationRequest createRequest =
 CreateAccessGrantsLocationRequest.builder()
.accountId("111122223333")
.locationScope("s3://DOC-BUCKET-EXAMPLE/")
.iamRoleArn("arn:aws:iam::111122223333:role/accessGrantsTestRole")
.build();
CreateAccessGrantsLocationResponse createResponse =
 s3Control.createAccessGrantsLocation(createRequest);
LOGGER.info("CreateAccessGrantsLocationResponse: " + createResponse);
}
```

Respons:

```
CreateAccessGrantsLocationResponse(
CreatedAt=2023-06-07T04:35:10.027Z,
AccessGrantsLocationId=18cfe6fb-eb5a-4ac5-aba9-8d79f04c2012,
AccessGrantsLocationArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
location/18cfe6fb-eb5a-4ac5-aba9-8d79f04c2666,
LocationScope= s3://test-bucket-access-grants-user123/,
IAMRoleArn=arn:aws:iam::111122223333:role/accessGrantsTestRole
)
```

Lihat detail lokasi yang terdaftar

Anda bisa mendapatkan detail lokasi yang terdaftar di instans S3 Access Grants dengan menggunakan konsol Amazon S3,AWS CLI(), AWS Command Line Interface Amazon S3 REST API, dan file. AWS SDKs

Menggunakan konsol S3

Untuk melihat lokasi yang terdaftar di instans S3 Access Grants

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Pada panel navigasi di sebelah kiri, pilih Access Grants.
3. Pada halaman S3 Access Grants, pilih Region yang berisi instance S3 Access Grants yang ingin Anda gunakan.
4. Pilih Lihat detail untuk contoh.
5. Pada halaman detail untuk contoh, pilih tab Lokasi.
6. Temukan lokasi terdaftar yang ingin Anda lihat. Untuk memfilter daftar lokasi terdaftar, gunakan kotak pencarian.

Menggunakan AWS CLI

Untuk menginstal AWS CLI, lihat Menginstal AWS CLI di Panduan AWS Command Line Interface Pengguna.

Untuk menggunakan perintah contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri.

Example — Dapatkan detail lokasi yang terdaftar

```
aws s3control get-access-grants-location \
--account-id 111122223333 \
--access-grants-location-id default
```

Respons:

```
{
    "CreatedAt": "2023-05-31T18:23:48.107000+00:00",
```

```
    "AccessGrantsLocationId": "default",
    "AccessGrantsLocationArn": "arn:aws:s3:us-east-2:111122223333:access-grants/
default/location/default",
    "IAMRoleArn": "arn:aws:iam::111122223333:role/accessGrantsTestRole"
}
```

Example — Daftar semua lokasi yang terdaftar dalam instance S3 Access Grants

Untuk membatasi hasil ke awalan atau bucket S3, Anda dapat menggunakan parameter secara opsional. `--location-scope s3://bucket-and-or-prefix`

```
aws s3control list-access-grants-locations \
--account-id 111122223333 \
--region us-east-2
```

Respons:

```
{"AccessGrantsLocationsList": [
  {
    "CreatedAt": "2023-05-31T18:23:48.107000+00:00",
    "AccessGrantsLocationId": "default",
    "AccessGrantsLocationArn": "arn:aws:s3:us-east-2:111122223333:access-grants/
default/location/default",
    "LocationScope": "s3://"
    "IAMRoleArn": "arn:aws:iam::111122223333:role/accessGrantsTestRole"
     },
  {
    "CreatedAt": "2023-05-31T18:23:48.107000+00:00",
    "AccessGrantsLocationId": "635f1139-1af2-4e43-8131-a4de006eb456",
    "AccessGrantsLocationArn": "arn:aws:s3:us-east-2:111122223333:access-grants/
default/location/635f1139-1af2-4e43-8131-a4de006eb888",
    "LocationScope": "s3://amzn-s3-demo-bucket/prefixA*",
    "IAMRoleArn": "arn:aws:iam::111122223333:role/accessGrantsTestRole"
     }
   ]
  }
```

Penggunaan API REST

Untuk informasi tentang dukungan Amazon S3 REST API untuk mendapatkan detail lokasi terdaftar atau mencantumkan semua lokasi yang terdaftar dengan instans S3 Access Grants, lihat bagian berikut di Referensi API Layanan Penyimpanan Sederhana Amazon:

- GetAccessGrantsLocation
- ListAccessGrantsLocations

Menggunakan AWS SDKs

Bagian ini memberikan contoh cara mendapatkan detail lokasi terdaftar atau mencantumkan semua lokasi terdaftar dalam instance S3 Access Grants dengan menggunakan. AWS SDKs

Untuk menggunakan contoh berikut, ganti *`user input placeholders`* dengan informasi Anda sendiri.

Java

Example — Dapatkan detail lokasi yang terdaftar

```
public void getAccessGrantsLocation() {
GetAccessGrantsLocationRequest getAccessGrantsLocationRequest =
 GetAccessGrantsLocationRequest.builder()
.accountId("111122223333")
.accessGrantsLocationId("default")
.build();
GetAccessGrantsLocationResponse getAccessGrantsLocationResponse =
 s3Control.getAccessGrantsLocation(getAccessGrantsLocationRequest);
LOGGER.info("GetAccessGrantsLocationResponse: " + getAccessGrantsLocationResponse);
}
```

Respons:

```
GetAccessGrantsLocationResponse(
CreatedAt=2023-06-07T04:35:10.027Z,
AccessGrantsLocationId=default,
AccessGrantsLocationArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
location/default,
LocationScope= s3://,
IAMRoleArn=arn:aws:iam::111122223333:role/accessGrantsTestRole
)
```

Example — Daftar semua lokasi terdaftar dalam instance S3 Access Grants

Untuk membatasi hasil ke awalan atau bucket S3, Anda dapat meneruskan URI S3 secara opsional, seperti`s3://`*`bucket-and-or-prefix`*, dalam parameter. `LocationScope`

```
public void listAccessGrantsLocations() {

ListAccessGrantsLocationsRequest listRequest =
 ListAccessGrantsLocationsRequest.builder()
.accountId("111122223333")
.build();

ListAccessGrantsLocationsResponse listResponse =
 s3Control.listAccessGrantsLocations(listRequest);
LOGGER.info("ListAccessGrantsLocationsResponse: " + listResponse);
}
```

Respons:

```
ListAccessGrantsLocationsResponse(
AccessGrantsLocationsList=[
ListAccessGrantsLocationsEntry(
CreatedAt=2023-06-07T04:35:11.027Z,
AccessGrantsLocationId=default,
AccessGrantsLocationArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
location/default,
LocationScope=s3://,
IAMRoleArn=arn:aws:iam::111122223333:role/accessGrantsTestRole
),
ListAccessGrantsLocationsEntry(
CreatedAt=2023-06-07T04:35:10.027Z,
AccessGrantsLocationId=635f1139-1af2-4e43-8131-a4de006eb456,
AccessGrantsLocationArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
location/635f1139-1af2-4e43-8131-a4de006eb888,
LocationScope=s3://amzn-s3-demo-bucket/prefixA*,
IAMRoleArn=arn:aws:iam::111122223333:role/accessGrantsTestRole
)
]
)
```

Memperbarui lokasi terdaftar

Anda dapat memperbarui peran AWS Identity and Access Management (IAM) lokasi yang terdaftar di instans Amazon S3 Access Grants. Untuk setiap peran IAM baru yang Anda gunakan untuk mendaftarkan lokasi di S3 Access Grants, pastikan untuk memberikan akses utama layanan S3 Access Grants (`access-grants.s3.amazonaws.com`) ke peran ini. Untuk melakukannya,

tambahkan entri untuk peran IAM baru dalam file JSON kebijakan kepercayaan yang sama yang Anda gunakan saat pertama kali mendaftarkan lokasi.

Anda dapat memperbarui lokasi di instans S3 Access Grants dengan menggunakan konsol Amazon S3,AWS CLI(), AWS Command Line Interface Amazon S3 REST API, dan. AWS SDKs

Menggunakan konsol S3

Untuk memperbarui peran IAM dari lokasi yang terdaftar dengan instans S3 Access Grants

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Pada panel navigasi di sebelah kiri, pilih Access Grants.
3. Pada halaman S3 Access Grants, pilih Region yang berisi instance S3 Access Grants yang ingin Anda gunakan.
4. Pilih Lihat detail untuk contoh.
5. Pada halaman detail untuk contoh, pilih tab Lokasi.
6. Temukan lokasi yang ingin Anda perbarui. Untuk memfilter daftar lokasi, gunakan kotak pencarian.
7. Pilih tombol opsi di sebelah lokasi terdaftar yang ingin Anda perbarui.
8. Perbarui peran IAM, lalu pilih Simpan perubahan.

Menggunakan AWS CLI

Untuk menginstal AWS CLI, lihat Menginstal AWS CLI di Panduan AWS Command Line Interface Pengguna.

Untuk menggunakan perintah contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri.

Example — Perbarui peran IAM dari lokasi terdaftar

```
aws s3control update-access-grants-location \
--account-id 111122223333 \
--access-grants-location-id 635f1139-1af2-4e43-8131-a4de006eb999 \
--iam-role-arn arn:aws:iam::777788889999:role/accessGrantsTestRole
```

Respons:

```
{
    "CreatedAt": "2023-05-31T18:23:48.107000+00:00",
    "AccessGrantsLocationId": "635f1139-1af2-4e43-8131-a4de006eb999",
    "AccessGrantsLocationArn": "arn:aws:s3:us-east-2:777788889999:access-grants/
default/location/635f1139-1af2-4e43-8131-a4de006eb888",
    "LocationScope": "s3://amzn-s3-demo-bucket/prefixB*",
    "IAMRoleArn": "arn:aws:iam::777788889999:role/accessGrantsTestRole"
}
```

Penggunaan API REST

Untuk informasi tentang dukungan Amazon S3 REST API untuk memperbarui lokasi di instans S3 Access Grants, lihat di Referensi API UpdateAccessGrantsLocationLayanan Penyimpanan Sederhana Amazon.

Menggunakan AWS SDKs

Bagian ini memberikan contoh cara memperbarui peran IAM dari lokasi terdaftar dengan menggunakan. AWS SDKs

Untuk menggunakan contoh berikut, ganti *`user input placeholders`* dengan informasi Anda sendiri.

Java

Example — Perbarui peran IAM dari lokasi terdaftar

```
public void updateAccessGrantsLocation() {
UpdateAccessGrantsLocationRequest updateRequest =
 UpdateAccessGrantsLocationRequest.builder()
.accountId("111122223333")
.accessGrantsLocationId("635f1139-1af2-4e43-8131-a4de006eb999")
.iamRoleArn("arn:aws:iam::777788889999:role/accessGrantsTestRole")
.build();
UpdateAccessGrantsLocationResponse updateResponse =
 s3Control.updateAccessGrantsLocation(updateRequest);
LOGGER.info("UpdateAccessGrantsLocationResponse: " + updateResponse);
}
```

Respons:

```
UpdateAccessGrantsLocationResponse(
```

```
CreatedAt=2023-06-07T04:35:10.027Z,
AccessGrantsLocationId=635f1139-1af2-4e43-8131-a4de006eb999,
AccessGrantsLocationArn=arn:aws:s3:us-east-2:777788889999:access-grants/default/
location/635f1139-1af2-4e43-8131-a4de006eb888,
LocationScope=s3://amzn-s3-demo-bucket/prefixB*,
IAMRoleArn=arn:aws:iam::777788889999:role/accessGrantsTestRole
)
```

Hapus lokasi terdaftar

Anda dapat menghapus pendaftaran lokasi dari instans Amazon S3 Access Grants. Menghapus lokasi membatalkan pendaftarannya dari instance S3 Access Grants.

Sebelum Anda dapat menghapus pendaftaran lokasi dari instance S3 Access Grants, Anda harus menghapus semua hibah yang terkait dengan lokasi ini. Untuk informasi tentang cara menghapus hibah, lihat Menghapus hibah.

Anda dapat menghapus lokasi di instans S3 Access Grants dengan menggunakan konsol Amazon S3,AWS CLI(), AWS Command Line Interface Amazon S3 REST API, dan. AWS SDKs

Menggunakan konsol S3

Untuk menghapus pendaftaran lokasi dari instans S3 Access Grants

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Pada panel navigasi di sebelah kiri, pilih Access Grants.
3. Pada halaman S3 Access Grants, pilih Region yang berisi instance S3 Access Grants yang ingin Anda gunakan.
4. Pilih Lihat detail untuk contoh.
5. Pada halaman detail untuk contoh, pilih tab Lokasi.
6. Temukan lokasi yang ingin Anda perbarui. Untuk memfilter daftar lokasi, gunakan kotak pencarian.
7. Pilih tombol opsi di sebelah lokasi terdaftar yang ingin Anda hapus.
8. Pilih Batalkan pendaftaran.
9. Kotak dialog muncul yang memperingatkan Anda bahwa tindakan ini tidak dapat dibatalkan. Untuk menghapus lokasi, pilih Deregister.

Menggunakan AWS CLI

Untuk menginstal AWS CLI, lihat Menginstal AWS CLI di Panduan AWS Command Line Interface Pengguna.

Untuk menggunakan perintah contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri.

Example — Hapus pendaftaran lokasi

```
aws s3control delete-access-grants-location \
--account-id 111122223333 \
--access-grants-location-id  a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
 // No response body
```

Penggunaan API REST

Untuk informasi tentang dukungan Amazon S3 REST API untuk menghapus lokasi dari instance S3 Access Grants, lihat di DeleteAccessGrantsLocationReferensi API Layanan Penyimpanan Sederhana Amazon.

Menggunakan AWS SDKs

Bagian ini memberikan contoh cara menghapus lokasi dengan menggunakan AWS SDKs.

Untuk menggunakan contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri.

Java

Example — Hapus pendaftaran lokasi

```
public void deleteAccessGrantsLocation() {
DeleteAccessGrantsLocationRequest deleteRequest =
 DeleteAccessGrantsLocationRequest.builder()
.accountId("111122223333")
.accessGrantsLocationId("a1b2c3d4-5678-90ab-cdef-EXAMPLE11111")
.build();
DeleteAccessGrantsLocationResponse deleteResponse =
 s3Control.deleteAccessGrantsLocation(deleteRequest);
LOGGER.info("DeleteAccessGrantsLocationResponse: " + deleteResponse);
```

```
}
```

Respons:

```
DeleteAccessGrantsLocationResponse()
```

Bekerja dengan hibah di S3 Access Grants

Pemberian akses individual dalam instans S3 Access Grants memungkinkan identitas tertentu —prinsipal AWS Identity and Access Management (IAM), atau pengguna atau grup di direktori perusahaan—untuk mendapatkan akses di dalam lokasi yang terdaftar di instans S3 Access Grants Anda. Lokasi memetakan bucket atau awalan ke peran IAM. S3 Access Grants mengasumsikan peran IAM ini untuk menjual kredensil sementara kepada penerima hibah.

Setelah mendaftarkan setidaknya satu lokasi di instans S3 Access Grants, Anda dapat membuat hibah akses.

Penerima hibah dapat berupa pengguna IAM atau peran atau pengguna direktori atau grup. Pengguna direktori adalah pengguna dari direktori perusahaan atau sumber identitas eksternal yang terkait dengan instans S3 Access Grants Anda. Untuk informasi selengkapnya, lihat S3 Access Grants dan identitas direktori perusahaan. Untuk membuat hibah untuk pengguna atau grup direktori tertentu dari IAM Identity Center, temukan GUID yang digunakan IAM Identity Center untuk mengidentifikasi pengguna tersebut di Pusat Identitas IAM, misalnya,. `a1b2c3d4-5678-90ab-cdef-EXAMPLE11111` Untuk informasi selengkapnya tentang cara menggunakan Pusat Identitas IAM untuk melihat informasi pengguna, lihat Melihat penetapan pengguna dan grup di panduan AWS IAM Identity Center pengguna.

Anda dapat memberikan akses ke bucket, awalan, atau objek. Prefiks di Amazon S3 adalah string karakter di awal nama kunci objek yang digunakan untuk mengatur objek dalam bucket. Ini bisa berupa string karakter yang diizinkan, misalnya, nama kunci objek di bucket Anda yang dimulai dengan `engineering/` awalan.

Topik

- Membuat pemberian
- Lihat izin
- Hapus izin

Membuat pemberian

Pemberian akses individual dalam instans S3 Access Grants memungkinkan identitas tertentu —prinsipal AWS Identity and Access Management (IAM), atau pengguna atau grup di direktori perusahaan—untuk mendapatkan akses di dalam lokasi yang terdaftar di instans S3 Access Grants Anda. Lokasi memetakan bucket atau awalan ke peran IAM. S3 Access Grants mengasumsikan peran IAM ini untuk menjual kredensil sementara kepada penerima hibah.

Setelah mendaftarkan setidaknya satu lokasi di instans S3 Access Grants, Anda dapat membuat hibah akses.

Penerima hibah dapat berupa pengguna IAM atau peran atau pengguna direktori atau grup. Pengguna direktori adalah pengguna dari direktori perusahaan atau sumber identitas eksternal yang terkait dengan instans S3 Access Grants Anda. Untuk informasi selengkapnya, lihat S3 Access Grants dan identitas direktori perusahaan. Untuk membuat hibah untuk pengguna atau grup direktori tertentu dari IAM Identity Center, temukan GUID yang digunakan IAM Identity Center untuk mengidentifikasi pengguna tersebut di Pusat Identitas IAM, misalnya,. `a1b2c3d4-5678-90ab-cdef-EXAMPLE11111` Untuk informasi selengkapnya tentang cara menggunakan Pusat Identitas IAM untuk melihat informasi pengguna, lihat Melihat penetapan pengguna dan grup di panduan AWS IAM Identity Center pengguna.

Anda dapat memberikan akses ke bucket, awalan, atau objek. Prefiks di Amazon S3 adalah string karakter di awal nama kunci objek yang digunakan untuk mengatur objek dalam bucket. Ini bisa berupa string karakter yang diizinkan, misalnya, nama kunci objek di bucket Anda yang dimulai dengan `engineering/` awalan.

Subprefiks

Saat memberikan akses ke lokasi terdaftar, Anda dapat menggunakan `Subprefix` bidang untuk mempersempit cakupan akses ke subset cakupan lokasi. Jika lokasi terdaftar yang Anda pilih untuk hibah adalah jalur S3 default (`s3://`), Anda harus mempersempit cakupan hibah. Anda tidak dapat membuat hibah akses untuk lokasi default (`s3://`), yang akan memberikan akses kepada penerima hibah ke setiap bucket dalam file. Wilayah AWS Sebagai gantinya, Anda harus mempersempit ruang lingkup hibah menjadi salah satu dari yang berikut:

- Sebuah ember: `s3://`*`bucket`*`/*`
- Awalan dalam ember: `s3://`*`bucket`*`/`*`prefix`*`*`
- Awalan dalam awalan: `s3://`*`bucket`*`/`*`prefixA`*`/`*`prefixB`*`*`

- Sebuah objek: `s3://`*`bucket`*`/`*`object-key-name`*

Jika Anda membuat hibah akses di mana lokasi terdaftar adalah bucket, Anda dapat meneruskan salah satu dari berikut ini di `Subprefix` bidang untuk mempersempit cakupan hibah:

- Awalan di dalam ember: *`prefix`**
- Awalan dalam awalan: *`prefixA/prefixB`**
- Sebuah objek: */`*`object-key-name`*

Setelah Anda membuat hibah, cakupan hibah yang ditampilkan di konsol Amazon S3 atau `GrantScope` yang dikembalikan dalam respons API atau AWS Command Line Interface (AWS CLI) adalah hasil dari menggabungkan jalur lokasi dengan. `Subprefix` Pastikan jalur gabungan ini dipetakan dengan benar ke bucket, prefiks, atau S3 Object yang ingin Anda berikan aksesnya.

> **Note**
>
> - Jika Anda perlu membuat hibah akses yang memberikan akses hanya ke satu objek, Anda harus menentukan bahwa jenis hibah adalah untuk objek. Untuk melakukan ini dalam panggilan API atau perintah CLI, berikan `s3PrefixType` parameter dengan nilai. `Object` Di konsol Amazon S3, saat Anda membuat hibah, setelah memilih lokasi, di bawah Grant Scope, pilih kotak centang Grant scope is an object.
> - Anda tidak dapat membuat pemberian ke bucket jika bucket belum ada. Namun, Anda dapat membuat hibah ke awalan yang belum ada.
> - Untuk jumlah maksimum hibah yang dapat Anda buat di instance S3 Access Grants, lihat. Batasan S3 Access Grants

Anda dapat membuat hibah akses dengan menggunakan konsol Amazon S3 AWS CLI, Amazon S3 REST API, dan. AWS SDKs

Menggunakan konsol S3

Untuk membuat pemberian akses

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

2. Pada panel navigasi di sebelah kiri, pilih Access Grants.
3. Pada halaman S3 Access Grants, pilih Region yang berisi instance S3 Access Grants yang ingin Anda gunakan.

   Jika Anda menggunakan instans S3 Access Grants untuk pertama kalinya, pastikan Anda telah menyelesaikan Langkah 2-daftarkan lokasi dan navigasikan ke Langkah 3 dari wizard Buat instans Access Grants. Jika Anda sudah memiliki instans S3 Access Grants, pilih Lihat detail, lalu dari tab Izin, pilih Buat pemberian.

   a. Di bagian Cakupan pemberian, pilih atau masukkan lokasi yang telah terdaftar.

      Jika Anda memilih lokasi default `s3://`, gunakan kotak Subprefiks untuk mempersempit cakupan pemberian akses. Untuk informasi selengkapnya, lihat Subprefiks. Jika Anda memberikan akses hanya ke objek, pilih Ruang lingkup pemberian adalah sebuah objek.

   b. Di bawah Izin dan akses, pilih tingkat Izin, baik Baca, Tulis, atau keduanya.

      Kemudian pilih jenis Jenis pemberian. Jika Anda telah menambahkan direktori perusahaan Anda ke IAM Identity Center dan mengaitkan instans IAM Identity Center ini dengan instans S3 Access Grants Anda, Anda dapat memilih Identitas direktori dari IAM Identity Center. Jika Anda memilih opsi ini, dapatkan ID pengguna atau grup dari IAM Identity Center dan masukkan di bagian ini.

      Jika Jenis penerima adalah pengguna IAM atau peran, pilih Pengguna utama IAM. Di bawah Jenis pengguna utama IAM, pilih Pengguna atau Peran. Kemudian, di bawah Pengguna utama IAM, pilih dari daftar atau masukkan ID identitas.

   c. Untuk membuat pemberian S3 Access Grants, pilih Selanjutnya atau Buat pemberian.
4. Jika Selanjutnya atau Buat pemberian dinonaktifkan:

   Tidak dapat membuat pemberian

   - Anda mungkin perlu mendaftarkan lokasi terlebih dahulu di dalam instans S3 Access Grants Anda.
   - Anda mungkin tidak memiliki izin `s3:CreateAccessGrant` untuk membuat akses. Menghubungi administrator akun Anda.

Menggunakan AWS CLI

Untuk menginstal AWS CLI, lihat Menginstal AWS CLI di Panduan AWS Command Line Interface Pengguna.

Contoh berikut menunjukkan cara membuat permintaan pemberian akses untuk pengguna utama IAM dan cara membuat permintaan pemberian akses untuk pengguna atau grup direktori perusahaan.

Untuk menggunakan perintah contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri.

> **Note**
>
> Jika Anda membuat pemberian akses yang memberikan akses hanya ke satu objek, sertakan parameter `--s3-prefix-type Object` yang diperlukan.

Example Buat permintaan pemberian akses untuk pengguna utama IAM

```
aws s3control create-access-grant \
--account-id 111122223333 \
--access-grants-location-id a1b2c3d4-5678-90ab-cdef-EXAMPLE22222 \
--access-grants-location-configuration S3SubPrefix=prefixB* \
--permission READ \
--grantee GranteeType=IAM,GranteeIdentifier=arn:aws:iam::123456789012:user/data-
consumer-3
```

Example Membuat respons pemberian akses

```
{"CreatedAt": "2023-05-31T18:41:34.663000+00:00",
    "AccessGrantId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "AccessGrantArn": "arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Grantee": {
        "GranteeType": "IAM",
        "GranteeIdentifier": "arn:aws:iam::111122223333:user/data-consumer-3"
    },
    "AccessGrantsLocationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "AccessGrantsLocationConfiguration": {
        "S3SubPrefix": "prefixB*"
```

```
    },
    "GrantScope": "s3://DOC-BUCKET-EXAMPLE/prefix*",
    "Permission": "READ"
}
```

Membuat permintaan pemberian akses untuk pengguna atau grup direktori

Untuk membuat permintaan pemberian akses untuk pengguna atau grup direktori, Anda harus terlebih dahulu mendapatkan GUID untuk pengguna direktori atau grup dengan menjalankan salah satu perintah berikut.

Example Dapatkan GUID untuk pengguna direktori atau grup

Anda dapat menemukan GUID pengguna IAM Identity Center melalui konsol IAM Identity Center atau dengan menggunakan atau. AWS CLI AWS SDKs Perintah berikut mencantumkan pengguna dalam contoh instans IAM Identity Center yang ditentukan, dengan nama dan pengenal mereka.

```
aws identitystore list-users --identity-store-id d-1a2b3c4d1234
```

Perintah ini mencantumkan grup dalam instans IAM Identity Center yang ditentukan.

```
aws identitystore list-groups --identity-store-id d-1a2b3c4d1234
```

Example Membuat pemberian akses untuk pengguna atau grup direktori

Perintah ini mirip dengan membuat pemberian untuk pengguna atau peran IAM, kecuali jenis penerima pemberian adalah `DIRECTORY_USER` atau `DIRECTORY_GROUP`, dan pengidentifikasi penerima pemberian adalah GUID untuk pengguna direktori atau grup.

```
aws s3control create-access-grant \
--account-id 123456789012 \
--access-grants-location-id default \
--access-grants-location-configuration S3SubPrefix="amzn-s3-demo-bucket/rafael/*" \
--permission READWRITE \
--grantee GranteeType=DIRECTORY_USER,GranteeIdentifier=83d43802-00b1-7054-db02-
f1d683aacba5 \
```

Penggunaan API REST

Untuk informasi tentang support API REST Amazon S3 untuk mengelola pemberian akses, lihat bagian berikut di Referensi API Amazon Storage Service:

- CreateAccessGrant
- DeleteAccessGrant
- GetAccessGrant
- ListAccessGrants

Menggunakan AWS SDKs

Bagian ini memberikan contoh cara membuat hibah akses dengan menggunakan AWS SDKs.

Java

Untuk menggunakan contoh berikut, ganti *`user input placeholders`* dengan informasi Anda sendiri:

> **Note**
>
> Jika Anda membuat pemberian akses yang memberikan akses hanya ke satu objek, sertakan parameter `.s3PrefixType(S3PrefixType.Object)` yang diperlukan.

Example Buat permintaan pemberian akses

```
public void createAccessGrant() {
CreateAccessGrantRequest createRequest = CreateAccessGrantRequest.builder()
.accountId("111122223333")
.accessGrantsLocationId("a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa")
.permission("READ")
.accessGrantsLocationConfiguration(AccessGrantsLocationConfiguration.builder().s3SubPrefix("
.grantee(Grantee.builder().granteeType("IAM").granteeIdentifier("arn:aws:iam::111122223333:u
data-consumer-3").build())
.build();
CreateAccessGrantResponse createResponse =
 s3Control.createAccessGrant(createRequest);
LOGGER.info("CreateAccessGrantResponse: " + createResponse);
}
```

Example Membuat respons pemberian akses

```
CreateAccessGrantResponse(
CreatedAt=2023-06-07T05:20:26.330Z,
```

```
AccessGrantId=a1b2c3d4-5678-90ab-cdef-EXAMPLE33333,
AccessGrantArn=arn:aws:s3:us-east-2:444455556666:access-grants/default/grant/
a1b2c3d4-5678-90ab-cdef-EXAMPLE33333,
Grantee=Grantee(
GranteeType=IAM,
GranteeIdentifier=arn:aws:iam::111122223333:user/data-consumer-3
),
AccessGrantsLocationId=a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa,
AccessGrantsLocationConfiguration=AccessGrantsLocationConfiguration(
S3SubPrefix=prefixB*
),
GrantScope=s3://DOC-BUCKET-EXAMPLE/prefixB,
Permission=READ
)
```

Lihat izin

Anda dapat melihat detail hibah akses di instans Amazon S3 Access Grants dengan menggunakan konsol Amazon S3, AWS Command Line Interface ()AWS CLI, Amazon S3 REST API, dan. AWS SDKs

Menggunakan konsol S3

Untuk melihat detail hibah akses

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Pada panel navigasi di sebelah kiri, pilih Access Grants.
3. Pada halaman S3 Access Grants, pilih Region yang berisi instance S3 Access Grants yang ingin Anda gunakan.
4. Pilih Lihat detail untuk contoh.
5. Pada halaman detail, pilih tab Hibah.
6. Di bagian Hibah, temukan akses hibah yang ingin Anda lihat. Untuk memfilter daftar hibah, gunakan kotak pencarian.

Menggunakan AWS CLI

Untuk menginstal AWS CLI, lihat Menginstal AWS CLI di Panduan AWS Command Line Interface Pengguna.

Untuk menggunakan perintah contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri.

Example — Dapatkan detail hibah akses

```
aws s3control get-access-grant \
--account-id 111122223333 \
--access-grant-id a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

Respons:

```
{
    "CreatedAt": "2023-05-31T18:41:34.663000+00:00",
    "AccessGrantId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "AccessGrantArn": "arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "Grantee": {
        "GranteeType": "IAM",
        "GranteeIdentifier": "arn:aws:iam::111122223333:user/data-consumer-3"
    },
    "Permission": "READ",
    "AccessGrantsLocationId": "12a6710f-5af8-41f5-b035-0bc795bf1a2b",
    "AccessGrantsLocationConfiguration": {
        "S3SubPrefix": "prefixB*"
    },
    "GrantScope": "s3://amzn-s3-demo-bucket/"
}
```

Example — Daftar semua hibah akses dalam instance S3 Access Grants

Anda dapat menggunakan parameter berikut secara opsional untuk membatasi hasil ke awalan S3 atau identitas AWS Identity and Access Management (IAM):

- Subprefiks - `--grant-scope s3://bucket-name/prefix*`
- Identitas IAM — `--grantee-type IAM` dan `--grantee-identifier arn:aws:iam::123456789000:role/accessGrantsConsumerRole`

```
aws s3control list-access-grants \
--account-id 111122223333
```

Respons:

```
{
    "AccessGrantsList": [{"CreatedAt": "2023-06-14T17:54:46.542000+00:00",
            "AccessGrantId": "dd8dd089-b224-4d82-95f6-975b4185bbaa",
            "AccessGrantArn": "arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant/dd8dd089-b224-4d82-95f6-975b4185bbaa",
            "Grantee": {
                "GranteeType": "IAM",
                "GranteeIdentifier": "arn:aws:iam::111122223333:user/data-consumer-3"
            },
            "Permission": "READ",
            "AccessGrantsLocationId": "23514a34-ea2e-4ddf-b425-d0d4bfcarda1",
            "GrantScope": "s3://amzn-s3-demo-bucket/prefixA*"
        },
        {"CreatedAt": "2023-06-24T17:54:46.542000+00:00",
            "AccessGrantId": "ee8ee089-b224-4d72-85f6-975b4185a1b2",
            "AccessGrantArn": "arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant/ee8ee089-b224-4d72-85f6-975b4185a1b2",
            "Grantee": {
                "GranteeType": "IAM",
                "GranteeIdentifier": "arn:aws:iam::111122223333:user/data-consumer-9"
            },
            "Permission": "READ",
            "AccessGrantsLocationId": "12414a34-ea2e-4ddf-b425-d0d4bfcacao0",
            "GrantScope": "s3://amzn-s3-demo-bucket/prefixB*"
        },

    ]
}
```

Penggunaan API REST

Anda dapat menggunakan operasi Amazon S3 API untuk melihat detail pemberian akses dan mencantumkan semua hibah akses dalam instance S3 Access Grants. Untuk informasi tentang dukungan REST API untuk mengelola hibah akses, lihat bagian berikut di Referensi API Amazon Simple Storage Service:

- GetAccessGrant
- ListAccessGrants

Menggunakan AWS SDKs

Bagian ini memberikan contoh cara mendapatkan rincian hibah akses dengan menggunakan AWS SDKs.

Untuk menggunakan contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri.

Java

Example — Dapatkan detail hibah akses

```
public void getAccessGrant() {
GetAccessGrantRequest getRequest = GetAccessGrantRequest.builder()
.accountId("111122223333")
.accessGrantId("a1b2c3d4-5678-90ab-cdef-EXAMPLE22222")
.build();
GetAccessGrantResponse getResponse = s3Control.getAccessGrant(getRequest);
LOGGER.info("GetAccessGrantResponse: " + getResponse);
}
```

Respons:

```
GetAccessGrantResponse(
CreatedAt=2023-06-07T05:20:26.330Z,
AccessGrantId=a1b2c3d4-5678-90ab-cdef-EXAMPLE22222,
AccessGrantArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant-fd3a5086-42f7-4b34-9fad-472e2942c70e,
Grantee=Grantee(
GranteeType=IAM,
GranteeIdentifier=arn:aws:iam::111122223333:user/data-consumer-3
),
Permission=READ,
AccessGrantsLocationId=12a6710f-5af8-41f5-b035-0bc795bf1a2b,
AccessGrantsLocationConfiguration=AccessGrantsLocationConfiguration(
S3SubPrefix=prefixB*
),
GrantScope=s3://amzn-s3-demo-bucket/
)
```

Example — Daftar semua hibah akses dalam instance S3 Access Grants

Anda dapat menggunakan parameter ini secara opsional untuk membatasi hasil ke prefiks S3, atau identitas IAM:

- Lingkup - `GrantScope=s3://bucket-name/prefix*`
- Penerima hibah — dan `GranteeType=IAM GranteeIdentifier= arn:aws:iam::111122223333:role/accessGrantsConsumerRole`

```
public void listAccessGrants() {
ListAccessGrantsRequest listRequest = ListAccessGrantsRequest.builder()
.accountId("111122223333")
.build();
ListAccessGrantsResponse listResponse = s3Control.listAccessGrants(listRequest);
LOGGER.info("ListAccessGrantsResponse: " + listResponse);
}
```

Respons:

```
ListAccessGrantsResponse(
AccessGrantsList=[
ListAccessGrantEntry(
CreatedAt=2023-06-14T17:54:46.540z,
AccessGrantId=dd8dd089-b224-4d82-95f6-975b4185bbaa,
AccessGrantArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant/dd8dd089-b224-4d82-95f6-975b4185bbaa,
Grantee=Grantee(
GranteeType=IAM, GranteeIdentifier= arn:aws:iam::111122223333:user/data-consumer-3
),
Permission=READ,
AccessGrantsLocationId=23514a34-ea2e-4ddf-b425-d0d4bfcarda1,
GrantScope=s3://amzn-s3-demo-bucket/prefixA
),
ListAccessGrantEntry(
CreatedAt=2023-06-24T17:54:46.540Z,
AccessGrantId=ee8ee089-b224-4d72-85f6-975b4185a1b2,
AccessGrantArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant/ee8ee089-b224-4d72-85f6-975b4185a1b2,
Grantee=Grantee(
GranteeType=IAM, GranteeIdentifier= arn:aws:iam::111122223333:user/data-consumer-9
```

```
),
Permission=READ,
AccessGrantsLocationId=12414a34-ea2e-4ddf-b425-d0d4bfcacao0,
GrantScope=s3://amzn-s3-demo-bucket/prefixB*
)
]
)
```

Hapus izin

Anda dapat menghapus hibah akses dari instans Amazon S3 Access Grants. Anda tidak dapat membatalkan penghapusan hibah akses. Setelah Anda menghapus hibah akses, penerima hibah tidak akan lagi memiliki akses ke data Amazon S3 Anda.

Anda dapat menghapus hibah akses dengan menggunakan konsol Amazon S3, AWS Command Line Interface (AWS CLI), Amazon S3 REST API, dan. AWS SDKs

Menggunakan konsol S3

Untuk menghapus hibah akses

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Pada panel navigasi di sebelah kiri, pilih Access Grants.
3. Pada halaman S3 Access Grants, pilih Region yang berisi instance S3 Access Grants yang ingin Anda gunakan.
4. Pilih Lihat detail untuk contoh.
5. Pada halaman detail, pilih tab Hibah.
6. Cari hibah yang ingin Anda hapus. Saat Anda menemukan hibah, pilih tombol radio di sebelahnya.
7. Pilih Hapus. Kotak dialog muncul dengan peringatan bahwa tindakan Anda tidak dapat dibatalkan. Pilih Hapus lagi untuk menghapus hibah.

Menggunakan AWS CLI

Untuk menginstal AWS CLI, lihat Menginstal AWS CLI di Panduan AWS Command Line Interface Pengguna.

Untuk menggunakan perintah contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri.

Example — Hapus hibah akses

```
aws s3control delete-access-grant \
--account-id 111122223333 \
--access-grant-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

// No response body
```

Penggunaan API REST

Untuk informasi tentang dukungan Amazon S3 REST API untuk mengelola hibah akses, lihat DeleteAccessGrantdi Referensi API Amazon Simple Storage Service.

Menggunakan AWS SDKs

Bagian ini memberikan contoh cara menghapus hibah akses dengan menggunakan AWS SDKs. Untuk menggunakan contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri.

Java

Example — Hapus hibah akses

```
public void deleteAccessGrant() {
DeleteAccessGrantRequest deleteRequest = DeleteAccessGrantRequest.builder()
.accountId("111122223333")
.accessGrantId("a1b2c3d4-5678-90ab-cdef-EXAMPLE11111")
.build();
DeleteAccessGrantResponse deleteResponse =
 s3Control.deleteAccessGrant(deleteRequest);
LOGGER.info("DeleteAccessGrantResponse: " + deleteResponse);
}
```

Respons:

```
DeleteAccessGrantResponse()
```

Mendapatkan data S3 menggunakan hibah akses

Penerima hibah yang telah diberikan akses ke data S3 melalui S3 Access Grants harus meminta kredensil sementara dari S3 Access Grants, yang mereka gunakan untuk mengakses data S3. Untuk informasi selengkapnya, lihat Minta akses ke data Amazon S3 melalui S3 Access Grants. Penerima hibah kemudian menggunakan kredensil sementara untuk melakukan tindakan S3 yang diijinkan pada data S3. Untuk informasi selengkapnya, lihat Mengakses data S3 menggunakan kredensil yang dijual oleh S3 Access Grants. Penerima hibah secara opsional dapat meminta daftar hibah akses mereka untuk Akun AWS sebelum meminta kredensil ini. Untuk informasi selengkapnya, lihat Daftar hibah akses penelepon.

Topik

- Minta akses ke data Amazon S3 melalui S3 Access Grants
- Mengakses data S3 menggunakan kredensil yang dijual oleh S3 Access Grants
- Daftar hibah akses penelepon

Minta akses ke data Amazon S3 melalui S3 Access Grants

Setelah Anda membuat hibah akses menggunakan S3 Access Grants, penerima hibah dapat meminta kredensyal untuk mengakses data S3 yang diberi akses kepada mereka. Penerima hibah dapat berupa kepala sekolah AWS Identity and Access Management (IAM), identitas direktori perusahaan Anda, atau aplikasi resmi.

Aplikasi atau Layanan AWS dapat menggunakan operasi S3 Access Grants `GetDataAccess` API untuk meminta S3 Access Grants untuk akses ke data S3 Anda atas nama penerima hibah. `GetDataAccess`pertama memverifikasi bahwa Anda telah memberikan akses identitas ini ke data. Kemudian, S3 Access Grants menggunakan operasi AssumeRoleAPI untuk mendapatkan token kredenal sementara dan menjualnya ke pemohon. Token kredensial sementara ini adalah token AWS Security Token Service (AWS STS).

Permintaan `GetDataAccess` harus menyertakan parameter `target`, yang menentukan ruang lingkup data S3 yang berlaku untuk kredensial sementara. `target`Ruang lingkup ini dapat sama dengan ruang lingkup hibah atau bagian dari ruang lingkup itu, tetapi `target` ruang lingkup harus dalam ruang lingkup hibah yang diberikan kepada penerima hibah. Permintaan juga harus menentukan `permission` parameter untuk menunjukkan tingkat izin untuk kredensial sementara, apakah `READ`, `WRITE`, atau `READWRITE`.

Keistimewaan

Pemohon dapat menentukan tingkat hak istimewa token sementara dalam permintaan kredensialnya. Dengan menggunakan parameter `privilege`, pemohon dapat mengurangi atau meningkatkan ruang lingkup akses kredensial sementara, dalam batas-batas ruang lingkup pemberian. Nilai default `privilege` parameter adalah`Default`, yang berarti bahwa cakupan target dari kredensi yang dikembalikan adalah cakupan hibah asli. Nilai lain yang mungkin untuk `privilege` adalah `Minimal`. Jika ruang lingkup `target` dikurangi dari ruang lingkup pemberian asli, maka kredensial sementara dihilangkan cakupannya agar sesuai dengan ruang lingkup `target`, selama ruang lingkup berada dalam ruang lingkup `target` pemberian.

Tabel berikut merinci efek parameter `privilege` pada dua pemberian. Satu pemberian memiliki ruang lingkup `S3://`*`amzn-s3-demo-bucket1`*`/bob/*`, yang mencakup seluruh `bob/` prefiks dalam bucket *`amzn-s3-demo-bucket1`*. Hibah lainnya memiliki ruang lingkup `S3://`*`amzn-s3-demo-bucket1`*`/bob/reports/*`, yang hanya mencakup `bob/reports/` prefiks di dalam bucket *`amzn-s3-demo-bucket1`*.

Lingkup pemberian	Lingkup yang diminta	Keistimewaan	Lingkup yang dikembalikan	Efek
`S3://`*`amzn-s3-demo-bucket1`*`/bob/*`	*`amzn-s3-demo-bucket1`*`/bob/*`	`Default`	*`amzn-s3-demo-bucket1`*`/bob/*`	Pemohon memiliki akses ke semua objek yang memiliki nama kunci yang dimulai dengan prefiks *`bob/`* di dalam bucket *`amzn-s3-demo-bucket1`* .
`S3://`*`amzn-s3-demo-bucket1`*`/bob/*`	*`amzn-s3-demo-bucket1`*`/bob/`	`Minimal`	*`amzn-s3-demo-bucket1`*`/bob/`	Tanpa karakter wild card* setelah nama prefiksbob/, pemohon hanya memiliki akses ke objek yang disebutkan bob/ dalam bucket *`amzn-s3-demo-bucket1`* . Tidak umum memiliki objek seperti itu. Pemohon tidak memiliki

Lingkup pemberian	Lingkup yang diminta	Keistimewaan	Lingkup yang dikembalikan	Efek
				akses ke objek lain, termasuk yang memiliki nama kunci yang dimulai dengan prefiks bob/.
S3://*amzn-s3-demo-bucket1* /bob/*	*amzn-s3-demo-bucket1* /bob/images/*	Minimal	*amzn-s3-demo-bucket1* /bob/images/*	Pemohon memiliki akses ke semua objek yang memiliki nama kunci yang dimulai dengan prefiks *bob/images/** di dalam bucket *amzn-s3-demo-bucket1* .
S3://*amzn-s3-demo-bucket1* /bob/reports/*	*amzn-s3-demo-bucket1* /bob/reports/*	Default	*amzn-s3-demo-bucket1* /bob/reports/*	Pemohon memiliki akses ke semua objek yang memiliki nama kunci yang dimulai dengan prefiks bob/reports di dalam bucket *amzn-s3-demo-bucket1* , yang merupakan cakupan pemberian yang cocok.
S3://*amzn-s3-demo-bucket1* /bob/reports/*	*amzn-s3-demo-bucket1* /bob/reports/file.txt	Minimal	*amzn-s3-demo-bucket1* /bob/reports/file.txt	Pemohon hanya memiliki akses ke objek dengan nama kunci bob/reports/file.txt di dalam bucket *amzn-s3-demo-bucket1* . Pemohon tidak memiliki akses ke objek lain.

Identitas direktori

`GetDataAccess`mempertimbangkan semua identitas yang terlibat dalam permintaan saat mencocokkan hibah yang sesuai. Untuk identitas direktori perusahaan, `GetDataAccess` juga mengembalikan hibah identitas IAM yang digunakan untuk sesi sadar identitas. Untuk informasi selengkapnya tentang sesi sadar identitas, lihat Memberikan izin untuk menggunakan sesi konsol sadar identitas di Panduan Pengguna.AWS Identity and Access Management `GetDataAccess`menghasilkan kredensi yang membatasi ruang lingkup untuk hibah yang paling ketat, seperti yang ditunjukkan pada tabel berikut:

Ruang lingkup hibah untuk identitas IAM	Ruang lingkup hibah untuk identitas direktori	Lingkup yang diminta	Lingkup yang dikembalikan	Keistimewaan	Efek
`S3://amzn-s3-demo-bucket1/bob/*`	`amzn-s3-demo-bucket1/bob/images/*`	`S3://amzn-s3-demo-bucket1/bob/images/image1.jpeg`	`S3://amzn-s3-demo-bucket1/bob/images/*`	`Default`	Pemohon memiliki akses ke semua objek yang memiliki nama kunci yang dimulai dengan awalan bob/ sebagai bagian dari hibah untuk peran IAM tetapi dibatasi pada awalan bob/images/ sebagai bagian dari hibah untuk identitas direktori. Baik peran IAM dan identitas direktori menyediakan akses ke ruang lingkup yang diminta, yaitu`bob/images/image1.jpeg` , tetapi identitas direktori memiliki hibah yang

Ruang lingkup hibah untuk identitas IAM	Ruang lingkup hibah untuk identitas direktori	Lingkup yang diminta	Lingkup yang dikembalikan	Keistimewaan	Efek
					lebih ketat. Jadi, ruang lingkup yang dikembalikan dibatasi pada hibah yang lebih ketat untuk identitas direktori.
S3://*amzn-s3-demo-bucket1* / bob/*	*amzn-s3-demo-bucket1* / bob/images/*	S3://*amzn-s3-demo-bucket1* / bob/images/image1.jpeg	S3://*amzn-s3-demo-bucket1* / bob/images/image1.jpeg	Minimal	Karena Privilege diatur keMinimal, meskipun identitas memiliki akses ke cakupan yang lebih besar, hanya lingkup yang diminta dikembalikanbob/images/image1.jpeg .

Ruang lingkup hibah untuk identitas IAM	Ruang lingkup hibah untuk identitas direktori	Lingkup yang diminta	Lingkup yang dikembalikan	Keistimewaan	Efek
S3://*amzn-s3-demo-bucket1* / bob/images/*	*amzn-s3-demo-bucket1* / bob/*	S3://*amzn-s3-demo-bucket1* / bob/images/image1.jpeg	S3://*amzn-s3-demo-bucket1* / bob/images/*	Default	Pemohon memiliki akses ke semua objek yang memiliki nama kunci yang dimulai dengan awalan bob/ sebagai bagian dari hibah untuk identitas direktori tetapi dibatasi pada awalan bob/images/ sebagai bagian dari hibah untuk peran IAM. Baik peran IAM dan identitas direktori menyediakan akses ke ruang lingkup yang diminta, yaitubob/images/image1.jpeg , tetapi peran IAM memiliki hibah yang lebih ketat. Jadi, ruang lingkup yang dikembalikan dibatasi pada hibah yang lebih ketat untuk peran IAM.

Ruang lingkup hibah untuk identitas IAM	Ruang lingkup hibah untuk identitas direktori	Lingkup yang diminta	Lingkup yang dikembalikan	Keistimewaan	Efek
S3://*amzn-s3-demo-bucket1* / bob/images/*	*amzn-s3-demo-bucket1* / bob/*	S3://*amzn-s3-demo-bucket1* / bob/images/image1.jpeg	S3://*amzn-s3-demo-bucket1* / bob/images/image1.jpeg	Minimal	Karena Privilege diatur keMinimal, meskipun identitas memiliki akses ke cakupan yang lebih besar, hanya lingkup yang diminta dikembalikanbob/images/image1.jpeg .

Durasi

Parameter durationSeconds menetapkan durasi kredensial sementara, dalam hitungan detik. Nilai default adalah 3600 detik (1 jam), tetapi pemohon (penerima pemberian) dapat menentukan rentang dari 900 detik (15 menit) hingga 43200 detik (12 jam). Jika penerima pemberian meminta nilai yang lebih tinggi dari maksimum ini, permintaan gagal.

> **Note**
>
> Dalam permintaan Anda untuk token sementara, jika lokasi adalah objek, tetapkan nilai parameter targetType dalam permintaan Object Anda. Parameter ini diperlukan hanya jika lokasi adalah objek dan tingkat hak istimewa adalah Minimal. Jika lokasi adalah bucket atau prefiks, Anda tidak perlu menentukan parameter ini.

Contoh

Anda dapat meminta kredensil sementara dengan menggunakan AWS Command Line Interface (AWS CLI), Amazon S3 REST API, dan file. AWS SDKs Lihat contoh-contoh ini.

Untuk informasi tambahan, lihat GetDataAccessdi Referensi API Amazon Simple Storage Service.

Menggunakan AWS CLI

Untuk menginstal AWS CLI, lihat Menginstal AWS CLI di Panduan AWS Command Line Interface Pengguna.

Untuk menggunakan perintah contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri.

Example Meminta kredensial sementara

Permintaan:

```
aws s3control get-data-access \
--account-id 111122223333 \
--target s3://amzn-s3-demo-bucket/prefixA* \
--permission READ \
--privilege Default \
--region us-east-2
```

Respons:

```
{
"Credentials": {
"AccessKeyId": "Example-key-id",
"SecretAccessKey": "Example-access-key",
"SessionToken": "Example-session-token",
"Expiration": "2023-06-14T18:56:45+00:00"},
"MatchedGrantTarget": "s3://amzn-s3-demo-bucket/prefixA**",
"Grantee": {
    "GranteeType": "IAM",
    "GranteeIdentifier": "arn:aws:iam::111122223333:role/role-name"
 }
}
```

Penggunaan API REST

Untuk informasi tentang dukungan Amazon S3 REST API untuk meminta kredensil sementara dari S3 Access Grants, lihat GetDataAccessdi Referensi API Amazon Simple Storage Service.

Menggunakan AWS SDKs

Bagian ini memberikan contoh bagaimana penerima hibah meminta kredensil sementara dari S3 Access Grants dengan menggunakan. AWS SDKs

Java

Contoh kode berikut mengembalikan kredensial sementara yang digunakan penerima pemberian untuk mengakses data S3 Anda. Untuk menggunakan contoh kode ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

Example Dapatkan kredensial sementara

Permintaan:

```
public void getDataAccess() {
GetDataAccessRequest getDataAccessRequest = GetDataAccessRequest.builder()
.accountId("111122223333")
.permission(Permission.READ)
.privilege(Privilege.MINIMAL)
.target("s3://amzn-s3-demo-bucket/prefixA*")
.build();
GetDataAccessResponse getDataAccessResponse =
 s3Control.getDataAccess(getDataAccessRequest);
LOGGER.info("GetDataAccessResponse: " + getDataAccessResponse);
}
```

Respons:

```
GetDataAccessResponse(
Credentials=Credentials(
AccessKeyId="Example-access-key-id",
SecretAccessKey="Example-secret-access-key",
SessionToken="Example-session-token",
Expiration=2023-06-07T06:55:24Z
))
```

Mengakses data S3 menggunakan kredensil yang dijual oleh S3 Access Grants

Setelah penerima hibah memperoleh kredensyal sementara melalui hibah akses mereka, mereka dapat menggunakan kredensyal sementara ini untuk memanggil operasi Amazon S3 API untuk mengakses data Anda.

Penerima hibah dapat mengakses data S3 dengan menggunakan AWS Command Line Interface (AWS CLI), the AWS SDKs, dan Amazon S3 REST API. Selain itu, Anda dapat menggunakan plugin AWS Python dan Java untuk memanggil S3 Access Grants

Menggunakan AWS CLI

Setelah penerima hibah memperoleh kredensi sementara mereka dari S3 Access Grants, mereka dapat mengatur profil dengan kredensil ini untuk mengambil data.

Untuk menginstal AWS CLI, lihat Menginstal AWS CLI di Panduan AWS Command Line Interface Pengguna.

Untuk menggunakan perintah contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri.

Example — Mengatur profil

```
aws configure set aws_access_key_id "$accessKey" --profile access-grants-consumer-access-profile
aws configure set aws_secret_access_key "$secretKey" --profile access-grants-consumer-access-profile
aws configure set aws_session_token "$sessionToken" --profile access-grants-consumer-access-profile
```

Untuk menggunakan perintah contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri.

Example — Dapatkan data S3

Penerima hibah dapat menggunakan get-object AWS CLI perintah untuk mengakses data. Penerima hibah juga dapat menggunakan put-object, ls, dan perintah AWS CLI S3 lainnya.

```
aws s3api get-object \
--bucket amzn-s3-demo-bucket1 \
--key myprefix \
```

```
--region us-east-2 \
--profile access-grants-consumer-access-profile
```

Menggunakan AWS SDKs

Bagian ini memberikan contoh bagaimana penerima hibah dapat mengakses data S3 Anda dengan menggunakan. AWS SDKs

Java

Untuk contoh cara mendapatkan data S3 dengan menggunakan kredensil sementara, lihat cara mendapatkan objek dengan menggunakan contoh kode Amazon AWS SDKs S3 dan. AWS SDK for Java 2.x

Tindakan S3 yang didukung di Hibah Akses S3

Penerima hibah dapat menggunakan kredensi sementara yang dijual oleh S3 Access Grants untuk melakukan tindakan S3 pada data S3 yang dapat mereka akses. Berikut ini adalah daftar tindakan S3 yang diijinkan yang dapat dilakukan oleh penerima hibah. Tindakan mana yang diizinkan tergantung pada tingkat izin yang diberikan dalam hibah akses, baik, `READWRITE`, atau`READWRITE`.

> **Note**
>
> Selain izin Amazon S3 yang tercantum di bawah ini, Amazon S3 dapat memanggil izin AWS Key Management Service (AWS KMS) Decrypt () atau izin (`kms:decrypt`)`READ`. AWS KMS GenerateDataKey`kms:generateDataKey``WRITE` Izin ini tidak mengizinkan akses langsung ke AWS KMS kunci.

Tindakan IAM S3	Tindakan & dokumen API	Izin Hibah Akses S3	Sumber daya S3			
`s3:GetObject`	GetObject	`READ`	Objek			
`s3:GetObjectVersion`	GetObject	`READ`	Objek			

Tindakan IAM S3	Tindakan & dokumen API	Izin Hibah Akses S3	Sumber daya S3			
s3:GetObjectAcl	GetObjectAcl	READ	Objek			
s3:GetObjectVersionAcl	GetObjectAcl	READ	Objek			
s3:ListMultipartUploads	ListParts	READ	Objek			
s3:PutObject	PutObject, CreateMultipartUpload, UploadPart, UploadPartCopy, CompleteMultipartUpload	WRITE	Objek			
s3:PutObjectAcl	PutObjectAcl	WRITE	Objek			
s3:PutObjectVersionAcl	PutObjectAcl	WRITE	Objek			
s3:DeleteObject	DeleteObject	WRITE	Objek			
s3:DeleteObjectVersion	DeleteObject	WRITE	Objek			
s3:AbortMultipartUpload	AbortMultipartUpload	WRITE	Objek			

Tindakan IAM S3	Tindakan & dokumen API	Izin Hibah Akses S3	Sumber daya S3			
`s3:ListBucket`	HeadBucket, ListObjectsV2, ListObjects	`READ`	Bucket			
`s3:ListBucketVersions`	ListObjectVersions	`READ`	Bucket			
`s3:ListBucketMultipartUploads`	ListMultipartUploads	`READ`	Bucket			

Daftar hibah akses penelepon

Pemilik data S3 dapat menggunakan Hibah Akses S3 untuk membuat hibah akses untuk identitas AWS Identity and Access Management (IAM) atau untuk identitas direktori perusahaan. AWS IAM Identity Center Identifikasi IAM dan identitas direktori Pusat Identitas IAM pada gilirannya dapat menggunakan `ListCallerAccessGrants` API untuk mencantumkan semua bucket, awalan, dan objek Amazon S3 yang dapat mereka akses, seperti yang ditentukan oleh Hibah Akses S3 mereka. Gunakan API ini untuk menemukan semua data S3 yang dapat diakses oleh IAM atau identitas direktori melalui S3 Access Grants.

Anda dapat menggunakan fitur ini untuk membangun aplikasi yang menampilkan data yang dapat diakses oleh pengguna akhir tertentu. Misalnya, AWS Storage Browser untuk S3, komponen UI open source yang digunakan pelanggan untuk mengakses bucket S3, menggunakan fitur ini untuk menyajikan data yang dapat diakses oleh pengguna akhir di Amazon S3, berdasarkan Hibah Akses S3 mereka. Contoh lain adalah ketika membangun aplikasi untuk browsing, mengunggah, atau men-download data di Amazon S3, Anda dapat menggunakan fitur ini untuk membangun struktur pohon dalam aplikasi Anda yang kemudian dapat dijelajahi oleh pengguna akhir.

> **Note**
>
> Untuk identitas direktori perusahaan, saat mencantumkan hibah akses pemanggil, S3 Access Grants mengembalikan hibah identitas IAM yang digunakan untuk sesi sadar identitas.

> Untuk informasi selengkapnya tentang sesi sadar identitas, lihat Memberikan izin untuk menggunakan sesi konsol sadar identitas di Panduan Pengguna.AWS Identity and Access Management

Penerima hibah apakah identitas IAM, atau identitas direktori perusahaan bisa mendapatkan daftar hibah akses mereka dengan menggunakan AWS Command Line Interface (AWS CLI), Amazon S3 REST API, dan. AWS SDKs

Menggunakan AWS CLI

Untuk menginstal AWS CLI, lihat Menginstal AWS CLI di Panduan AWS Command Line Interface Pengguna.

Untuk menggunakan perintah contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri.

Example Daftar hibah akses penelepon

Permintaan:

```
aws s3control list-caller-access-grants \
--account-id 111122223333 \
--region us-east-2
--max-results 5
```

Respons:

```
{
 "NextToken": "6J9S...",
 "CallerAccessGrantsList": [
  {
   "Permission": "READWRITE",
   "GrantScope": "s3://amzn-s3-demo-bucket/prefix1/sub-prefix1/*",
   "ApplicationArn": "NA"
  },
  {
   "Permission": "READWRITE",
   "GrantScope": "s3://amzn-s3-demo-bucket/prefix1/sub-prefix2/*",
```

```
    "ApplicationArn": "ALL"
   },
   {
    "Permission": "READWRITE",
    "GrantScope": "s3://amzn-s3-demo-bucket/prefix1/sub-prefix3/*",
    "ApplicationArn": "arn:aws:sso::111122223333:application/ssoins-
ssoins-1234567890abcdef/apl-abcd1234a1b2c3d"
   }
 ]
}
```

Example Buat daftar hibah akses penelepon untuk ember

Anda dapat mempersempit ruang lingkup hasil menggunakan `grantscope` parameter.

Permintaan:

```
aws s3control list-caller-access-grants \
--account-id 111122223333 \
--region us-east-2
--grant-scope "s3://amzn-s3-demo-bucket""
--max-results 1000
```

Respons:

```
{
 "NextToken": "6J9S...",
 "CallerAccessGrantsList": [
  {
   "Permission": "READ",
   "GrantScope": "s3://amzn-s3-demo-bucket*",
   "ApplicationArn": "ALL"
  },
  {
   "Permission": "READ",
   "GrantScope": "s3://amzn-s3-demo-bucket/prefix1/*",
   "ApplicationArn": "arn:aws:sso::111122223333:application/ssoins-
ssoins-1234567890abcdef/apl-abcd1234a1b2c3d"
  }
 ]
}
```

Penggunaan API REST

Untuk informasi tentang dukungan Amazon S3 REST API untuk mendapatkan daftar hibah akses pemanggil API, lihat di ListCallerAccessGrantsReferensi API Layanan Penyimpanan Sederhana Amazon.

Menggunakan AWS SDKs

Bagian ini memberikan contoh bagaimana penerima hibah meminta kredensil sementara dari S3 Access Grants dengan menggunakan. AWS SDKs

Java

Contoh kode berikut mengembalikan hibah akses pemanggil API ke data S3 tertentu. Akun AWS Untuk menggunakan contoh kode ini, ganti *user input placeholders* dengan informasi Anda sendiri.

Example Daftar hibah akses penelepon

Permintaan:

```
Public void ListCallerAccessGrants() {
 ListCallerAccessGrantsRequest listRequest = ListCallerAccessGrantsRequest.builder()
    .withMaxResults(1000)
    .withGrantScope("s3://")
    .accountId("111122223333");
 ListCallerAccessGrantsResponse listResponse =
 s3control.listCallerAccessGrants(listRequest);
 LOGGER.info("ListCallerAccessGrantsResponse: " + listResponse);
 }
```

Respons:

```
ListCallerAccessGrantsResponse(
CallerAccessGrantsList=[
 ListCallerAccessGrantsEntry(
  S3Prefix=s3://amzn-s3-demo-bucket/prefix1/,
  Permission=READ,
  ApplicationArn=ALL
 )
])
```

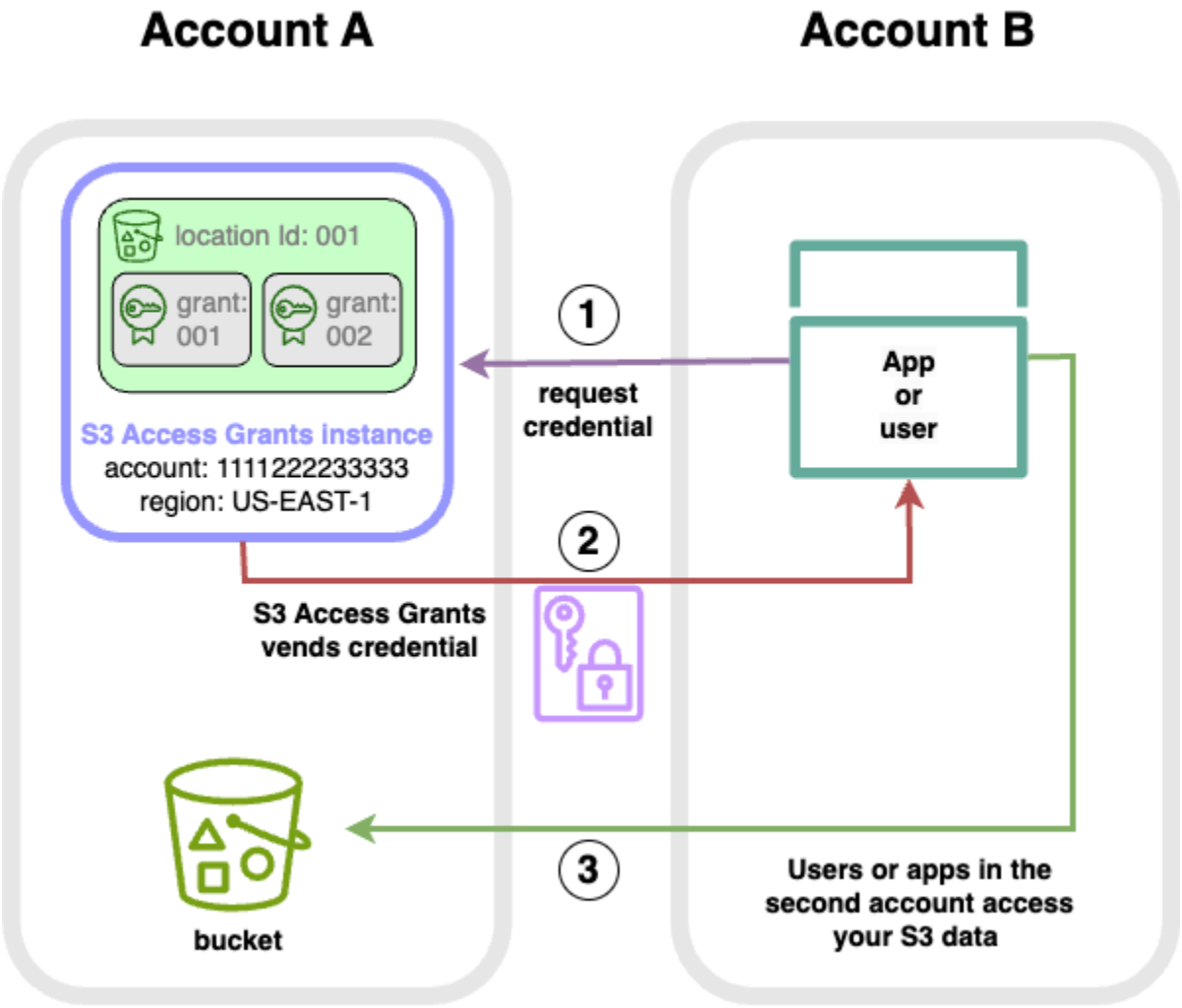
Akses S3 memberikan akses lintas akun

Dengan S3 Access Grants, Anda dapat memberikan akses data Amazon S3 ke hal-hal berikut:

- AWS Identity and Access Management (IAM) identitas dalam akun Anda
- Identitas IAM di akun lain AWS
- Pengguna direktori atau grup dalam AWS IAM Identity Center instans Anda

Pertama, konfigurasikan akses lintas akun untuk akun lain. Ini termasuk memberikan akses ke instans S3 Access Grants Anda dengan menggunakan kebijakan sumber daya. Kemudian, berikan akses ke data S3 Anda (bucket, awalan, atau objek) dengan menggunakan hibah.

Setelah Anda mengonfigurasi akses lintas akun, akun lain dapat meminta kredenal akses sementara ke data Amazon S3 Anda dari S3 Access Grants. Gambar berikut menunjukkan alur pengguna untuk akses S3 lintas akun melalui S3 Access Grants:



1. Pengguna atau aplikasi di akun kedua (B) meminta kredensil dari instans S3 Access Grants di akun Anda (A), tempat data Amazon S3 disimpan. Untuk informasi selengkapnya, lihat Minta akses ke data Amazon S3 melalui S3 Access Grants.
2. Instans S3 Access Grants di akun Anda (A) mengembalikan kredensyal sementara jika ada hibah yang memberikan akses akun kedua ke data Amazon S3 Anda. Untuk informasi lebih lanjut tentang hibah akses, lihatBekerja dengan hibah di S3 Access Grants.
3. Pengguna atau aplikasi di akun kedua (B) menggunakan kredensil yang dijual oleh S3 Access Grants untuk mengakses data S3 di akun Anda (A).

Mengkonfigurasi Akses S3 Memberikan akses lintas akun

Untuk memberikan akses S3 lintas akun melalui S3 Access Grants, ikuti langkah-langkah berikut:

- Langkah 1: Konfigurasikan instance S3 Access Grants di akun Anda, misalnya, ID akun111122223333, tempat data S3 disimpan.
- Langkah 2: Konfigurasikan kebijakan sumber daya untuk instans S3 Access Grants di akun Anda 111122223333 untuk memberikan akses ke akun kedua, misalnya, ID akun. 444455556666
- Langkah 3: Konfigurasikan izin IAM untuk Kepala Sekolah IAM di akun kedua 444455556666 untuk meminta kredensil dari instans S3 Access Grants di akun Anda. 111122223333
- Langkah 4: Buat hibah di akun Anda 111122223333 yang memberikan Kepala Sekolah IAM di akun kedua 444455556666 akses ke beberapa data S3 di akun Anda. 111122223333

Langkah 1: Konfigurasikan instance S3 Access Grants di akun Anda

Pertama, Anda harus memiliki instans S3 Access Grants di akun Anda 111122223333 untuk mengelola akses ke data Amazon S3 Anda. Anda harus membuat instance S3 Access Grants di setiap Wilayah AWS tempat data S3 yang ingin Anda bagikan disimpan. Jika Anda berbagi data di lebih dari satu Wilayah AWS, ulangi setiap langkah konfigurasi ini untuk masing-masing Wilayah AWS. Jika Anda sudah memiliki instance S3 Access Grants di Wilayah AWS tempat penyimpanan data S3 Anda, lanjutkan ke langkah berikutnya. Jika Anda belum mengonfigurasi instance S3 Access Grants, lihat Bekerja dengan instans S3 Access Grants untuk menyelesaikan langkah ini.

Langkah 2: Konfigurasikan kebijakan sumber daya untuk instans S3 Access Grants Anda untuk memberikan akses lintas akun

Setelah Anda membuat instans S3 Access Grants di akun Anda 111122223333 untuk akses lintas akun, konfigurasikan kebijakan berbasis sumber daya untuk instans S3 Access Grants di akun Anda untuk memberikan akses lintas akun. 111122223333 Instans S3 Access Grants sendiri mendukung kebijakan berbasis sumber daya. Dengan kebijakan berbasis sumber daya yang benar, Anda dapat memberikan akses untuk pengguna AWS Identity and Access Management (IAM) atau peran dari orang lain Akun AWS ke instans Hibah Akses S3 Anda. Akses lintas akun hanya memberikan izin (tindakan) ini:

- `s3:GetAccessGrantsInstanceForPrefix`— pengguna, peran, atau aplikasi dapat mengambil instance S3 Access Grants yang berisi awalan tertentu.
- `s3:ListAccessGrants`
- `s3:ListAccessLocations`
- `s3:ListCallerAccessGrants`

- `s3:GetDataAccess`— pengguna, peran, atau aplikasi dapat meminta kredensil sementara berdasarkan akses yang diberikan kepada Anda melalui Hibah Akses S3. Gunakan kredensial ini untuk mengakses data S3 yang telah diberikan akses kepada Anda.

Anda dapat memilih izin mana yang akan disertakan dalam kebijakan sumber daya. Kebijakan sumber daya pada instans S3 Access Grants ini adalah kebijakan berbasis sumber daya normal dan mendukung semua yang didukung oleh bahasa kebijakan IAM. Dalam kebijakan yang sama, Anda dapat memberikan akses ke identitas IAM tertentu di akun Anda`111122223333`, misalnya, dengan menggunakan `aws:PrincipalArn` kondisi, tetapi Anda tidak harus melakukannya dengan Hibah Akses S3. Sebagai gantinya, dalam instance S3 Access Grants Anda, Anda dapat membuat hibah untuk identitas IAM individual dari akun Anda, serta untuk akun lainnya. Dengan mengelola setiap hibah akses melalui S3 Access Grants, Anda dapat menskalakan izin Anda.

Jika Anda sudah menggunakan AWS Resource Access Manager(AWS RAM), Anda dapat menggunakannya untuk membagikan `s3:AccessGrants`sumber daya Anda dengan akun lain atau di dalam organisasi Anda. Lihat Bekerja dengan AWS sumber daya bersama untuk informasi selengkapnya. Jika tidak digunakan AWS RAM, Anda juga dapat menambahkan kebijakan sumber daya dengan menggunakan operasi S3 Access Grants API atau AWS Command Line Interface ()AWS CLI.

Menggunakan konsol S3

Kami menyarankan Anda menggunakan AWS Resource Access Manager (AWS RAM) Konsol untuk membagikan `s3:AccessGrants` sumber daya Anda dengan akun lain atau di dalam organisasi Anda. Untuk membagikan akun lintas akun S3 Access Grants, lakukan hal berikut:

Untuk mengonfigurasi kebijakan sumber daya instans S3 Access Grants:

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Pilih Wilayah AWS dari Wilayah AWS pemilih.
3. Dari panel navigasi kiri, pilih Access Grants.
4. Pada halaman instance Access Grants, di bagian Instans di akun ini, pilih Bagikan instance. Ini akan mengarahkan Anda ke AWS RAM Konsol.
5. Pilih Buat berbagi sumber daya.
6. Ikuti AWS RAM langkah-langkah untuk membuat pembagian sumber daya. Untuk informasi selengkapnya, lihat Membuat berbagi sumber daya di AWS RAM.

Menggunakan AWS CLI

Untuk menginstal AWS CLI, lihat Menginstal AWS CLI di Panduan AWS Command Line Interface Pengguna.

Anda dapat menambahkan kebijakan sumber daya dengan menggunakan perintah `put-access-grants-instance-resource-policy` CLI.

Jika Anda ingin memberikan akses lintas akun untuk instans S3 Access Grants ada di akun Anda `111122223333` ke akun kedua`444455556666`, kebijakan sumber daya untuk instance S3 Access Grants di akun Anda `111122223333` harus memberikan `444455556666` izin akun kedua untuk melakukan tindakan berikut:

- `s3:ListAccessGrants`
- `s3:ListAccessGrantsLocations`
- `s3:GetDataAccess`
- `s3:GetAccessGrantsInstanceForPrefix`

Dalam kebijakan sumber daya instans S3 Access Grants, tentukan ARN instance S3 Access Grants Anda sebagai`Resource`, dan akun kedua sebagai. `444455556666` `Principal` Untuk menggunakan contoh berikut, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
{
"Version": "2012-10-17",
"Statement": [
{
 "Effect": "Allow",
 "Principal": {
 "AWS": "444455556666"
},
 "Action": [
  "s3:ListAccessGrants",
  "s3:ListAccessGrantsLocations",
  "s3:GetDataAccess",
  "s3:GetAccessGrantsInstanceForPrefix"
 ],
 "Resource": "arn:aws:s3:us-east-2:111122223333:access-grants/default"
} ]
}
```

Untuk menambahkan atau memperbarui kebijakan sumber daya instans S3 Access Grants, gunakan perintah berikut. Bila Anda menggunakan perintah contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri.

Example Menambahkan atau memperbarui kebijakan sumber daya instans S3 Access Grants

```
aws s3control put-access-grants-instance-resource-policy \
--account-id 111122223333 \
--policy file://resourcePolicy.json \
--region us-east-2
{
 "Policy": "{\n
   \"Version\": \"2012-10-17\",\n
   \"Statement\": [{\n
  \"Effect\": \"Allow\",\n
  \"Principal\": {\n
    \"AWS\": \"444455556666\"\n
  },\n
  \"Action\": [\n
    \"s3:ListAccessGrants\",\n
    \"s3:ListAccessGrantsLocations\",\n
    \"s3:GetDataAccess\",\n
    \"s3:GetAccessGrantsInstanceForPrefix\",\n
    \"s3:ListCallerAccessGrants"\n
  ],\n
  \"Resource\": \"arn:aws:s3:us-east-2:111122223333:access-grants/default\"\n
    }\n
   ]\n
   }\n",
 "CreatedAt": "2023-06-16T00:07:47.473000+00:00"
}
```

Example Mendapatkan kebijakan sumber daya S3 Access Grants

Anda juga dapat menggunakan CLI untuk mendapatkan atau menghapus kebijakan sumber daya untuk instance S3 Access Grants.

Untuk mendapatkan kebijakan sumber daya S3 Access Grants, gunakan perintah contoh berikut. Untuk menggunakan contoh perintah ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3control get-access-grants-instance-resource-policy \
```

```
--account-id 111122223333 \
--region us-east-2

{
"Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"AWS\":\"arn:aws:iam::111122223333:root\"},\"Action\":
[\"s3:ListAccessGrants\",\"s3:ListAccessGrantsLocations\",\"s3:GetDataAccess\",
\"s3:GetAccessGrantsInstanceForPrefix\",\"s3:ListCallerAccessGrants\"],\"Resource\":
\"arn:aws:
s3:us-east-2:111122223333:access-grants/default\"}]}",
"CreatedAt": "2023-06-16T00:07:47.473000+00:00"
}
```

Example Menghapus kebijakan sumber daya S3 Access Grants

Untuk menghapus kebijakan sumber daya S3 Access Grants, gunakan perintah contoh berikut. Untuk menggunakan contoh perintah ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3control delete-access-grants-instance-resource-policy \
--account-id 111122223333 \
--region us-east-2

// No response body
```

Penggunaan API REST

Anda dapat menambahkan kebijakan sumber daya dengan menggunakan PutAccessGrantsInstanceResourcePolicy API.

Jika Anda ingin memberikan akses lintas akun untuk instans S3 Access Grants ada di akun Anda 111122223333 ke akun kedua444455556666, kebijakan sumber daya untuk instance S3 Access Grants di akun Anda 111122223333 harus memberikan 444455556666 izin akun kedua untuk melakukan tindakan berikut:

- `s3:ListAccessGrants`
- `s3:ListAccessGrantsLocations`
- `s3:GetDataAccess`
- `s3:GetAccessGrantsInstanceForPrefix`

Dalam kebijakan sumber daya instans S3 Access Grants, tentukan ARN instance S3 Access Grants Anda sebagai`Resource`, dan akun kedua sebagai. `444455556666` `Principal` Untuk menggunakan contoh berikut, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
{
"Version": "2012-10-17",
"Statement": [
{
 "Effect": "Allow",
 "Principal": {
 "AWS": "444455556666"
},
 "Action": [
  "s3:ListAccessGrants",
  "s3:ListAccessGrantsLocations",
  "s3:GetDataAccess",
  "s3:GetAccessGrantsInstanceForPrefix"
 ],
 "Resource": "arn:aws:s3:us-east-2:111122223333:access-grants/default"
} ]
}
```

Anda kemudian dapat menggunakan PutAccessGrantsInstanceResourcePolicy API untuk mengonfigurasi kebijakan.

Untuk informasi tentang dukungan REST API untuk memperbarui, mendapatkan, atau menghapus kebijakan sumber daya untuk instance S3 Access Grants, lihat bagian berikut di Referensi API Amazon Simple Storage Service:

- PutAccessGrantsInstanceResourcePolicy
- GetAccessGrantsInstanceResourcePolicy
- DeleteAccessGrantsInstanceResourcePolicy

Menggunakan AWS SDKs

Bagian ini memberi Anda contoh AWS SDK tentang cara mengonfigurasi kebijakan sumber daya S3 Access Grants Anda untuk memberikan akses AWS akun kedua ke beberapa data S3 Anda.

Java

Tambahkan, perbarui, dapatkan, atau hapus kebijakan sumber daya untuk mengelola akses lintas akun ke instans S3 Access Grants Anda.

Example Menambahkan atau memperbarui kebijakan sumber daya instans S3 Access Grants

Jika Anda ingin memberikan akses lintas akun untuk instans S3 Access Grants ada di akun Anda 111122223333 ke akun kedua444455556666, kebijakan sumber daya untuk instance S3 Access Grants di akun Anda 111122223333 harus memberikan 444455556666 izin akun kedua untuk melakukan tindakan berikut:

- `s3:ListAccessGrants`
- `s3:ListAccessGrantsLocations`
- `s3:GetDataAccess`
- `s3:GetAccessGrantsInstanceForPrefix`

Dalam kebijakan sumber daya instans S3 Access Grants, tentukan ARN instance S3 Access Grants Anda sebagai`Resource`, dan akun kedua sebagai. 444455556666 `Principal` Untuk menggunakan contoh berikut, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
{
"Version": "2012-10-17",
"Statement": [
{
 "Effect": "Allow",
 "Principal": {
 "AWS": "444455556666"
},
 "Action": [
  "s3:ListAccessGrants",
  "s3:ListAccessGrantsLocations",
  "s3:GetDataAccess",
  "s3:GetAccessGrantsInstanceForPrefix"
 ],
 "Resource": "arn:aws:s3:us-east-2:111122223333:access-grants/default"
} ]
}
```

Untuk menambah atau memperbarui kebijakan sumber daya instans S3 Access Grants, gunakan contoh kode berikut:

```
public void putAccessGrantsInstanceResourcePolicy() {
 PutAccessGrantsInstanceResourcePolicyRequest putRequest =
 PutAccessGrantsInstanceResourcePolicyRequest.builder()
 .accountId(111122223333)
 .policy(RESOURCE_POLICY)
 .build();
 PutAccessGrantsInstanceResourcePolicyResponse putResponse =
 s3Control.putAccessGrantsInstanceResourcePolicy(putRequest);
 LOGGER.info("PutAccessGrantsInstanceResourcePolicyResponse: " + putResponse);
 }
```

Respons:

```
PutAccessGrantsInstanceResourcePolicyResponse(
 Policy={
 "Version": "2012-10-17",
 "Statement": [{
 "Effect": "Allow",
 "Principal": {
 "AWS": "444455556666"
 },
 "Action": [
 "s3:ListAccessGrants",
 "s3:ListAccessGrantsLocations",
 "s3:GetDataAccess",
 "s3:GetAccessGrantsInstanceForPrefix",
 "s3:ListCallerAccessGrants"
 ],
 "Resource": "arn:aws:s3:us-east-2:111122223333:access-grants/default"
 }]
 }
 )
```

Example Mendapatkan kebijakan sumber daya S3 Access Grants

Untuk mendapatkan kebijakan sumber daya S3 Access Grants, gunakan contoh kode berikut. Untuk menggunakan perintah contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri.

```
public void getAccessGrantsInstanceResourcePolicy() {
 GetAccessGrantsInstanceResourcePolicyRequest getRequest =
 GetAccessGrantsInstanceResourcePolicyRequest.builder()
 .accountId(111122223333)
 .build();
 GetAccessGrantsInstanceResourcePolicyResponse getResponse =
 s3Control.getAccessGrantsInstanceResourcePolicy(getRequest);
 LOGGER.info("GetAccessGrantsInstanceResourcePolicyResponse: " + getResponse);
 }
```

Respons:

```
GetAccessGrantsInstanceResourcePolicyResponse(
 Policy={"Version":"2012-10-17","Statement":[{"Effect":"Allow","Principal":
{"AWS":"arn:aws:iam::444455556666:root"},"Action":
["s3:ListAccessGrants","s3:ListAccessGrantsLocations","s3:GetDataAccess","s3:GetAccessGrants
east-2:111122223333:access-grants/default"}]},
 CreatedAt=2023-06-15T22:54:44.319Z
 )
```

Example Menghapus kebijakan sumber daya S3 Access Grants

Untuk menghapus kebijakan sumber daya S3 Access Grants, gunakan contoh kode berikut. Untuk menggunakan perintah contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri.

```
public void deleteAccessGrantsInstanceResourcePolicy() {
 DeleteAccessGrantsInstanceResourcePolicyRequest deleteRequest =
 DeleteAccessGrantsInstanceResourcePolicyRequest.builder()
 .accountId(111122223333)
 .build();
 DeleteAccessGrantsInstanceResourcePolicyResponse deleteResponse =
 s3Control.putAccessGrantsInstanceResourcePolicy(deleteRequest);
 LOGGER.info("DeleteAccessGrantsInstanceResourcePolicyResponse: " + deleteResponse);
 }
```

Respons:

```
DeleteAccessGrantsInstanceResourcePolicyResponse()
```

Langkah 3: Berikan identitas IAM di izin akun kedua untuk memanggil instans S3 Access Grants di akun Anda

Setelah pemilik data Amazon S3 mengonfigurasi kebijakan lintas akun untuk instans S3 Access Grants di akun`111122223333`, pemilik akun kedua `444455556666` harus membuat kebijakan berbasis identitas untuk pengguna atau peran IAM-nya, dan pemilik harus memberi mereka akses ke instans S3 Access Grants. Dalam kebijakan berbasis identitas, sertakan satu atau beberapa tindakan berikut, tergantung pada apa yang diberikan dalam kebijakan sumber daya instans S3 Access Grants dan izin yang ingin Anda berikan:

- `s3:ListAccessGrants`
- `s3:ListAccessGrantsLocations`
- `s3:GetDataAccess`
- `s3:GetAccessGrantsInstanceForPrefix`
- `s3:ListCallerAccessGrants`

Mengikuti pola akses AWS lintas akun, pengguna IAM atau peran di akun kedua `444455556666` harus secara eksplisit memiliki satu atau lebih izin ini. Misalnya, berikan `s3:GetDataAccess` izin agar pengguna atau peran IAM dapat memanggil instans S3 Access Grants di akun `111122223333` untuk meminta kredensyal.

Untuk menggunakan perintah contoh ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
  "Effect": "Allow",
  "Action": [
   "s3:GetDataAccess",
  ],
   "Resource": "arn:aws:s3:us-east-2:111122223333:access-grants/default"
  }
 ]
}
```

Untuk informasi tentang mengedit kebijakan berbasis identitas IAM, lihat Mengedit kebijakan IAM dalam panduan.AWS Identity and Access Management

Langkah 4: Buat hibah di instance S3 Access Grants dari akun Anda yang memberikan identitas IAM di akun kedua akses ke beberapa data S3 Anda

Untuk langkah konfigurasi terakhir, Anda dapat membuat hibah di instans S3 Access Grants di akun 111122223333 yang memberikan akses ke identitas IAM di akun kedua 444455556666 ke beberapa data S3 di akun Anda. Anda dapat melakukan ini dengan menggunakan Konsol Amazon S3, CLI, API, dan. SDKs Untuk informasi selengkapnya, lihat Membuat pemberian.

Dalam hibah, tentukan AWS ARN identitas IAM dari akun kedua, dan tentukan lokasi mana dalam data S3 Anda (ember, awalan, atau objek) yang Anda berikan akses. Lokasi ini harus sudah terdaftar dengan instans S3 Access Grants Anda. Untuk informasi selengkapnya, lihat Mendaftarkan lokasi. Anda dapat secara opsional menentukan subprefix. Misalnya, jika lokasi yang Anda berikan akses adalah bucket, dan Anda ingin membatasi akses lebih jauh ke objek tertentu di bucket tersebut, maka berikan nama kunci objek di `S3SubPrefix` bidang tersebut. Atau jika Anda ingin membatasi akses ke objek di bucket dengan nama kunci yang dimulai dengan awalan tertentu, seperti`2024-03-research-results/`, maka teruskan`S3SubPrefix=2024-03-research-results/`.

Berikut ini adalah contoh perintah CLI untuk membuat hibah akses untuk identitas di akun kedua. Untuk informasi selengkapnya, lihat Membuat pemberian. Untuk menggunakan perintah contoh ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3control create-access-grant \
--account-id 111122223333 \
--access-grants-location-id default \
--access-grants-location-configuration S3SubPrefix=prefixA* \
--permission READ \
--grantee GranteeType=IAM,GranteeIdentifier=arn:aws:iam::444455556666:role/data-consumer-1
```

Setelah mengonfigurasi akses lintas akun, pengguna atau peran di akun kedua dapat melakukan hal berikut:

- Panggilan `ListAccessGrantsInstances` untuk mencantumkan instance S3 Access Grants yang dibagikan dengannya. AWS RAM Untuk informasi selengkapnya, lihat Dapatkan detail instans S3 Access Grants.

- Meminta kredensi sementara dari S3 Access Grants. Untuk informasi selengkapnya tentang cara membuat permintaan ini, lihatMinta akses ke data Amazon S3 melalui S3 Access Grants.

Menggunakan AWS tag dengan S3 Access Grants

Tag di Amazon S3 Access Grants memiliki karakteristik yang mirip dengan tag objek di Amazon S3. Setiap tag adalah pasangan nilai kunci. Sumber daya di S3 Access Grants yang dapat Anda tag adalah instans, lokasi, dan pemberian S3 Access Grants.

> **Note**
>
> Penandaan di S3 Access Grants menggunakan operasi API yang berbeda dari penandaan objek. S3 Access Grants menggunakan operasi TagResource, UntagResource, dan ListTagsForResourceAPI, di mana sumber daya dapat berupa instance S3 Access Grants, lokasi terdaftar, atau hibah akses.

Mirip dengan tag objek, batasan berikut berlaku:

- Anda dapat menambahkan tag ke sumber daya baru S3 Access Grants saat Anda membuatnya, atau Anda dapat menambahkan tag ke sumber daya yang sudah ada.
- Anda dapat mengaitkan hingga 10 tag dengan sumber daya. Jika beberapa tag terkait dengan sumber daya yang sama, mereka harus memiliki kunci tag unik.
- Kunci tanda dapat terdiri dari hingga 128 karakter Unicode, dan nilai tanda dapat terdiri dari hingga 256 karakter Unicode. Tag diwakili secara internal dalam UTF-16. Dalam UTF-16, karakter mengkonsumsi 1 atau 2 posisi karakter.
- Kunci dan nilainya peka huruf besar/kecil.

Untuk informasi selengkapnya tentang pembatasan tag, lihat Pembatasan tag yang ditentukan pengguna di AWS Billing Panduan Pengguna.

Anda dapat menandai resource di S3 Access Grants dengan menggunakan AWS Command Line Interface (AWS CLI), Amazon S3 REST API, atau file. AWS SDKs

Menggunakan AWS CLI

Untuk menginstal AWS CLI, lihat Menginstal AWS CLI di Panduan AWS Command Line Interface Pengguna.

Anda dapat menandai sumber daya S3 Access Grants saat Anda membuatnya atau setelah Anda membuatnya. Contoh berikut menunjukkan bagaimana Anda menandai atau menghapus tag instans S3 Access Grants. Anda dapat melakukan operasi serupa untuk lokasi terdaftar dan pemberian akses.

Untuk menggunakan perintah contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri.

Example — Buat instance S3 Access Grants dengan tag

```
aws s3control create-access-grants-instance \
 --account-id 111122223333 \
 --profile access-grants-profile \
 --region us-east-2 \
 --tags Key=tagKey1,Value=tagValue1
```

Respons:

```
 {
    "CreatedAt": "2023-10-25T01:09:46.719000+00:00",
    "AccessGrantsInstanceId": "default",
    "AccessGrantsInstanceArn": "arn:aws:s3:us-east-2:111122223333:access-grants/
default"
}
```

Example — Tandai instance S3 Access Grants yang sudah dibuat

```
aws s3control tag-resource \
--account-id 111122223333 \
--resource-arn "arn:aws:s3:us-east-2:111122223333:access-grants/default" \
--profile access-grants-profile \
--region us-east-2 \
--tags Key=tagKey2,Value=tagValue2
```

Example — Daftar tag untuk instance S3 Access Grants

```
aws s3control list-tags-for-resource \
--account-id 111122223333 \
--resource-arn "arn:aws:s3:us-east-2:111122223333:access-grants/default" \
--profile access-grants-profile \
```

```
--region us-east-2
```

Respons:

```
{
    "Tags": [
        {
            "Key": "tagKey1",
            "Value": "tagValue1"
        },
        {
            "Key": "tagKey2",
            "Value": "tagValue2"
        }
    ]
}
```

Example — Hapus tag instance Hibah Akses S3

```
aws s3control untag-resource \
 --account-id 111122223333 \
 --resource-arn "arn:aws:s3:us-east-2:111122223333:access-grants/default" \
 --profile access-grants-profile \
 --region us-east-2 \
 --tag-keys "tagKey2"
```

Penggunaan API REST

Anda dapat menggunakan Amazon S3 API untuk menandai, menghapus tag, atau mencantumkan tag untuk instance S3 Access Grants, lokasi terdaftar, atau hibah akses. Untuk informasi tentang dukungan REST API untuk mengelola tag S3 Access Grants, lihat bagian berikut di Referensi API Amazon Simple Storage Service:

- TagResource
- UntagResource
- ListTagsForResource

Batasan S3 Access Grants

S3 Access Grants memiliki keterbatasan sebagai berikut:

> **Note**
>
> Jika kasus penggunaan Anda melebihi batasan ini, hubungi AWS dukungan untuk meminta batas yang lebih tinggi.

Instans S3 Access Grants

Anda dapat membuat 1 instance Hibah Akses S3 Wilayah AWS per akun. Lihat Membuat instans S3 Access Grants.

Lokasi S3 Access Grants

Anda dapat mendaftarkan 1.000 lokasi S3 Access Grants per instans S3 Access Grants. Lihat Daftarkan lokasi S3 Access Grants.

Pemberian Izin

Anda dapat membuat 100.000 pemberian per instans S3 Access Grants. Lihat Membuat pemberian.

Hibah Akses S3 Wilayah AWS

Hibah Akses S3 saat ini tersedia dalam hal berikut: Wilayah AWS

Wilayah AWS kode	Wilayah AWS nama
`us-east-1`	US East (N. Virginia)
`us-east-2`	AS Timur (Ohio)
`us-west-1`	AS Barat (California Utara)
`us-west-2`	AS Barat (Oregon)
`af-south-1`	Afrika (Cape Town)
`ap-east-1`	Asia Pacific (Hong Kong)
`ap-northeast-1`	Asia Pacific (Tokyo)

Wilayah AWS kode	Wilayah AWS nama
ap-northeast-2	Asia Pasifik (Seoul)
ap-northeast-3	Asia Pasifik (Osaka)
ap-south-1	Asia Pasifik (Mumbai)
ap-south-2	Asia Pasifik (Hyderabad)
ap-southeast-1	Asia Pasifik (Singapura)
ap-southeast-2	Asia Pasifik (Sydney)
ap-southeast-3	Asia Pasifik (Jakarta)
ap-southeast-4	Asia Pasifik (Melbourne)
ca-central-1	Kanada (Pusat)
ca-west-1	Kanada Barat (Calgary)
eu-central-1	Eropa (Frankfurt)
eu-central-2	Eropa (Zürich)
eu-north-1	Eropa (Stockholm)
eu-south-1	Eropa (Milan)
eu-south-2	Eropa (Spanyol)
eu-west-1	Eropa (Irlandia)
eu-west-2	Eropa (London)
eu-west-3	Eropa (Paris)
il-central-1	Israel (Tel Aviv)
me-central-1	Timur Tengah (UEA)

Wilayah AWS kode	Wilayah AWS nama
me-south-1	Timur Tengah (Bahrain)
sa-east-1	Amerika Selatan (Sao Paulo)
us-gov-east-1	AWS GovCloud (AS-Timur)
us-gov-west-1	AWS GovCloud (AS-Barat)

Integrasi S3 Access Grants

Hibah Akses S3 dapat digunakan dengan AWS layanan dan fitur berikut. Halaman ini akan diperbarui saat integrasi baru tersedia.

> **Tip**
>
> AWS Workshop untuk S3 Access Grants ini memandu Anda melalui penggunaan S3 Access Grants dengan pengguna AWS Identity and Access Management (IAM), pengguna IAM Identity Center, Amazon EMR, dan. AWS Transfer Family

Amazon Athena

- Menggunakan IAM Identity Center mengaktifkan kelompok kerja Athena

Amazon EMR

- Luncurkan klaster Amazon EMR dengan S3 Access Grants

Amazon EMR di EKS

- Luncurkan Amazon EMR di klaster EKS dengan S3 Access Grants

Aplikasi Amazon EMR Nirserver

- Luncurkan aplikasi Amazon EMR Nirserver dengan S3 Access Grants

Amazon Redshift

- Integrasi Amazon Redshift dengan Hibah Akses Amazon S3

Studio SageMaker AI Amazon

- Menggunakan Amazon S3 Access Grants dengan Amazon SageMaker AI Studio dan plugin SDK for Python (Boto3)

  Menggunakan S3 Access Grants di notebook Amazon SageMaker AI Studio sekarang lebih mudah saat menggunakan plugin SDK for Python (Boto3). Siapkan hibah akses untuk kepala sekolah IAM dan AWS IAM Identity Center pengguna direktori, sebelumnya. Meskipun Amazon SageMaker AI Studio tidak secara native mendukung pengguna direktori penyedia identitas, Anda dapat menulis kode Python khusus, menggunakan plugin yang memungkinkan identitas ini mengakses data S3 melalui S3 Access Grants. Akses data berlangsung dengan bantuan plugin dan bukan melalui Amazon SageMaker AI.

AWS Glue

- Hibah Akses Amazon S3 dengan AWS Glue

AWS IAM Identity Center

- Propagasi identitas tepercaya di seluruh aplikasi

AWS Transfer Family

- Konfigurasikan Hibah Akses Amazon S3 untuk AWS Transfer Family

Browser Penyimpanan untuk S3

- Mengelola akses data dalam skala besar menggunakan Storage Browser untuk S3

Kerangka kerja Python sumber terbuka

- Amazon S3 Access Grants sekarang terintegrasi dengan kerangka kerja Python open source

Mengelola akses dengan ACLs

Access control lists (ACLs) adalah salah satu opsi berbasis sumber daya yang dapat Anda gunakan untuk mengelola akses ke bucket dan objek Anda. Anda dapat menggunakan ACLs untuk memberikan izin baca/tulis dasar kepada yang lain. Akun AWS Ada batasan untuk mengelola izin menggunakan ACLs.

Misalnya, Anda hanya dapat memberikan izin kepada orang lain Akun AWS; Anda tidak dapat memberikan izin kepada pengguna di akun Anda. Anda tidak dapat memberikan izin bersyarat, Anda juga tidak dapat secara eksplisit menolak izin. ACLs cocok untuk skenario tertentu. Misalnya, jika

pemilik bucket mengizinkan orang lain Akun AWS untuk mengunggah objek, izin ke objek ini hanya dapat dikelola menggunakan objek ACL oleh Akun AWS yang memiliki objek tersebut.

S3 Object Ownership adalah setelan tingkat ember Amazon S3 yang dapat Anda gunakan untuk mengontrol kepemilikan objek yang diunggah ke bucket dan untuk menonaktifkan atau mengaktifkan. ACLs Secara default, Kepemilikan Objek disetel ke setelan diberlakukan pemilik Bucket, dan semuanya ACLs dinonaktifkan. Ketika ACLs dinonaktifkan, pemilik bucket memiliki semua objek di bucket dan mengelola akses ke mereka secara eksklusif dengan menggunakan kebijakan manajemen akses.

Mayoritas kasus penggunaan modern di Amazon S3 tidak lagi memerlukan penggunaan. ACLs Kami menyarankan agar Anda tetap ACLs dinonaktifkan, kecuali dalam keadaan yang tidak biasa di mana Anda perlu mengontrol akses untuk setiap objek secara individual. Dengan ACLs dinonaktifkan, Anda dapat menggunakan kebijakan untuk mengontrol akses ke semua objek di bucket, terlepas dari siapa yang mengunggah objek ke bucket Anda. Untuk informasi selengkapnya, lihat Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.

> **Important**
>
> Jika bucket tujuan umum Anda menggunakan pengaturan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek S3, Anda harus menggunakan kebijakan untuk memberikan akses ke bucket tujuan umum dan objek di dalamnya. Dengan pengaturan yang diberlakukan pemilik Bucket diaktifkan, permintaan untuk menyetel daftar kontrol akses (ACLs) atau pembaruan ACLs gagal dan mengembalikan kode `AccessControlListNotSupported` kesalahan. Permintaan untuk membaca ACLs masih didukung.

Untuk informasi selengkapnya ACLs, lihat topik berikut.

Topik

- Gambaran umum daftar kontrol akses (ACL)
- Mengkonfigurasi ACLs
- Contoh kebijakan untuk ACLs

Gambaran umum daftar kontrol akses (ACL)

Daftar kontrol akses Amazon S3 (ACLs) memungkinkan Anda mengelola akses ke bucket dan objek. Setiap bucket dan objek memiliki ACL yang melekat padanya sebagai sumber daya tambahan.

Ini mendefinisikan Akun AWS atau kelompok mana yang diberikan akses dan jenis akses. Saat permintaan diterima terhadap sumber daya, Amazon S3 memeriksa ACL yang sesuai untuk memverifikasi bahwa pengirim permintaan memiliki izin akses yang diperlukan.

S3 Object Ownership adalah setelan tingkat ember Amazon S3 yang dapat Anda gunakan untuk mengontrol kepemilikan objek yang diunggah ke bucket dan untuk menonaktifkan atau mengaktifkan. ACLs Secara default, Kepemilikan Objek disetel ke setelan diberlakukan pemilik Bucket, dan semuanya ACLs dinonaktifkan. Ketika ACLs dinonaktifkan, pemilik bucket memiliki semua objek di bucket dan mengelola akses ke mereka secara eksklusif dengan menggunakan kebijakan manajemen akses.

Mayoritas kasus penggunaan modern di Amazon S3 tidak lagi memerlukan penggunaan. ACLs Kami menyarankan agar Anda tetap ACLs dinonaktifkan, kecuali dalam keadaan yang tidak biasa di mana Anda perlu mengontrol akses untuk setiap objek secara individual. Dengan ACLs dinonaktifkan, Anda dapat menggunakan kebijakan untuk mengontrol akses ke semua objek di bucket, terlepas dari siapa yang mengunggah objek ke bucket Anda. Untuk informasi selengkapnya, lihat Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.

> **Important**
>
> Jika bucket tujuan umum Anda menggunakan pengaturan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek S3, Anda harus menggunakan kebijakan untuk memberikan akses ke bucket tujuan umum dan objek di dalamnya. Dengan pengaturan yang diberlakukan pemilik Bucket diaktifkan, permintaan untuk menyetel daftar kontrol akses (ACLs) atau pembaruan ACLs gagal dan mengembalikan kode `AccessControlListNotSupported` kesalahan. Permintaan untuk membaca ACLs masih didukung.

Saat Anda membuat bucket atau objek, Amazon S3 membuat ACL bawaan yang memberikan kendali penuh kepada pemilik sumber daya atas sumber daya tersebut. Hal ini ditunjukkan dalam ACL bucket sampel berikut (ACL objek bawaan memiliki struktur yang sama):

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<AccessControlPolicy xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Owner>
    <ID>*** Owner-Canonical-User-ID ***</ID>
    <DisplayName>owner-display-name</DisplayName>
  </Owner>
```

```
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xsi:type="Canonical User">
        <ID>*** Owner-Canonical-User-ID ***</ID>
        <DisplayName>display-name</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

ACL sampel menyertakan elemen `Owner` yang mengidentifikasi pemilik dengan ID pengguna kanonik Akun AWS. Untuk petunjuk bagaimana menemukan id pengguna kanonik Anda, lihat Menemukan ID Akun AWS pengguna kanonik. `Grant`Elemen mengidentifikasi penerima hibah (baik grup Akun AWS atau yang telah ditentukan sebelumnya) dan izin yang diberikan. ACL bawaan ini memiliki satu elemen `Grant` untuk pemilik. Anda memberikan izin dengan menambahkan elemen `Grant`, dengan setiap pemberian yang mengidentifikasi penerimanya dan izinnya.

> **Note**
>
> ACL dapat memiliki hingga 100 pemberian.

Topik

- Siapa itu penerima?
- Izin apa yang dapat saya berikan?
- Nilai aclRequired untuk permintaan Amazon S3 umum
- ACL Sampel
- ACL Terekam

Siapa itu penerima?

> **Important**
>
> Pemberitahuan akhir dukungan: Mulai 1 Oktober 2025, Amazon S3 akan menghentikan dukungan untuk membuat Daftar Kontrol Akses Penerima Email (ACL) baru. Penerima Email yang ACLs dibuat sebelum tanggal ini akan terus bekerja dan tetap dapat diakses melalui

AWS Management Console, Command Line Interface (CLI) SDKs, dan REST API. Namun, Anda tidak akan lagi dapat membuat Penerima Email baru. ACLs
Antara 15 Juli 2025 dan 1 Oktober 2025, Anda akan mulai melihat peningkatan tingkat HTTP `405` kesalahan untuk permintaan ke Amazon S3 saat mencoba membuat Penerima Email baru. ACLs
Perubahan ini mempengaruhi hal-hal berikut Wilayah AWS: AS Timur (Virginia N.), AS Barat (California N.), AS Barat (Oregon), Asia Pasifik (Singapura), Asia Pasifik (Sydney), Asia Pasifik (Tokyo), Eropa (Irlandia), dan Amerika Selatan (São Paulo).

Penerima hibah dapat berupa Akun AWS atau salah satu grup Amazon S3 yang telah ditentukan sebelumnya. Anda memberikan izin untuk Akun AWS menggunakan alamat email atau ID pengguna kanonik. Namun, jika Anda memberikan alamat email pada permintaan pemberian Anda, Amazon S3 akan menemukan ID pengguna kanonik untuk akun tersebut dan menambahkannya ke ACL. Hasil ACLs selalu berisi ID pengguna kanonik untuk Akun AWS, bukan alamat email dari. Akun AWS

Saat Anda memberikan hak akses, Anda menentukan setiap penerima sebagai pasangan *`type`*="*`value`*", di mana salah satu dari berikut *`type`* ini:

- `id`— Jika nilai yang ditentukan adalah ID pengguna kanonik dari Akun AWS
- `uri`–jika Anda memberikan izin untuk grup yang telah ditentukan sebelumnya
- `emailAddress`–jika nilai yang disebutkan adalah alamat email dari Akun AWS

Important

Menggunakan alamat email untuk menentukan penerima hanya diberi support di Wilayah AWS berikut ini:

- AS Timur (Virginia Utara)
- AS Barat (California Utara)
- AS Barat (Oregon)
- Asia Pasifik (Singapura)
- Asia Pasifik (Sydney)
- Asia Pasifik (Tokyo)
- Eropa (Irlandia)
- Amerika Selatan (Sao Paulo)

Untuk daftar semua wilayah dan titik akhir yang didukung Amazon S3, lihat Wilayah dan Titik Akhir di Referensi Umum Amazon Web Services.

Example Contoh: Alamat Email

Misalnya, `x-amz-grant-read` header berikut memberikan izin yang Akun AWS diidentifikasi oleh alamat email untuk membaca data objek dan metadatanya:

```
x-amz-grant-read: emailAddress="xyz@example.com", emailAddress="abc@example.com"
```

Warning

Saat Anda memberikan Akun AWS akses lain ke sumber daya Anda, ketahuilah bahwa mereka Akun AWS dapat mendelegasikan izin mereka kepada pengguna di bawah akun mereka. Ini dikenal sebagai akses lintas akun. Untuk informasi tentang menggunakan akses lintas akun, lihat Membuat Peran untuk Mendelegasikan Izin kepada Pengguna IAM dalam Panduan Pengguna IAM.

Menemukan ID Akun AWS pengguna kanonik

ID pengguna kanonik berkaitan dengan Akun AWS Anda. ID ini adalah string panjang karakter, seperti:

`79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be`

Untuk informasi tentang cara menemukan ID pengguna kanonik untuk akun Anda, lihat Menemukan ID pengguna kanonik untuk Anda Akun AWS di Panduan Referensi Manajemen AWS Akun.

Anda juga dapat mencari ID pengguna kanonik Akun AWS dengan membaca ACL bucket atau objek yang Akun AWS memiliki izin akses. Ketika seseorang Akun AWS diberikan izin oleh permintaan hibah, entri hibah ditambahkan ke ACL dengan ID pengguna kanonik akun.

Note

Jika Anda membuat bucket menjadi publik (tidak disarankan), pengguna yang tidak diautentikasi dapat mengunggah objek ke dalam bucket tersebut. Pengguna

anonim ini tidak memiliki Akun AWS. Ketika pengguna anonim mengunggah objek ke bucket Anda, Amazon S3 akan menambahkan ID pengguna kanonik kustom (`65a011a29cdf8ec533ec3d1ccaae921c`) sebagai pemilik objek dalam ACL. Untuk informasi selengkapnya, lihat bucket Amazon S3 dan kepemilikan objek.

Grup Amazon S3 yang sudah ditentukan sebelumnya

Amazon S3 memiliki serangkaian grup yang telah ditentukan sebelumnya. Saat memberikan akses akun ke grup, Anda menentukan salah satu Amazon URIs S3, bukan ID pengguna kanonik. Amazon S3 menyediakan grup yang telah ditentukan sebelumnya berikut ini:

- Grup Pengguna yang Diautentikasi–Diwakili oleh `http://acs.amazonaws.com/groups/global/AuthenticatedUsers`.

  Kelompok ini mewakili semua Akun AWS. Izin akses ke grup ini memungkinkan siapa pun Akun AWS untuk mengakses sumber daya. Namun demikian, semua permintaan harus ditandatangani (diautentikasi).

  > **Warning**
  >
  > Saat Anda memberikan akses ke grup Pengguna Terautentikasi, setiap pengguna yang AWS diautentikasi di dunia dapat mengakses sumber daya Anda.

- Grup Semua Pengguna–Diwakili oleh `http://acs.amazonaws.com/groups/global/AllUsers`.

  Izin akses ke grup ini memungkinkan siapa pun di dunia mengakses ke sumber daya. Permintaan dapat ditandatangani (diautentikasi) atau tidak ditandatangani (anonim). Permintaan yang tidak ditandatangani menghilangkan header Autentikasi dalam permintaan.

  > **Warning**
  >
  > Kami sangat merekomendasikan agar Anda tidak pernah memberikan Grup Semua Pengguna izin `WRITE`, `WRITE_ACP`, atau `FULL_CONTROL`. Misalnya, meskipun `WRITE` izin menolak kemampuan non-pemilik untuk menimpa atau menghapus objek yang ada, `WRITE` izin tetap memungkinkan siapa pun untuk menyimpan objek di bucket Anda, yang mana

Anda ditagih. Untuk perincian lebih lanjut tentang izin ini, lihat bagian berikut Izin apa yang dapat saya berikan?.

- Grup Log Pengiriman–Diwakili oleh `http://acs.amazonaws.com/groups/s3/LogDelivery`.

  Izin `WRITE` pada bucket memungkinkan grup ini menulis log akses server (lihat Pencatatan permintaan dengan pencatatan akses server) ke bucket.

> **Note**
>
> Saat menggunakan ACLs, penerima hibah dapat berupa Akun AWS atau salah satu grup Amazon S3 yang telah ditentukan sebelumnya. Namun, penerima tidak dapat berupa pengguna IAM. Untuk informasi lebih lanjut tentang AWS pengguna dan izin dalam IAM, lihat Menggunakan AWS Identity and Access Management.

Izin apa yang dapat saya berikan?

Tabel berikut mencantumkan serangkaian izin yang didukung Amazon S3 di ACL. Serangkaian izin ACL tersebut sama dengan izin untuk sebuah objek ACL dan bucket ACL. Namun, tergantung pada konteks (ACL bucket atau ACL objek), izin-izin ACL ini memberikan izin untuk bucket atau operasi objek tertentu. Tabel ini mencantumkan izin dan menjelaskan apa artinya dalam konteks objek dan bucket.

Untuk informasi selengkapnya tentang izin ACL di konsol Amazon S3, lihat Mengkonfigurasi ACLs.

Izin	Saat diberikan pada bucket	Saat diberikan pada objek
`READ`	Memungkinkan penerima untuk mencantumkan objek di dalam bucket	Memungkinkan penerima untuk membaca data objek dan metadatanya
`WRITE`	Mengizinkan penerima untuk membuat objek baru di dalam bucket. Untuk pemilik bucket dan objek dari objek yang sudah ada, serta mengizinkan penghapusan dan penimpaan objek tersebut.	Tidak berlaku

Izin	Saat diberikan pada bucket	Saat diberikan pada objek
`READ_ACP`	Memungkinkan penerima untuk membaca ACL bucket	Memungkinkan penerima untuk membaca ACL objek
`WRITE_ACP`	Memungkinkan penerima untuk menulis ACL untuk bucket yang berlaku	Memungkinkan penerima untuk menulis ACL untuk objek yang berlaku
`FULL_CONTROL`	Memungkinkan penerima pemberian `READ`, `WRITE`, `READ_ACP`, dan `WRITE_ACP` izin pada bucket	Memungkinkan penerima pemberian `READ`, `WRITE_ACP` , dan `READ_ACP` izin pada objek

> **Warning**
>
> Berhati-hatilah saat memberikan izin akses ke bucket dan S3 Object Anda. Misalnya, memberikan akses `WRITE` ke bucket mengizinkan penerima membuat, menimpa, dan menghapus objek dalam bucket tersebut. Kami sangat menyarankan Anda membaca seluruh Gambaran umum daftar kontrol akses (ACL) bagian sebelum memberikan izin.

Pemetaan izin ACL dan izin kebijakan akses

Sebagaimana yang ditunjukkan dalam tabel sebelumnya, ACL hanya mengizinkan serangkaian izin terbatas, dibandingkan dengan jumlah izin yang dapat Anda tetapkan dalam kebijakan akses (lihat Tindakan kebijakan untuk Amazon S3). Masing-masing izin ini memungkinkan satu operasi Amazon S3 atau lebih.

Tabel berikut menunjukkan cara setiap izin ACL memetakan ke izin kebijakan akses yang sesuai. Seperti yang dapat Anda lihat, kebijakan akses memungkinkan lebih banyak izin dibandingkan ACL. Anda menggunakan ACLs terutama untuk memberikan read/write izin dasar, mirip dengan izin sistem file. Untuk informasi lebih lanjut tentang kapan harus menggunakan ACL, lihat Identity and Access Management untuk Amazon S3.

Untuk informasi selengkapnya tentang izin ACL di konsol Amazon S3, lihat Mengkonfigurasi ACLs.

Izin ACL	Menyesuaikan izin kebijakan akses ketika izin ACL diberikan pada bucket	Menyesuaikan izin kebijakan akses ketika izin ACL diberikan pada objek
READ	s3:ListBucket , s3:ListBucketVersions , dan s3:ListBucketMultipartUploads	s3:GetObject dan s3:GetObjectVersion
WRITE	s3:PutObject Pemilik bucket dapat membuat, menimpa, dan menghapus objek apa pun di bucket, dan pemilik objek memiliki FULL_CONTROL atas objek mereka. Selain itu, ketika penerima merupakan pemilik bucket, memberikan izin WRITE dalam ACL bucket memungkinkan tindakan s3:DeleteObjectVersion yang harus dilakukan pada versi apa pun dalam bucket tersebut.	Tidak berlaku
READ_ACP	s3:GetBucketAcl	s3:GetObjectAcl dan s3:GetObjectVersionAcl
WRITE_ACP	s3:PutBucketAcl	s3:PutObjectAcl dan s3:PutObjectVersionAcl
FULL_CONTROL	Setara dengan pemberian izin READ, WRITE, READ_ACP, dan izin ACL WRITE_ACP . Oleh karena itu, izin ACL ini dipetakan ke kombinasi izin kebijakan akses yang sesuai.	Setara dengan pemberian izin READ, READ_ACP, dan izin ACL WRITE_ACP . Oleh karena itu, izin ACL ini dipetakan ke kombinasi izin kebijakan akses yang sesuai.

Kunci syarat

Saat Anda memberikan izin kebijakan akses, Anda dapat menggunakan kunci kondisi untuk membatasi nilai ACL pada objek dengan menggunakan kebijakan bucket. Kunci konteks berikut sesuai dengan ACLs. Anda dapat menggunakan kunci konteks ini untuk mewajibkan penggunaan ACL tertentu dalam permintaan:

- `s3:x-amz-grant-read` - Memerlukan akses baca.
- `s3:x-amz-grant-write` - Memerlukan akses tulis.
- `s3:x-amz-grant-read-acp` - Memerlukan akses baca ke ACL bucket.
- `s3:x-amz-grant-write-acp` - Memerlukan akses tulis ke ACL bucket.
- `s3:x-amz-grant-full-control` - Memerlukan kontrol penuh.
- `s3:x-amz-acl` - Memerlukan ACL Terekam.

Misalnya kebijakan yang melibatkan header spesifik ACL, lihat. Pemberian s3: PutObject izin dengan syarat yang mengharuskan pemilik bucket untuk mendapatkan kontrol penuh Untuk daftar lengkap kunci kondisi khusus Amazon S3, lihat Tindakan, sumber daya, dan kunci kondisi untuk Amazon S3 di Referensi Otorisasi Layanan.

Untuk informasi selengkapnya tentang izin operasi S3 API menurut jenis sumber daya S3, lihat. Izin yang diperlukan untuk operasi API Amazon S3

Nilai **`aclRequired`** untuk permintaan Amazon S3 umum

Untuk mengidentifikasi permintaan Amazon S3 yang diperlukan ACLs untuk otorisasi, Anda dapat menggunakan `aclRequired` nilai di log akses server Amazon S3 atau. AWS CloudTrail`aclRequired`Nilai yang muncul di log akses server Amazon S3 CloudTrail atau Amazon S3 bergantung pada operasi mana yang dipanggil dan informasi tertentu tentang pemohon, pemilik objek, dan pemilik bucket. Jika tidak ACLs diperlukan, atau jika Anda menyetel ACL `bucket-owner-full-control` kalengan, atau jika permintaan diizinkan oleh kebijakan bucket Anda, string `aclRequired` nilainya adalah "`-`" di log akses server Amazon S3 dan tidak ada di dalamnya. CloudTrail

Tabel berikut mencantumkan `aclRequired` nilai yang diharapkan dalam CloudTrail atau log akses server Amazon S3 untuk berbagai operasi API Amazon S3. Anda dapat menggunakan informasi ini untuk memahami operasi Amazon S3 mana yang bergantung pada otorisasi ACLs . Dalam tabel berikut, A, B, dan C mewakili akun berbeda yang terkait dengan pemohon, pemilik objek, dan pemilik bucket. Entri dengan tanda bintang (*) menunjukkan salah satu akun A, B, atau C.

> **Note**
>
> Operasi `PutObject` dalam tabel berikut ini, kecuali ditentukan lain, menunjukkan permintaan yang tidak menetapkan ACL, kecuali ACL merupakan ACL `bucket-owner-full-control`. Nilai null untuk `aclRequired` menunjukkan bahwa tidak `aclRequired` ada dalam AWS CloudTrail log.

Tabel berikut menunjukkan `aclRequired` nilai untuk CloudTrail.

Nama operasi	Peminta	Pemilik objek	Pemilik bucket	Kebijakan bucket memberikan akses	Nilai **`aclRequired`**	Alasan
`GetObject`	A	A	A	Ya atau Tidak	null	Akses akun yang sama
`GetObject`	A	B	A	Ya atau Tidak	null	Akses akun yang sama dengan pemilik bucket diberlakukan
`GetObject`	A	A	B	Ya	null	Akses lintas akun yang diberikan oleh kebijakan bucket
`GetObject`	A	A	B	Tidak	Ya	Akses lintas akun

Nama operasi	Peminta	Pemilik objek	Pemilik bucket	Kebijakan bucket memberikan akses	Nilai **aclRequired**	Alasan
						bergantung pada ACL
`GetObject`	A	A	B	Ya	null	Akses lintas akun yang diberikan oleh kebijakan bucket
`GetObject`	A	B	B	Tidak	Ya	Akses lintas akun bergantung pada ACL
`GetObject`	A	B	C	Ya	null	Akses lintas akun yang diberikan oleh kebijakan bucket
`GetObject`	A	B	C	Tidak	Ya	Akses lintas akun bergantung pada ACL
`PutObject`	A	Tidak berlaku	A	Ya atau Tidak	null	Akses akun yang sama

Nama operasi	Peminta	Pemilik objek	Pemilik bucket	Kebijakan bucket memberikan akses	Nilai **aclRequired**	Alasan
PutObject	A	Tidak berlaku	B	Ya	null	Akses lintas akun yang diberikan oleh kebijakan bucket
PutObject	A	Tidak berlaku	B	Tidak	Ya	Akses lintas akun bergantung pada ACL
PutObject dengan ACL (kecuali untuk bucket-owner-full-control)	*	Tidak berlaku	*	Ya atau Tidak	Ya	Permintaan pemberian ACL
ListObjects	A	Tidak berlaku	A	Ya atau Tidak	null	Akses akun yang sama

Nama operasi	Peminta	Pemilik objek	Pemilik bucket	Kebijakan bucket memberikan akses	Nilai **aclRequired**	Alasan
`ListObjects`	A	Tidak berlaku	B	Ya	null	Akses lintas akun yang diberikan oleh kebijakan bucket
`ListObjects`	A	Tidak berlaku	B	Tidak	Ya	Akses lintas akun bergantung pada ACL
`DeleteObject`	A	Tidak berlaku	A	Ya atau Tidak	null	Akses akun yang sama
`DeleteObject`	A	Tidak berlaku	B	Ya	null	Akses lintas akun yang diberikan oleh kebijakan bucket
`DeleteObject`	A	Tidak berlaku	B	Tidak	Ya	Akses lintas akun bergantung pada ACL
`PutObjectAcl`	*	*	*	Ya atau Tidak	Ya	Permintaan pemberian ACL

Nama operasi	Peminta	Pemilik objek	Pemilik bucket	Kebijakan bucket memberikan akses	Nilai **aclRequired**	Alasan
`PutBucketAcl`	*	Tidak berlaku	*	Ya atau Tidak	Ya	Permintaan pemberian ACL

> **Note**
>
> Operasi `REST.PUT.OBJECT` dalam tabel berikut ini, kecuali ditentukan lain, menunjukkan permintaan yang tidak menetapkan ACL, kecuali ACL merupakan ACL `bucket-owner-full-control`. String `aclRequired` nilai "`-`" menunjukkan nilai nol di log akses server Amazon S3.

Tabel berikut menunjukkan `aclRequired` nilai untuk log akses server Amazon S3.

Nama operasi	Peminta	Pemilik objek	Pemilik bucket	Kebijakan bucket memberikan akses	Nilai **aclRequired**	Alasan
`REST.GET.OBJECT`	A	A	A	Ya atau Tidak	-	Akses akun yang sama
`REST.GET.OBJECT`	A	B	A	Ya atau Tidak	-	Akses akun yang sama dengan pemilik bucket

Nama operasi	Peminta	Pemilik objek	Pemilik bucket	Kebijakan bucket memberikan akses	Nilai **aclRequired**	Alasan
						diberlakukan
REST.GET.OBJECT	A	A	B	Ya	-	Akses lintas akun yang diberikan oleh kebijakan bucket
REST.GET.OBJECT	A	A	B	Tidak	Ya	Akses lintas akun bergantung pada ACL
REST.GET.OBJECT	A	B	B	Ya	-	Akses lintas akun yang diberikan oleh kebijakan bucket
REST.GET.OBJECT	A	B	B	Tidak	Ya	Akses lintas akun bergantung pada ACL

Nama operasi	Peminta	Pemilik objek	Pemilik bucket	Kebijakan bucket memberikan akses	Nilai **aclRequired**	Alasan
REST.GET.OBJECT	A	B	C	Ya	-	Akses lintas akun yang diberikan oleh kebijakan bucket
REST.GET.OBJECT	A	B	C	Tidak	Ya	Akses lintas akun bergantung pada ACL
REST.PUT.OBJECT	A	Tidak berlaku	A	Ya atau Tidak	-	Akses akun yang sama
REST.PUT.OBJECT	A	Tidak berlaku	B	Ya	-	Akses lintas akun yang diberikan oleh kebijakan bucket
REST.PUT.OBJECT	A	Tidak berlaku	B	Tidak	Ya	Akses lintas akun bergantung pada ACL

Nama operasi	Peminta	Pemilik objek	Pemilik bucket	Kebijakan bucket memberikan akses	Nilai **aclRequired**	Alasan
REST.PUT.OBJECT dengan ACL (kecuali untuk bucket-owner-full-control)	*	Tidak berlaku	*	Ya atau Tidak	Ya	Permintaan pemberian ACL
REST.GET.BUCKET	A	Tidak berlaku	A	Ya atau Tidak	-	Akses akun yang sama
REST.GET.BUCKET	A	Tidak berlaku	B	Ya	-	Akses lintas akun yang diberikan oleh kebijakan bucket
REST.GET.BUCKET	A	Tidak berlaku	B	Tidak	Ya	Akses lintas akun bergantung pada ACL
REST.DELETE.OBJECT	A	Tidak berlaku	A	Ya atau Tidak	-	Akses akun yang sama

Nama operasi	Peminta	Pemilik objek	Pemilik bucket	Kebijakan bucket memberikan akses	Nilai **aclRequired**	Alasan
REST.DELETE.OBJECT	A	Tidak berlaku	B	Ya	-	Akses lintas akun yang diberikan oleh kebijakan bucket
REST.DELETE.OBJECT	A	Tidak berlaku	B	Tidak	Ya	Akses lintas akun bergantung pada ACL
REST.PUT.ACL	*	*	*	Ya atau Tidak	Ya	Permintaan pemberian ACL

ACL Sampel

ACL sampel pada suatu bucket berikut ini mengidentifikasi pemilik sumber daya dan serangkaian pemberian. Formatnya adalah representasi XML dari ACL dalam Amazon S3 API REST. Pemilik bucket memiliki FULL_CONTROL sumber daya. Selain itu, ACL menunjukkan bagaimana izin diberikan pada sumber daya ke dua Akun AWS, diidentifikasi oleh ID pengguna kanonik, dan dua grup Amazon S3 yang telah ditentukan sebelumnya yang dibahas di bagian sebelumnya.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<AccessControlPolicy xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Owner>
    <ID>Owner-canonical-user-ID</ID>
    <DisplayName>display-name</DisplayName>
  </Owner>
```

```
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:type="CanonicalUser">
        <ID>Owner-canonical-user-ID</ID>
        <DisplayName>display-name</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>

    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:type="CanonicalUser">
        <ID>user1-canonical-user-ID</ID>
        <DisplayName>display-name</DisplayName>
      </Grantee>
      <Permission>WRITE</Permission>
    </Grant>

    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:type="CanonicalUser">
        <ID>user2-canonical-user-ID</ID>
        <DisplayName>display-name</DisplayName>
      </Grantee>
      <Permission>READ</Permission>
    </Grant>

    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
        <URI>http://acs.amazonaws.com/groups/global/AllUsers</URI>
      </Grantee>
      <Permission>READ</Permission>
    </Grant>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
        <URI>http://acs.amazonaws.com/groups/s3/LogDelivery</URI>
      </Grantee>
      <Permission>WRITE</Permission>
    </Grant>

  </AccessControlList>
</AccessControlPolicy>
```

ACL Terekam

Amazon S3 mendukung satu set hibah yang telah ditentukan, yang dikenal sebagai kaleng. ACLs Setiap ACL terekam memiliki seperangkat penerima dan izin yang telah ditetapkan. Tabel berikut mencantumkan kumpulan kaleng ACLs dan hibah yang telah ditentukan sebelumnya.

ACL Terekam	Berlaku untuk	Izin ditambahkan ke ACL
`private`	Bucket dan objek	Pemilik mendapatkan `FULL_CONTROL` . Tidak ada orang lain yang memiliki hak mengakses (default).
`public-read`	Bucket dan objek	Pemilik mendapatkan `FULL_CONTROL` . Grup `AllUsers` (lihat Siapa itu penerima?) mendapat akses `READ`.
`public-read-write`	Bucket dan objek	Pemilik mendapatkan `FULL_CONTROL` . Grup `AllUsers` mendapatkan akses `READ` dan `WRITE`. memberikan ini pada bucket umumnya tidak disarankan.
`aws-exec-read`	Bucket dan objek	Pemilik mendapatkan `FULL_CONTROL` . Amazon EC2 mendapatkan `READ` akses ke `GET` bundel Amazon Machine Image (AMI) dari Amazon S3.
`authenticated-read`	Bucket dan objek	Pemilik mendapatkan `FULL_CONTROL` . Grup `AuthenticatedUsers` mendapat akses `READ`.
`bucket-owner-read`	Objek	Pemilik objek mendapat `FULL_CONTROL` . Pemilik bucket mendapat akses `READ`. Jika Anda menyebutk an ACL terekam ini saat membuat bucket, Amazon S3 akan mengabaikannya.
`bucket-owner-full-control`	Objek	Pemilik objek dan pemilik bucket mendapatkan `FULL_CONTROL` atas objek. Jika Anda menyebutk an ACL terekam ini saat membuat bucket, Amazon S3 akan mengabaikannya.
`log-delivery-write`	Bucket	Grup `LogDelivery` mendapatkan izin `WRITE` dan `READ_ACP` pada bucket. Untuk informasi lebih lanjut

ACL Terekam	Berlaku untuk	Izin ditambahkan ke ACL
		tentang log, lihat (Pencatatan permintaan dengan pencatatan akses server).

> **Note**
>
> Anda hanya dapat menentukan satu dari kaleng ini ACLs dalam permintaan Anda.

Anda menentukan ACL terekam dalam permintaan Anda dengan menggunakan header `x-amz-acl` permintaan. Ketika Amazon S3 menerima permintaan dengan ACL terekam dalam permintaan tersebut, itu akan menambahkan pemberian yang telah ditentukan sebelumnya ke ACL dari sumber daya itu.

Mengkonfigurasi ACLs

Bagian ini menjelaskan cara mengelola izin akses untuk bucket dan objek S3 menggunakan daftar kontrol akses (). ACLs Anda dapat menambahkan hibah ke ACL sumber daya Anda menggunakan, (AWS Command Line Interface CLI) AWS Management Console, REST API, atau. AWS SDKs

bucket dan izin objek bersifat independen satu sama lain. Objek tidak mewarisi izin dari bucketnya. Misalnya, jika Anda membuat bucket dan memberikan akses tertulis kepada pengguna, Anda tidak dapat mengakses objek pengguna tersebut kecuali jika pengguna memberikan Anda akses secara eksplisit.

Anda dapat memberikan izin kepada Akun AWS pengguna lain atau ke grup yang telah ditentukan sebelumnya. Pengguna atau grup yang Anda berikan izin disebut penerima pemberian. Secara default, pemilik, Akun AWS yang membuat bucket, memiliki izin penuh.

Setiap izin yang Anda berikan untuk pengguna atau kelompok menambahkan entri ke dalam ACL yang terkait dengan bucket. ACL mencantumkan daftar pemberian, yang mengidentifikasi penerima pemberian dan izin yang diberikan.

S3 Object Ownership adalah setelan tingkat ember Amazon S3 yang dapat Anda gunakan untuk mengontrol kepemilikan objek yang diunggah ke bucket dan untuk menonaktifkan atau mengaktifkan. ACLs Secara default, Kepemilikan Objek disetel ke setelan diberlakukan pemilik Bucket, dan

semuanya ACLs dinonaktifkan. Ketika ACLs dinonaktifkan, pemilik bucket memiliki semua objek di bucket dan mengelola akses ke mereka secara eksklusif dengan menggunakan kebijakan manajemen akses.

Mayoritas kasus penggunaan modern di Amazon S3 tidak lagi memerlukan penggunaan. ACLs Kami menyarankan agar Anda tetap ACLs dinonaktifkan, kecuali dalam keadaan yang tidak biasa di mana Anda perlu mengontrol akses untuk setiap objek secara individual. Dengan ACLs dinonaktifkan, Anda dapat menggunakan kebijakan untuk mengontrol akses ke semua objek di bucket, terlepas dari siapa yang mengunggah objek ke bucket Anda. Untuk informasi selengkapnya, lihat Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.

> **Important**
>
> Jika bucket tujuan umum Anda menggunakan pengaturan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek S3, Anda harus menggunakan kebijakan untuk memberikan akses ke bucket tujuan umum dan objek di dalamnya. Dengan pengaturan yang diberlakukan pemilik Bucket diaktifkan, permintaan untuk menyetel daftar kontrol akses (ACLs) atau pembaruan ACLs gagal dan mengembalikan kode `AccessControlListNotSupported` kesalahan. Permintaan untuk membaca ACLs masih didukung.

> **Warning**
>
> Kami sangat menyarankan agar Anda menghindari pemberian akses tulis ke grup Semua Orang (akses publik) atau grup Pengguna Terautentikasi (semua pengguna yang AWS diautentikasi). Untuk informasi lebih lanjut tentang efek pemberian akses tulis ke grup ini, lihat Grup Amazon S3 yang sudah ditentukan sebelumnya.

Menggunakan konsol S3 untuk mengatur izin ACL untuk bucket

Konsol tersebut menampilkan pemberian akses gabungan untuk penerima pemberian duplikat. Untuk melihat daftar lengkap ACLs, gunakan Amazon S3 REST API, AWS CLI, atau. AWS SDKs

Tabel berikut menunjukkan izin ACL yang dapat Anda konfigurasikan untuk bucket di konsol Amazon S3.

Izin ACL konsol Amazon S3 untuk bucket

Izin konsol	Izin ACL	Akses
Objek-Daftar	`READ`	Memungkinkan penerima untuk mencantumkan objek di dalam bucket.
Objek-Tulis	`WRITE`	Mengizinkan penerima untuk membuat objek baru di dalam bucket. Untuk pemilik bucket dan objek dari objek yang sudah ada, serta mengizinkan penghapusan dan penimpaan objek tersebut.
Bucket ACL-Baca	`READ_ACP`	Memungkinkan penerima untuk membaca ACL bucket.
Bucket ACL-Tulis	`WRITE_ACP`	Memungkinkan penerima untuk menulis ACL untuk bucket yang berlaku.
Semua orang (akses publik): Objek-Daftar	`READ`	Hibah akses baca publik untuk objek dalam bucket. Ketika Anda memberikan akses daftar ke Semua orang (akses publik), siapa pun di dunia ini dapat mengakses objek yang ada dalam bucket.
Semua orang (akses publik): Bucket ACL-Baca	`READ_ACP`	Hibah akses baca publik untuk objek dalam bucket ACL. Ketika Anda memberikan akses baca ke Semua orang (akses publik), siapa pun di dunia ini dapat mengakses objek yang ada dalam bucket ACL.

Untuk informasi lebih lanjut tentang izin ACL, lihat Gambaran umum daftar kontrol akses (ACL).

Important

Jika bucket tujuan umum Anda menggunakan pengaturan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek S3, Anda harus menggunakan kebijakan untuk memberikan akses ke bucket tujuan umum dan objek di dalamnya. Dengan pengaturan yang diberlakukan pemilik Bucket diaktifkan, permintaan untuk menyetel daftar kontrol akses (ACLs) atau

pembaruan ACLs gagal dan mengembalikan kode `AccessControlListNotSupported` kesalahan. Permintaan untuk membaca ACLs masih didukung.

Untuk mengatur izin ACL untuk bucket

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Di Bucket Anda, pilih nama bucket yang ingin Anda tetapkan izinnya.
4. Pilih Izin.
5. Di bawah Daftar kontrol akses, memilih Edit.

   Anda dapat mengedit izin ACL berikut untuk bucket:

   Objek

   - Daftar–Mengizinkan penerima untuk mencantumkan objek di dalam bucket.
   - Tulis–Mengizinkan penerima untuk membuat objek baru di dalam bucket. Untuk pemilik bucket dan objek dari objek yang sudah ada, serta mengizinkan penghapusan dan penimpaan objek tersebut.

     Di konsol S3, Anda hanya dapat memberikan akses tulis ke grup pengiriman log S3 dan pemilik bucket (milik Anda Akun AWS). Kami sangat merekomendasikan Anda untuk tidak memberikan akses tulis untuk penerima pemberian lainnya. Namun, jika Anda perlu memberikan akses tulis, Anda dapat menggunakan AWS CLI, AWS SDKs, atau REST API.

   ACL Bucket

   - Baca–Memungkinkan penerima untuk membaca ACL bucket.
   - Tulis–Memungkinkan penerima untuk menulis ACL untuk bucket yang berlaku.
6. Untuk mengubah izin pemilik bucket, di samping pemilik Bucket (Anda Akun AWS), hapus atau pilih dari izin ACL berikut:

   - Objek–Daftar atau Tulis
   - ACL bucket–Baca atau Tulis

Pemilik mengacu pada Pengguna root akun AWS, bukan pengguna AWS Identity and Access Management IAM. Untuk informasi selengkapnya tentang pengguna root, lihat Pengguna root akun AWS dalam Panduan Pengguna IAM.

7. Untuk memberikan atau membatalkan izin untuk masyarakat umum (semua orang di internet), di samping Semua orang (akses publik), hapus atau pilih dari izin ACL berikut:

   - Objek–Daftar
   - ACL bucket–Baca

   > ⚠ Warning
   >
   > Berhati-hatilah saat memberikan Semua akses publik kelompok ke bucket S3 Anda. Ketika Anda memberikan akses ke grup ini, siapa pun di dunia dapat mengakses bucket Anda. Kami sangat menyarankan Anda untuk tidak pernah memberikan akses tulis kepada publik ke bucket S3 Anda.

8. Untuk memberikan atau membatalkan izin bagi siapa pun yang memiliki Akun AWS, di samping grup Pengguna Terautentikasi (siapa pun yang memiliki Akun AWS), hapus atau pilih dari izin ACL berikut:

   - Objek–Daftar
   - ACL bucket–Baca

9. Untuk memberikan atau membatalkan izin bagi Amazon S3 untuk tulis log akses server ke bucket, di bawah Grup pengiriman log S3, hapus atau pilih dari izin ACL berikut:

   - Objek–Daftar atau Tulis
   - ACL bucket–Baca atau Tulis

   Jika bucket disiapkan sebagai bucket target untuk menerima catatan akses, izin bucket harus mengizinkan Log Pengiriman kelompok menulis akses ke bucket. Saat Anda mengaktifkan log masuk akses server ke dalam bucket, konsol Amazon S3 memberikan akses tertulis ke Log Pengiriman kelompok untuk bucket target yang Anda pilih untuk menerima log. Untuk informasi selengkapnya tentang pencatatan log akses server, lihat Mengaktifkan pencatatan akses server Amazon S3.

10. Untuk memberikan akses ke yang lain Akun AWS, lakukan hal berikut:

a. Pilih Tambahkan penerima.
b. Di kotak Penerima, memasukkan ID kanonik Akun AWS yang lain.
c. Pilih dari izin ACL berikut:

   - Objek–Daftar atau Tulis
   - ACL bucket–Baca atau Tulis

> **Warning**
>
> Saat Anda memberikan Akun AWS akses lain ke sumber daya Anda, ketahuilah bahwa mereka Akun AWS dapat mendelegasikan izin mereka kepada pengguna di bawah akun mereka. Ini dikenal sebagai akses lintas akun. Untuk informasi tentang menggunakan akses lintas akun, lihat Membuat Peran untuk Mendelegasikan Izin kepada Pengguna IAM dalam Panduan Pengguna IAM.

11. Untuk menghapus akses ke yang lain Akun AWS, di bawah Access for other Akun AWS, pilih Hapus.
12. Untuk menyimpan perubahan Anda, memilih Simpan perubahan.

Menggunakan konsol S3 untuk mengatur izin ACL untuk objek

Konsol tersebut menampilkan pemberian akses gabungan untuk penerima pemberian duplikat. Untuk melihat daftar lengkap ACLs, gunakan Amazon S3 REST API, AWS CLI, atau. AWS SDKs Tabel berikut menunjukkan izin ACL yang dapat Anda konfigurasikan untuk bucket di konsol Amazon S3.

Izin ACL konsol Amazon S3 untuk objek

Izin konsol	Izin ACL	Akses
Objek-Baca	`READ`	Memungkinkan penerima untuk membaca data objek dan metadatanya.
Objek ACL-Baca	`READ_ACP`	Mengizinkan penerima untuk membaca ACL objek.
Objek ACL-Tulis	`WRITE_ACP`	Memungkinkan penerima untuk menulis ACL untuk objek yang berlaku

Untuk informasi selengkapnya tentang izin ACL, lihat Gambaran umum daftar kontrol akses (ACL).

> **Important**
>
> Jika bucket tujuan umum Anda menggunakan pengaturan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek S3, Anda harus menggunakan kebijakan untuk memberikan akses ke bucket tujuan umum dan objek di dalamnya. Dengan pengaturan yang diberlakukan pemilik Bucket diaktifkan, permintaan untuk menyetel daftar kontrol akses (ACLs) atau pembaruan ACLs gagal dan mengembalikan kode `AccessControlListNotSupported` kesalahan. Permintaan untuk membaca ACLs masih didukung.

Untuk mengatur izin ACL untuk objek

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di dalam daftar Bucket, pilih nama bucket yang berisi objek.
3. Di daftar Objek, memilih nama objek yang ingin Anda atur izinnya.
4. Pilih Izin.
5. Di bagian bawah Daftar kontrol akses (ACL), memilih Edit.

   Anda dapat mengedit izin ACL berikut untuk objek:

   Objek

   - Baca–Memungkinkan penerima untuk membaca data objek dan metadatanya.

   ACL Objek

   - Baca–Mengizinkan penerima untuk baca ACL objek.
   - Tulis–Mengizinkan penerima untuk tulis ACL untuk objek yang dapat diaplikasikan. Di konsol S3, Anda hanya dapat memberikan akses tulis ke pemilik bucket (milik Anda Akun AWS). Kami sangat merekomendasikan Anda untuk tidak memberikan akses tulis untuk penerima pemberian lainnya. Namun, jika Anda perlu memberikan akses tulis, Anda dapat menggunakan AWS CLI, AWS SDKs, atau REST API.
6. Anda dapat mengelola izin akses objek untuk hal berikut:

a. Akses untuk pemilik objek

   Pemilik mengacu pada Pengguna root akun AWS, dan bukan pengguna AWS Identity and Access Management IAM. Untuk informasi selengkapnya tentang pengguna root, lihat Pengguna root akun AWS dalam Panduan Pengguna IAM.

   Untuk mengubah izin akses objek pemilik, di bawah Akses untuk pemilik objek, pilih AWS Akun Anda (pemilik).

   Pilih kotak periksa untuk izin yang ingin Anda berikan, lalu memilih Simpan.

b. Akses untuk lainnya Akun AWS

   Untuk memberikan izin kepada AWS pengguna dari yang lain Akun AWS, di bawah Access for other Akun AWS, pilih Tambah akun. Di bidang Masukkan ID, masukkan ID kanonik AWS pengguna yang ingin Anda berikan izin objek. Untuk informasi tentang menemukan ID kanonik, lihat Akun AWS Pengidentifikasi Anda di. Referensi Umum Amazon Web Services Anda dapat menambahkan sebanyak 99 pengguna.

   Pilih kotak periksa untuk izin yang ingin Anda berikan ke pengguna, lalu memilih Simpan. Untuk menampilkan informasi tentang izin, memilih ikon Bantuan.

c. Akses publik

   Untuk memberikan akses ke objek Anda ke masyarakat umum (semua orang di dunia), di bawah Akses publik, memilih Semua orang. Pemberian izin akses publik berarti siapa pun di dunia dapat mengakses objek tersebut.

   Pilih kotak centang untuk izin yang ingin Anda berikan, lalu pilih Simpan.

   > ⚠ Warning
   >
   > - Berhati-hatilah saat memberikan Semua akses grup anonim ke objek Amazon S3. Ketika Anda memberikan akses ke grup ini, siapa pun di dunia ini dapat mengakses objek Anda. Jika Anda perlu memberikan akses ke semua orang, kami sangat menyarankan agar Anda hanya memberikan izin untuk Baca objek.
   > - Kami sangat menyarankan Anda untuk tidak memberikan Semua kelompok menulis izin objek. Dengan begitu, siapa pun dapat menimpa izin ACL untuk objek tersebut.

Menggunakan AWS SDKs

Bagian ini menyediakan contoh-contoh cara mengonfigurasikan pemberian daftar kontrol akses (ACL) pada bucket dan objek.

> **Important**
>
> Jika bucket tujuan umum Anda menggunakan pengaturan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek S3, Anda harus menggunakan kebijakan untuk memberikan akses ke bucket tujuan umum dan objek di dalamnya. Dengan pengaturan yang diberlakukan pemilik Bucket diaktifkan, permintaan untuk menyetel daftar kontrol akses (ACLs) atau pembaruan ACLs gagal dan mengembalikan kode `AccessControlListNotSupported` kesalahan. Permintaan untuk membaca ACLs masih didukung.

Java

Bagian ini menyediakan contoh-contoh cara mengonfigurasikan pemberian daftar kontrol akses (ACL) pada bucket dan objek. Contoh pertama membuat bucket dengan ACL terekam (lihat ACL Terekam), membuat daftar pemberian izin kustom, lalu mengganti ACL terekam dengan ACL yang berisi pemberian kustom. Contoh kedua menunjukkan cara memodifikasi ACL dengan menggunakan metode `AccessControlList.grantPermission()`.

Example Membuat bucket dan menentukan ACL terekam yang memberikan izin ke grup pengiriman log S3

Contoh ini membuat bucket. Dalam permintaan tersebut, contoh menyebutkan ACL terekam yang memberikan izin kepada grup Pengiriman Log untuk menulis log ke bucket.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.io.IOException;
import java.util.ArrayList;

public class CreateBucketWithACL {
```

```
   public static void main(String[] args) throws IOException {
       Regions clientRegion = Regions.DEFAULT_REGION;
       String bucketName = "*** Bucket name ***";
       String userEmailForReadPermission = "*** user@example.com ***";

       try {
           AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                   .withRegion(clientRegion)
                   .build();

           // Create a bucket with a canned ACL. This ACL will be replaced by the
           // setBucketAcl()
           // calls below. It is included here for demonstration purposes.
           CreateBucketRequest createBucketRequest = new
CreateBucketRequest(bucketName, clientRegion.getName())
                   .withCannedAcl(CannedAccessControlList.LogDeliveryWrite);
           s3Client.createBucket(createBucketRequest);

           // Create a collection of grants to add to the bucket.
           ArrayList<Grant> grantCollection = new ArrayList<Grant>();

           // Grant the account owner full control.
           Grant grant1 = new Grant(new
CanonicalGrantee(s3Client.getS3AccountOwner().getId()),
                   Permission.FullControl);
           grantCollection.add(grant1);

           // Grant the LogDelivery group permission to write to the bucket.
           Grant grant2 = new Grant(GroupGrantee.LogDelivery, Permission.Write);
           grantCollection.add(grant2);

           // Save grants by replacing all current ACL grants with the two we just
created.
           AccessControlList bucketAcl = new AccessControlList();
           bucketAcl.grantAllPermissions(grantCollection.toArray(new Grant[0]));
           s3Client.setBucketAcl(bucketName, bucketAcl);

           // Retrieve the bucket's ACL, add another grant, and then save the new
ACL.
           AccessControlList newBucketAcl = s3Client.getBucketAcl(bucketName);
           Grant grant3 = new Grant(new
EmailAddressGrantee(userEmailForReadPermission), Permission.Read);
           newBucketAcl.grantAllPermissions(grant3);
```

```
            s3Client.setBucketAcl(bucketName, newBucketAcl);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Example Memperbarui ACL pada objek yang sudah ada

Contoh ini memperbarui ACL pada objek. Contoh tersebut melakukan tugas sebagai berikut:

- Mengambil ACL objek
- Menghapus ACL dengan menghapus semua izin yang ada
- Menambahkan dua izin: akses penuh kepada pemilik, dan WRITE_ACP (lihat Izin apa yang dapat saya berikan?) kepada pengguna yang diidentifikasi melalui alamat email
- Menyimpan ACL ke objek

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.AccessControlList;
import com.amazonaws.services.s3.model.CanonicalGrantee;
import com.amazonaws.services.s3.model.EmailAddressGrantee;
import com.amazonaws.services.s3.model.Permission;

import java.io.IOException;

public class ModifyACLExistingObject {

    public static void main(String[] args) throws IOException {
```

```
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String keyName = "*** Key name ***";
        String emailGrantee = "*** user@example.com ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            // Get the existing object ACL that we want to modify.
            AccessControlList acl = s3Client.getObjectAcl(bucketName, keyName);

            // Clear the existing list of grants.
            acl.getGrantsAsList().clear();

            // Grant a sample set of permissions, using the existing ACL owner for
 Full
            // Control permissions.
            acl.grantPermission(new CanonicalGrantee(acl.getOwner().getId()),
 Permission.FullControl);
            acl.grantPermission(new EmailAddressGrantee(emailGrantee),
 Permission.WriteAcp);

            // Save the modified ACL back to the object.
            s3Client.setObjectAcl(bucketName, keyName, acl);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

Example Membuat bucket dan menentukan ACL terekam yang memberikan izin ke grup pengiriman log S3

Contoh C# ini membuat bucket. Dalam permintaan tersebut, kode juga menentukan ACL terekam yang memberikan izin kepada grup Pengiriman Log untuk menulis log ke dalam bucket.

Untuk informasi tentang menyiapkan dan menjalankan contoh kode, lihat Memulai SDK for .NET di AWSAWS SDK for .NET Developer Guide.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class ManagingBucketACLTest
    {
        private const string newBucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            CreateBucketUseCannedACLAsync().Wait();
        }

        private static async Task CreateBucketUseCannedACLAsync()
        {
            try
            {
                // Add bucket (specify canned ACL).
                PutBucketRequest putBucketRequest = new PutBucketRequest()
                {
                    BucketName = newBucketName,
                    BucketRegion = S3Region.EUW1, // S3Region.US,
                                                  // Add canned ACL.
```

```
                    CannedACL = S3CannedACL.LogDeliveryWrite
                };
                PutBucketResponse putBucketResponse = await
 client.PutBucketAsync(putBucketRequest);

                // Retrieve bucket ACL.
                GetACLResponse getACLResponse = await client.GetACLAsync(new
 GetACLRequest
                {
                    BucketName = newBucketName
                });
            }
            catch (AmazonS3Exception amazonS3Exception)
            {
                Console.WriteLine("S3 error occurred. Exception: " +
 amazonS3Exception.ToString());
            }
            catch (Exception e)
            {
                Console.WriteLine("Exception: " + e.ToString());
            }
        }
    }
}
```

Example Memperbarui ACL pada objek yang sudah ada

Contoh C# ini memperbarui ACL pada objek yang sudah ada. Contoh tersebut melakukan tugas sebagai berikut:

- Mengambil ACL objek.
- Menghapus ACL dengan menghapus semua izin yang ada.
- Menambahkan dua izin: akses penuh kepada pemilik, dan WRITE_ACP kepada pengguna yang diidentifikasi melalui alamat email.
- Menyimpan ACL dengan mengirim permintaan `PutAcl`.

Untuk informasi tentang menyiapkan dan menjalankan contoh kode, lihat Memulai SDK for .NET di AWSAWS SDK for .NET Developer Guide.

```
using Amazon;
using Amazon.S3;
```

```
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class ManagingObjectACLTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string keyName = "*** object key name ***";
        private const string emailAddress = "*** email address ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 client;
        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            TestObjectACLTestAsync().Wait();
        }
        private static async Task TestObjectACLTestAsync()
        {
            try
            {
                    // Retrieve the ACL for the object.
                    GetACLResponse aclResponse = await client.GetACLAsync(new
 GetACLRequest
                    {
                        BucketName = bucketName,
                        Key = keyName
                    });

                    S3AccessControlList acl = aclResponse.AccessControlList;

                    // Retrieve the owner (we use this to re-add permissions after
 we clear the ACL).
                    Owner owner = acl.Owner;

                    // Clear existing grants.
                    acl.Grants.Clear();

                    // Add a grant to reset the owner's full permission (the
 previous clear statement removed all permissions).
```

```
                    S3Grant fullControlGrant = new S3Grant
                    {
                        Grantee = new S3Grantee { CanonicalUser = owner.Id },
                        Permission = S3Permission.FULL_CONTROL

                    };

                    // Describe the grant for the permission using an email address.
                    S3Grant grantUsingEmail = new S3Grant
                    {
                        Grantee = new S3Grantee { EmailAddress = emailAddress },
                        Permission = S3Permission.WRITE_ACP
                    };
                    acl.Grants.AddRange(new List<S3Grant> { fullControlGrant,
 grantUsingEmail });

                    // Set a new ACL.
                    PutACLResponse response = await client.PutACLAsync(new
 PutACLRequest
                    {
                        BucketName = bucketName,
                        Key = keyName,
                        AccessControlList = acl
                    });
            }
            catch (AmazonS3Exception amazonS3Exception)
            {
                Console.WriteLine("An AmazonS3Exception was thrown. Exception: " +
 amazonS3Exception.ToString());
            }
            catch (Exception e)
            {
                Console.WriteLine("Exception: " + e.ToString());
            }
        }
    }
}
```

Penggunaan API REST

Amazon S3 APIs memungkinkan Anda menyetel ACL saat membuat bucket atau objek. Amazon S3 juga menyediakan API untuk menetapkan ACL pada bucket atau objek yang ada. APIs ini memberikan metode-metode berikut untuk menetapkan ACL:

- Tetapkan ACL menggunakan header permintaan— Ketika Anda mengirim permintaan untuk membuat sumber daya (bucket atau objek), Anda mengatur ACL dengan menggunakan header permintaan. Dengan menggunakan header ini, Anda dapat menentukan ACL terekam atau menentukan pemberian secara jelas (mengidentifikasi penerima dan izin secara jelas).
- Tetapkan ACL menggunakan isi permintaan— Ketika Anda mengirim permintaan untuk menetapkan ACL pada sumber daya yang sudah ada, Anda dapat mengatur ACL baik pada header permintaan atau isinya.

Untuk informasi tentang dukungan REST API untuk mengelola ACLs, lihat bagian berikut di Referensi API Amazon Simple Storage Service:

- GetBucketAcl
- PutBucketAcl
- GetObjecAcl
- PutObjectAcl
- PutObject
- CreateBucket
- CopyObject
- CreateMultipartUpload

> **Important**
>
> Jika bucket tujuan umum Anda menggunakan pengaturan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek S3, Anda harus menggunakan kebijakan untuk memberikan akses ke bucket tujuan umum dan objek di dalamnya. Dengan pengaturan yang diberlakukan pemilik Bucket diaktifkan, permintaan untuk menyetel daftar kontrol akses (ACLs) atau pembaruan ACLs gagal dan mengembalikan kode `AccessControlListNotSupported` kesalahan. Permintaan untuk membaca ACLs masih didukung.

Daftar Kontrol Akses (ACL)-Header Permintaan Spesifik

Anda dapat menggunakan header untuk memberikan izin berbasis daftar kontrol akses (ACL). Secara bawaan, semua objek bersifat privat. Hanya pemilik yang memiliki kontrol akses penuh. Saat menambahkan objek baru, Anda dapat memberikan izin ke individu Akun AWS atau grup yang telah ditentukan sebelumnya yang ditentukan oleh Amazon S3. Izin ini kemudian ditambahkan ke Daftar Kontrol Akses (ACL) pada objek. Untuk informasi selengkapnya, lihat Gambaran umum daftar kontrol akses (ACL).

Dengan operasi ini, Anda dapat memberikan izin akses dengan menggunakan salah satu dari dua metode berikut:

- Canned ACL (**`x-amz-acl`**) - Amazon S3 mendukung satu set ACLs standar, yang dikenal sebagai kaleng. ACLs Setiap ACL terekam memiliki seperangkat penerima dan izin yang telah ditetapkan. Untuk informasi selengkapnya, lihat ACL Terekam.
- Izin Akses — Untuk secara eksplisit memberikan izin akses ke grup Akun AWS atau tertentu, gunakan header berikut. Setiap header memetakan ke izin tertentu yang didukung Amazon S3 di ACL. Untuk informasi selengkapnya, lihat Gambaran umum daftar kontrol akses (ACL). Pada header tersebut, Anda menentukan daftar penerima yang mendapatkan izin kustom.
  - x-amz-grant-read
  - x-amz-grant-write
  - x-amz-grant-read-ACP
  - x-amz-grant-write-ACP
  - x-amz-grant-full-kontrol

Menggunakan AWS CLI

Untuk informasi selengkapnya tentang mengelola ACLs penggunaan AWS CLI, lihat put-bucket-acldi Referensi AWS CLI Perintah.

> **Important**
>
> Jika bucket tujuan umum Anda menggunakan pengaturan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek S3, Anda harus menggunakan kebijakan untuk memberikan akses ke bucket tujuan umum dan objek di dalamnya. Dengan pengaturan yang diberlakukan pemilik Bucket diaktifkan, permintaan untuk menyetel daftar kontrol akses (ACLs) atau

pembaruan ACLs gagal dan mengembalikan kode `AccessControlListNotSupported` kesalahan. Permintaan untuk membaca ACLs masih didukung.

Contoh kebijakan untuk ACLs

Anda dapat menggunakan kunci kondisi dalam kebijakan bucket untuk mengontrol akses ke Amazon S3.

Topik

- Pemberian s3: PutObject izin dengan syarat yang mengharuskan pemilik bucket untuk mendapatkan kontrol penuh
- Pemberian s3: PutObject izin dengan kondisi di header x-amz-acl

Pemberian s3: PutObject izin dengan syarat yang mengharuskan pemilik bucket untuk mendapatkan kontrol penuh

Operasi PUT Objek memungkinkan header spesifik daftar kontrol akses (ACL) yang dapat Anda gunakan untuk memberikan izin berbasis ACL. Dengan menggunakan kunci ini, pemilik bucket dapat menetapkan kondisi-kondisi untuk mengharuskan izin akses kustom ketika pengguna mengunggah objek.

Seandainya ketika Akun A memiliki bucket, dan administrator akun ingin memberikan izin kepada Dave, pengguna di Akun B, untuk mengunggah objek. Secara bawaan, objek yang diunggah Dave dimiliki oleh Akun B, dan Akun A tidak memiliki izin pada objek tersebut. Karena pemilik bucket membayar tagihan, pemilik bucket menginginkan izin penuh pada objek yang diunggah oleh Dave. Administrator Akun A dapat melakukan hal ini dengan memberikan izin `s3:PutObject` kepada Dave, dengan kondisi bahwa permintaan tersebut mencakup header spesifik ACL yang memberikan izin penuh secara jelas atau menggunakan ACL terekam. Untuk informasi lebih lanjut, lihat PUT Objek.

Membutuhkan x-amz-full-control header

Anda dapat mengharuskan header `x-amz-full-control` pada permintaan dengan izin kontrol penuh kepada pemilik bucket. Kebijakan bucket berikut ini memberikan izin `s3:PutObject` bagi pengguna Dave dengan syarat yang menggunakan kunci syarat `s3:x-amz-grant-full-control`, yang mengharuskan permintaan untuk menyertakan header `x-amz-full-control`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "statement1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::AccountB-ID:user/Dave"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::awsexamplebucket1/*",
      "Condition": {
        "StringEquals": {
          "s3:x-amz-grant-full-control": "id=AccountA-CanonicalUserID"
        }
      }
    }
  ]
}
```

> **Note**
>
> Contoh ini adalah tentang izin lintas akun. Namun, jika Dave (yang mendapatkan izin) milik pemilik Akun AWS ember, izin bersyarat ini tidak diperlukan. Hal ini karena akun induk yang merupakan pemilik Dave memiliki objek yang diunggah pengguna.

Menambahkan penolakan eksplisit

Kebijakan bucket sebelumnya memberikan izin bersyarat kepada pengguna Dave yang ada di Akun B. Meskipun kebijakan ini berlaku, Dave mungkin mendapatkan izin yang sama tanpa persyaratan apa pun melalui kebijakan lain. Misalnya, Dave dapat menjadi bagian dari sebuah kelompok, dan Anda memberikan kepada kelompok tersebut izin `s3:PutObject` tanpa syarat apa pun. Untuk menghindari celah izin seperti itu, Anda dapat membuat kebijakan akses yang lebih ketat dengan menambahkan penolakan secara eksplisit. Dalam contoh ini, Anda secara jelas menolak izin pengunggahan pengguna Dave jika dia tidak menyertakan header yang diperlukan dalam permintaan yang memberikan izin penuh kepada pemilik bucket. Penolakan eksplisit selalu menggantikan izin lain yang diberikan. Berikut ini adalah contoh kebijakan akses yang direvisi dengan ditambahkan penolakan eksplisit.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "statement1",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::AccountB-ID:user/AccountBadmin"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::awsexamplebucket1/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-grant-full-control": "id=AccountA-CanonicalUserID"
                }
            }
        },
        {
            "Sid": "statement2",
            "Effect": "Deny",
            "Principal": {
                "AWS": "arn:aws:iam::AccountB-ID:user/AccountBadmin"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::awsexamplebucket1/*",
            "Condition": {
                "StringNotEquals": {
                    "s3:x-amz-grant-full-control": "id=AccountA-CanonicalUserID"
                }
            }
        }
    ]
}
```

Uji kebijakan dengan AWS CLI

Jika Anda memiliki dua Akun AWS, Anda dapat menguji kebijakan menggunakan AWS Command Line Interface (AWS CLI). Anda melampirkan kebijakan dan menggunakan kredensi Dave untuk menguji izin menggunakan perintah berikut. AWS CLI `put-object` Anda memberikan kredensial Dave dengan menambahkan parameter `--profile`. Anda memberikan izin kontrol penuh kepada pemilik bucket dengan menambahkan parameter `--grant-full-control`. Untuk informasi

selengkapnya tentang pengaturan dan penggunaan AWS CLI, lihat Mengembangkan dengan Amazon S3 menggunakan AWS CLI di Referensi API Amazon S3.

```
aws s3api put-object --bucket examplebucket --key HappyFace.jpg --body c:\HappyFace.jpg
 --grant-full-control id="AccountA-CanonicalUserID" --profile AccountBUserProfile
```

Membutuhkan x-amz-acl header

Anda dapat mengharuskan header `x-amz-acl` dengan ACL terekam yang memberikan izin kontrol penuh kepada pemilik bucket. Untuk mengharuskan header `x-amz-acl` pada permintaan, Anda dapat mengganti pasangan nilai kunci dalam pemblokiran `Condition` dan tentukan kunci syarat `s3:x-amz-acl`, sebagaimana yang ditunjukkan pada contoh berikut.

```
"Condition": {
    "StringEquals": {
        "s3:x-amz-acl": "bucket-owner-full-control"
    }
}
```

Untuk menguji izin menggunakan AWS CLI, Anda menentukan `--acl` parameter. AWS CLI Kemudian menambahkan `x-amz-acl` header ketika mengirim permintaan.

```
aws s3api put-object --bucket examplebucket --key HappyFace.jpg --body c:\HappyFace.jpg
 --acl "bucket-owner-full-control" --profile AccountBadmin
```

Pemberian s3: PutObject izin dengan kondisi di header x-amz-acl

Kebijakan bucket berikut memberikan `s3:PutObject` izin untuk dua orang Akun AWS jika permintaan menyertakan `x-amz-acl` header yang membuat objek dapat dibaca secara publik. Pemblokiran `Condition` menggunakan kondisi `StringEquals` dan diberi pasangan nilai kunci, `"s3:x-amz-acl":["public-read"]`, untuk evaluasi. Dalam pasangan nilai kunci, `s3:x-amz-acl` adalah kunci spesifik–Amazon S3, sebagaimana yang ditunjukkan dengan prefiks `s3:`.

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
         "Sid":"AddCannedAcl",
            "Effect":"Allow",
         "Principal": {
```

```
                "AWS": [
                "arn:aws:iam::Account1-ID:root",
    "arn:aws:iam::Account2-ID:root"
    ]
            },
         "Action":"s3:PutObject",
            "Resource": ["arn:aws:s3:::awsexamplebucket1/*"],
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl":["public-read"]
                }
            }
        }
    ]
}
```

> **Important**
>
> Tidak semua kondisi masuk akal untuk semua tindakan. Misalnya, memasukkan kondisi `s3:LocationConstraint` dalam kebijakan yang memberikan izin `s3:CreateBucket` Amazon S3 adalah hal yang masuk akal. Namun, tidak masuk akal untuk memasukkan syarat ini pada kebijakan yang memberikan izin `s3:GetObject`. Amazon S3 dapat menguji jenis kesalahan semantik seperti ini yang melibatkan syarat spesifik Amazon S3. Namun, jika Anda membuat kebijakan untuk pengguna IAM dan Anda menyertakan kondisi Amazon S3 yang tidak valid secara semantis, tidak akan ada kesalahan yang dilaporkan karena IAM tidak dapat memvalidasi syarat Amazon S3.

Melakukan blok akses publik ke penyimpanan Amazon S3 Anda

Fitur Blokir Akses Publik Amazon S3 menyediakan pengaturan untuk titik akses, bucket, dan akun untuk membantu Anda mengelola akses publik ke sumber daya Amazon S3. Secara bawaan, bucket baru, titik akses, dan objek baru tidak mengizinkan akses publik. Namun, pengguna dapat memodifikasi kebijakan bucket, kebijakan titik akses, atau izin objek untuk memungkinkan akses publik. Pengaturan S3 Blokir Akses Publik menolak kebijakan-kebijakan dan izin tersebut sehingga Anda dapat membatasi akses publik ke sumber daya.

Dengan S3 Blokir Akses Publik, administrator akun dan pemilik bucket dapat dengan mudah menyiapkan kontrol terpusat untuk membatasi akses publik ke sumber daya Amazon S3 mereka yang diberlakukan terlepas dari bagaimana sumber daya tersebut dibuat.

Untuk petunjuk tentang mengonfigurasi akses blok publik, lihat Mengonfigurasi blokir akses publik.

Saat Amazon S3 menerima permintaan untuk mengakses bucket atau objek, itu akan menentukan apakah akun dari bucket atau pemilik bucket menerapkan blokir akses publik. Jika permintaan tersebut dibuat melalui titik akses, maka Amazon S3 juga memeriksa pengaturan blokir akses publik untuk titik akses tersebut. Jika ada pengaturan blokir akses publik yang sudah ada yang melarang akses yang diminta tersebut, maka Amazon S3 akan menolak permintaan tersebut.

Blokir Akses Publik Amazon S3 menyediakan empat pengaturan. Pengaturan ini bersifat independen dan dapat digunakan dalam kombinasi apa pun. Setiap pengaturan dapat diterapkan ke titik akses, bucket, atau seluruh Akun AWS. Jika pengaturan blokir akses publik untuk titik akses, bucket, atau akun berbeda, maka Amazon S3 akan menerapkan kombinasi yang paling ketat dari pengaturan titik akses, bucket, dan akun.

Ketika Amazon S3 mengevaluasi apakah suatu operasi dilarang oleh pengaturan blokir akses publik, itu akan menolak permintaan apa pun yang melanggar titik akses, bucket, atau pengaturan akun.

Important

Akses publik diberikan ke bucket dan objek melalui daftar kontrol akses (ACLs), kebijakan titik akses, kebijakan bucket, atau semua. Untuk membantu memastikan bahwa semua titik akses Amazon S3, bucket, dan objek Anda diblokir akses publiknya, kami menyarankan Anda untuk mengaktifkan keempat pengaturan guna memblokir akses publik untuk akun Anda. Pengaturan ini memblokir akses publik untuk semua bucket dan titik akses saat ini dan mendatang.

Sebelum menerapkan pengaturan ini, verifikasi bahwa aplikasi Anda akan bekerja dengan baik tanpa akses publik. Jika Anda memerlukan beberapa tingkat akses publik ke bucket atau objek Anda—misalnya, untuk menjadi host situs web statis seperti yang dijelaskan di Hosting situs web statis menggunakan Amazon S3—Anda dapat menyesuaikan pengaturan individual agar sesuai dengan kasus penggunaan penyimpanan Anda.

Mengaktifkan Blokir Akses Publik membantu melindungi sumber daya Anda dengan mencegah akses publik diberikan melalui kebijakan sumber daya atau daftar kontrol akses (ACLs) yang secara langsung dilampirkan ke sumber daya S3. Selain mengaktifkan Blokir Akses Publik, periksa kebijakan berikut dengan cermat untuk mengonfirmasi bahwa kebijakan tersebut tidak memberikan akses publik:

- Kebijakan berbasis identitas yang dilampirkan pada AWS prinsipal terkait (misalnya, peran IAM)

- Kebijakan berbasis sumber daya yang dilampirkan pada AWS sumber daya terkait (misalnya, kunci (KMS AWS Key Management Service))

> **Note**
>
> - Anda dapat mengaktifkan pengaturan blokir akses publik hanya untuk titik akses, bucket, dan Akun AWS. Amazon S3 tidak mendukung pengaturan blokir akses publik secara per-objek.
> - Saat Anda menerapkan pengaturan blokir akses publik ke akun, pengaturan berlaku untuk semua Wilayah AWS secara global. Pengaturan tersebut mungkin tidak berlaku di semua Wilayah secara langsung atau bersamaan, tetapi pada akhirnya akan menyebar ke semua Wilayah.

Topik

- Pengaturan blokir akses publik
- Melakukan operasi akses publik blok pada titik akses
- Arti “publik”
- Menggunakan IAM Access Analyzer untuk S3 untuk meninjau bucket publik
- Izin
- Mengonfigurasi blokir akses publik
- Mengonfigurasi pengaturan blokir akses publik untuk akun Anda
- Mengonfigurasi pengaturan blokir akses publik untuk bucket S3 Anda

Pengaturan blokir akses publik

S3 Blokir Akses Publik menyediakan empat pengaturan. Anda dapat menerapkan pengaturan ini dalam kombinasi apa pun ke titik akses individu, bucket, atau seluruh Akun AWS. Jika Anda menerapkan pengaturan tersebut ke sebuah akun, pengaturan tersebut akan berlaku untuk semua bucket dan titik akses yang dimiliki oleh akun tersebut. Demikian pula, jika Anda menerapkan pengaturan tersebut pada bucket, maka pengaturan tersebut berlaku untuk semua titik akses yang terkait dengan bucket tersebut.

Tabel berikut berisi pengaturan-pengaturan yang tersedia.

Nama	Deskripsi
`BlockPublicAcls`	Mengatur opsi ini menjadi `TRUE` akan menyebabkan perilaku berikut ini: • `PutBucketAcl` dan `PutObjectAcl` panggilan gagal jika daftar kontrol akses yang ditentukan (ACL) bersifat publik. • `PutObject` panggilan gagal jika permintaan menyertakan ACL publik. • Jika pengaturan ini diterapkan ke akun, maka `PUT Bucket` panggilan gagal jika permintaan menyertakan ACL publik. Saat pengaturan ini disetel ke`TRUE`, operasi yang ditentukan gagal (baik dilakukan melalui REST API, AWS CLI, atau AWS SDKs). Namun, kebijakan yang ada dan ACLs untuk bucket dan objek tidak dimodifikasi. Pengaturan ini memungkinkan Anda untuk melindungi dari akses publik sekaligus memungkinkan Anda untuk mengaudit, menyempurnakan, atau mengubah kebijakan dan ACLs yang sudah ada untuk ember dan objek Anda. **Note** Titik akses tidak ACLs terkait dengan mereka. Jika Anda menerapkan pengaturan ini ke titik akses, maka pengaturan bertindak sebagai jalur ke bucket yang mendasarinya. Jika titik akses mengaktifkan pengaturan ini, maka permintaan yang dibuat melalui titik akses tersebut akan tetap berfungsi seolah-olah bucket yang mendasari akan mengaktifkan pengaturan ini, terlepas dari apakah bucket benar-benar mengaktifkan pengaturan tersebut atau tidak.
`IgnorePublicAcls`	Menyetel opsi ini untuk `TRUE` menyebabkan Amazon S3 mengabaikan semua publik ACLs di ember dan objek apa pun yang dikandungnya. Pengaturan ini memungkinkan Anda untuk memblokir akses publik yang diberikan dengan aman ACLs sambil tetap mengizinkan `PutObject` panggilan yang menyertakan ACL publik (sebagai lawan dari`BlockPublicAcls` , yang menolak `PutObject` panggilan yang menyertakan ACL

Nama	Deskripsi
	publik). Mengaktifkan pengaturan ini tidak memengaruhi persistensi yang ada ACLs dan tidak ACLs mencegah publik baru disetel. **Note** Titik akses tidak ACLs terkait dengan mereka. Jika Anda menerapkan pengaturan ini ke titik akses, maka pengaturan bertindak sebagai jalur ke bucket yang mendasarinya. Jika titik akses mengaktifkan pengaturan ini, maka permintaan yang dibuat melalui titik akses tersebut akan tetap berfungsi seolah-olah bucket yang mendasari akan mengaktifkan pengaturan ini, terlepas dari apakah bucket benar-benar mengaktifkan pengaturan tersebut atau tidak.

Nama	Deskripsi
`BlockPublicPolicy`	Menyetel opsi ini ke `TRUE` bucket menyebabkan Amazon S3 menolak panggilan `PutBucketPolicy` jika kebijakan bucket yang ditentukan mengizinkan akses publik. Menyetel opsi ini ke `TRUE` bucket juga menyebabkan Amazon S3 menolak panggilan ke `PutAccessPointPolicy` semua titik akses akun yang sama pada bucket jika kebijakan yang ditentukan mengizinkan akses publik. Menyetel opsi ini ke `TRUE` titik akses menyebabkan Amazon S3 menolak panggilan ke `PutAccessPointPolicy` dan `PutBucketPolicy` yang dilakukan melalui titik akses jika kebijakan yang ditentukan (baik untuk titik akses atau bucket yang mendasarinya) mengizinkan akses publik. Anda dapat menggunakan pengaturan ini untuk memungkinkan pengguna mengelola kebijakan titik akses dan bucket tanpa mengizinkan mereka membagikan bucket atau objek yang dimuatnya secara publik. Mengaktifkan pengaturan ini tidak memengaruhi kebijakan titik akses atau bucket yang sudah ada. **Important** Untuk menggunakan pengaturan ini secara efektif, kami menyarankan Anda menerapkannya di level akun. Kebijakan bucket dapat memungkinkan pengguna untuk mengubah pengaturan blokir akses publik bucket. Oleh karena itu, pengguna yang memiliki izin untuk mengubah kebijakan bucket dapat memasukkan kebijakan yang memungkinkan mereka menonaktifkan pengaturan blokir akses publik untuk bucket tersebut. Jika pengaturan ini diaktifkan untuk seluruh akun, bukannya untuk bucket tertentu, Amazon S3 memblokir kebijakan publik bahkan jika pengguna mengubah kebijakan bucket untuk menonaktifkan pengaturan ini.

Nama	Deskripsi
`RestrictPublicBuckets`	Menyetel opsi ini untuk `TRUE` membatasi akses ke titik akses atau bucket dengan kebijakan publik hanya untuk prinsipal AWS layanan dan pengguna resmi dalam akun pemilik bucket dan akun pemilik jalur akses. Pengaturan ini memblokir semua akses lintas akun ke titik akses atau bucket (kecuali oleh prinsipal AWS layanan), sambil tetap memungkinkan pengguna di dalam akun untuk mengelola titik akses atau bucket. Mengaktifkan pengaturan ini tidak memengaruhi kebijakan titik akses atau bucket yang sudah ada, kecuali bahwa Amazon S3 memblokir akses publik dan akses lintas akun yang berasal dari setiap titik akses publik atau kebijakan bucket, termasuk delegasi non-publik ke akun tertentu.

Important

- Panggilan ke `GetBucketAcl` dan `GetObjectAcl` selalu mengembalikan izin efektif untuk bucket atau objek yang ditentukan. Sebagai contoh, anggaplah sebuah bucket memiliki ACL yang memberikan akses publik, tetapi bucket tersebut juga mengaktifkan pengaturan `IgnorePublicAcls`. Dalam hal ini, `GetBucketAcl` mengembalikan ACL yang mencerminkan izin akses yang diterapkan Amazon S3, bukan ACL aktual yang terkait dengan bucket.
- Blokir setelan akses publik tidak mengubah kebijakan yang ada atau ACLs. Oleh karena itu, menghapus pengaturan blokir akses publik akan menyebabkan bucket atau objek yang memiliki kebijakan publik atau ACL dapat diakses oleh publik lagi.

Melakukan operasi akses publik blok pada titik akses

Untuk melakukan memblokir operasi akses publik pada titik akses, gunakan AWS CLI layanan ini`s3control`.

> **Important**
>
> Anda tidak dapat mengubah pengaturan akses publik blok titik akses setelah membuat titik akses. Anda dapat menentukan blokir pengaturan akses publik untuk titik akses hanya saat membuat titik akses.

Arti “publik”

ACLs

Amazon S3 menganggap ACL ember atau objek bersifat publik jika memberikan izin kepada anggota dari `AllUsers` atau grup `AuthenticatedUsers` yang telah ditentukan sebelumnya. Untuk informasi lebih lanjut tentang grup yang telah ditentukan sebelumnya, lihat Grup Amazon S3 yang sudah ditentukan sebelumnya.

Kebijakan bucket

Saat mengevaluasi kebijakan bucket, Amazon S3 memulai dengan mengasumsikan bahwa kebijakan tersebut bersifat publik. Itu kemudian mengevaluasi kebijakan tersebut untuk menentukan apakah kebijakan tersebut memenuhi syarat sebagai non-publik. Agar dianggap non-publik, suatu kebijakan bucket harus memberikan akses hanya ke nilai-nilai tetap (nilai yang tidak mengandung wildcard atau Variabel AWS Identity and Access Management Kebijakan) untuk satu atau lebih berikut ini:

- AWS Prinsipal, pengguna, peran, atau prinsipal layanan (mis.`aws:PrincipalOrgID`)
- Satu set blok Classless Inter-Domain Routings (CIDR), menggunakan. `aws:SourceIp` Untuk informasi lebih lanjut tentang CIDR, lihat RFC 4632 di situs web Editor RFC.

  > **Note**
  >
  > Kebijakan bucket yang memberikan akses yang dikondisikan pada kunci kondisi `aws:SourceIp` dengan rentang IP yang sangat luas (misalnya, 0.0.0.0/1) dievaluasi sebagai “publik.” Ini termasuk nilai yang lebih luas daripada `/8` untuk IPv4 dan `/32` untuk IPv6 (tidak termasuk rentang RFC1918 pribadi). Blokir akses publik akan menolak kebijakan “publik” ini dan mencegah akses lintas akun ke bucket yang sudah menggunakan kebijakan “publik” ini.

- `aws:SourceArn`

- `aws:SourceVpc`
- `aws:SourceVpce`
- `aws:SourceOwner`
- `aws:SourceAccount`
- `aws:userid`, di luar dari pola `"AROLEID:*"`
- `s3:DataAccessPointArn`

> **Note**
>
> Saat digunakan dalam kebijakan bucket, nilai ini dapat berisi wildcard untuk nama titik akses tanpa membuat kebijakan publik, selama ID akun diperbaiki. Misalnya, memungkinkan akses ke `arn:aws:s3:us-west-2:123456789012:accesspoint/*` akan mengizinkan akses ke setiap titik akses yang terkait dengan akun `123456789012` di Wilayah `us-west-2`, tanpa membuat kebijakan bucket menjadi publik. Perilaku ini berbeda untuk kebijakan titik akses. Untuk informasi selengkapnya, lihat Titik Akses.

- `s3:DataAccessPointAccount`

Untuk informasi lebih lanjut tentang kebijakan bucket, lihat Kebijakan bucket untuk Amazon S3.

> **Note**
>
> Saat menggunakan kunci konteks multivalued, Anda harus menggunakan `ForAllValues` atau `ForAnyValue` mengatur operator.

Example : Kebijakan bucket publik

Menurut aturan ini, contoh kebijakan-kebijakan berikut ini dianggap publik.

```
{
  "Principal": "*",
  "Resource": "*",
  "Action": "s3:PutObject",
  "Effect": "Allow"
 }
```

```
{
  "Principal": "*",
  "Resource": "*",
  "Action": "s3:PutObject",
  "Effect": "Allow",
  "Condition": { "StringLike": {"aws:SourceVpc": "vpc-*"}}
 }
```

Anda dapat membuat kebijakan-kebijakan ini non-publik dengan memasukkan kunci kondisi yang tercantum sebelumnya, dengan menggunakan nilai tetap. Misalnya, Anda dapat membuat kebijakan terakhir sebelumnya menjadi non-publik dengan mengatur `aws:SourceVpc` dengan nilai tetap, seperti berikut ini.

```
{
  "Principal": "*",
  "Resource": "*",
  "Action": "s3:PutObject",
  "Effect": "Allow",
  "Condition": {"StringEquals": {"aws:SourceVpc": "vpc-91237329"}}
 }
```

Cara Amazon S3 mengevaluasi kebijakan bucket yang berisi pemberian akses publik dan non-publik

Contoh ini menunjukkan bagaimana Amazon S3 mengevaluasi kebijakan bucket yang berisi pemberian akses, baik publik maupun non-publik.

Anggaplah bucket memiliki kebijakan yang memberikan akses ke serangkaian pengguna utama tetap. Berdasarkan peraturan yang dijelaskan sebelumnya, kebijakan ini tidak boleh dibuat menjadi publik. Dengan demikian, jika Anda mengaktifkan pengaturan `RestrictPublicBuckets`, maka kebijakan tetap berlaku sebagaimana yang tertulis, karena `RestrictPublicBuckets` hanya berlaku untuk bucket yang memiliki kebijakan publik. Namun, jika Anda menambahkan pernyataan publik pada kebijakan tersebut, `RestrictPublicBuckets` berpengaruh pada bucket. Ini hanya memungkinkan kepala AWS layanan dan pengguna resmi akun pemilik bucket untuk mengakses bucket.

Sebagai contoh, anggaplah bahwa sebuah bucket yang dimiliki "Akun-1" memiliki kebijakan yang memuat hal-hal berikut:

1. Pernyataan yang memberikan akses ke AWS CloudTrail (yang merupakan prinsipal AWS layanan)

2. Pernyataan yang memberikan akses ke akun “Akun-2”
3. Pernyataan yang memberikan akses ke publik, misalnya dengan menentukan `"Principal": "*"` tanpa membatasi `Condition`

Kebijakan ini memenuhi kualifikasi sebagai kebijakan bersifat publik karena pernyataan ketiga. Dengan kebijakan ini diberlakukan dan `RestrictPublicBuckets` diaktifkan, Amazon S3 mengizinkan akses hanya dengan. CloudTrail Meskipun pernyataan 2 tidak bersifat publik, Amazon S3 nonaktifkan akses oleh "Akun-2." Hal ini karena pernyataan 3 menjadikan kebijakan tersebut bersifat publik, sehingga `RestrictPublicBuckets` berlaku. Oleh karena itu, Amazon S3 menonaktifkan akses lintas akun, meskipun kebijakan tersebut mendelegasikan akses ke akun tertentu, yakni “Akun-2.” Tetapi jika Anda menghapus pernyataan 3 dari kebijakan tersebut, maka kebijakan tersebut tidak akan memenuhi syarat sebagai kebijakan bersifat publik, dan `RestrictPublicBuckets` tidak lagi berlaku. Dengan demikian, “Akun-2” mendapatkan kembali akses ke bucket, bahkan jika Anda membiarkan `RestrictPublicBuckets` tetap aktif.

Titik Akses

Amazon S3 mengevaluasi pengaturan blokir akses publik dengan sedikit berbeda untuk titik akses dibandingkan dengan bucket. Aturan yang diterapkan Amazon S3 untuk menentukan kapan kebijakan titik akses publik biasanya sama dengan titik akses untuk bucket, kecuali dalam situasi berikut:

- Titik akses yang memiliki asal jaringan VPC selalu dianggap sebagai non-publik, terlepas dari konten kebijakan titik aksesnya.
- Kebijakan titik akses yang memberikan akses ke serangkaian titik akses dengan menggunakan `s3:DataAccessPointArn` dianggap bersifat publik. Perhatikan bahwa perilaku ini berbeda dengan perilaku untuk kebijakan bucket. Misalnya, kebijakan bucket yang memberikan akses ke nilai `s3:DataAccessPointArn` yang sesuai dengan `arn:aws:s3:us-west-2:123456789012:accesspoint/*` tidak dianggap publik. Namun demikian, pernyataan yang sama dalam kebijakan titik akses akan menjadikan titik akses tersebut menjadi publik.

Menggunakan IAM Access Analyzer untuk S3 untuk meninjau bucket publik

Anda dapat menggunakan IAM Access Analyzer for S3 untuk meninjau bucket dengan kebijakan bucket ACLs, bucket, atau access point yang memberikan akses publik. IAM Access Analyzer for S3 memberi tahu Anda tentang bucket yang dikonfigurasi untuk memungkinkan akses ke siapa pun di internet atau lainnya Akun AWS, termasuk Akun AWS di luar organisasi Anda. Untuk setiap bucket

publik atau bucket bersama, Anda akan menerima temuan yang melaporkan sumber dan tingkat akses publik atau akses bersama.

Pada Penganalisis Akses untuk S3, Anda dapat memblokir semua akses publik ke bucket dengan sekali klik. Anda juga dapat menelusuri pengaturan izin tingkat bucket untuk mengonfigurasi tingkat akses granular. Untuk kasus penggunaan tertentu dan terverifikasi yang memerlukan akses publik atau akses bersama, Anda dapat menyatakan dan mencatat maksud Anda untuk bucket agar tetap sebagai bucket dengan akses publik atau akses bersama dengan mengarsipkan temuan untuk bucket tersebut.

Dalam kasus yang jarang terjadi, IAM Access Analyzer untuk S3 dan Amazon S3 memblokir evaluasi akses publik mungkin berbeda pada apakah bucket bersifat publik. Perilaku ini terjadi karena Amazon S3 memblokir akses publik melakukan validasi pada keberadaan tindakan selain mengevaluasi akses publik. Misalkan kebijakan bucket berisi `Action` pernyataan yang mengizinkan akses publik untuk tindakan yang tidak didukung oleh Amazon S3 (misalnya,`s3:NotASupportedAction`). Dalam hal ini, Amazon S3 memblokir akses publik mengevaluasi bucket sebagai publik karena pernyataan seperti itu berpotensi membuat bucket publik jika tindakan tersebut kemudian didukung. Jika Amazon S3 memblokir akses publik dan IAM Access Analyzer untuk S3 berbeda dalam evaluasinya, kami sarankan untuk meninjau kebijakan bucket dan menghapus tindakan yang tidak didukung.

Untuk informasi selengkapnya tentang IAM Access Analyzer untuk Amazon S3, lihat Meninjau akses bucket menggunakan IAM Access Analyzer untuk S3.

Izin

Untuk menggunakan fitur-fitur yang dimiliki Blokir Akses Publik Amazon S3, Anda harus memiliki izin-izin berikut.

Operasi	Izin yang diperlukan
`GET`status kebijakan bucket	`s3:GetBucketPolicyStatus`
`GET`bucket Blokir pengaturan Akses Publik	`s3:GetBucketPublicAccessBlock`
`PUT`bucket Blokir pengaturan Akses Publik	`s3:PutBucketPublicAccessBlock`
`DELETE`bucket Blokir pengaturan Akses Publik	`s3:PutBucketPublicAccessBlock`
`GET`akun Blokir pengaturan Akses Publik	`s3:GetAccountPublicAccessBlock`

Operasi	Izin yang diperlukan
PUTakun Blokir pengaturan Akses Publik	`s3:PutAccountPublicAccessBlock`
DELETEakun Blokir pengaturan Akses Publik	`s3:PutAccountPublicAccessBlock`
PUTtitik akses Blokir pengaturan Akses Publik	`s3:CreateAccessPoint`

> **Note**
>
> DELETEOperasi memerlukan izin yang sama dengan PUT operasi. Tidak ada izin terpisah untuk DELETE operasi.

Mengonfigurasi blokir akses publik

Untuk informasi selengkapnya tentang mengonfigurasi blokir akses publik untuk bucket Amazon S3, dan titik akses Anda, lihat topik berikut: Akun AWS

- Mengonfigurasi pengaturan blokir akses publik untuk akun Anda
- Mengonfigurasi pengaturan blokir akses publik untuk bucket S3 Anda
- Melakukan operasi akses publik blok pada titik akses

Mengonfigurasi pengaturan blokir akses publik untuk akun Anda

Fitur Blokir Akses Publik Amazon S3 menyediakan pengaturan untuk titik akses, bucket, dan akun untuk membantu Anda mengelola akses publik ke sumber daya Amazon S3. Secara default, bucket, titik akses, dan objek yang baru tidak mengizinkan akses publik.

Untuk informasi selengkapnya, lihat Melakukan blok akses publik ke penyimpanan Amazon S3 Anda.

> **Note**
>
> Pengaturan tingkat akun mengesampingkan pengaturan pada objek individual. Mengkonfigurasi akun Anda untuk memblokir akses publik akan mengesampingkan pengaturan akses publik yang dibuat ke objek individual dalam akun Anda.

Anda dapat menggunakan konsol S3,, AWS CLI AWS SDKs, dan REST API untuk mengonfigurasi pengaturan blokir akses publik untuk semua bucket di akun Anda. Untuk informasi selengkapnya, lihat bagian di bawah ini.

Untuk mengonfigurasi pengaturan blokir akses publik untuk bucket Anda, lihat Mengonfigurasi pengaturan blokir akses publik untuk bucket S3 Anda. Untuk informasi lebih lanjut tentang titik akses, lihat Melakukan operasi akses publik blok pada titik akses.

Menggunakan konsol S3

Blokir akses publik Amazon S3 mencegah aplikasi dari pengaturan apa pun yang memungkinkan akses publik ke data dalam S3 bucket. Bagian ini menjelaskan cara mengedit pengaturan blokir akses publik untuk semua bucket S3 di Akun AWS Anda. Untuk informasi lebih lanjut tentang pemblokiran akses publik, lihat Melakukan blok akses publik ke penyimpanan Amazon S3 Anda.

Untuk mengedit blokir setelan akses publik untuk semua bucket S3 di Akun AWS

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Pilih Blokir pengaturan Akses Publik untuk akun ini.
3. Memilih Edit untuk mengubah pengaturan blokir akses publik untuk semua bucket di Akun AWS Anda.
4. Pilih pengaturan yang ingin Anda ubah, lalu pilih Simpan perubahan.
5. Saat Anda diminta untuk mengonfirmasi, masukkan **`confirm`**. Kemudian, pilih Konfirmasi untuk menyimpan perubahan Anda.

Menggunakan AWS CLI

Anda dapat menggunakan Blokir Akses Publik Amazon S3 melalui AWS CLI. Untuk informasi selengkapnya tentang pengaturan dan penggunaan AWS CLI, lihat Apa itu AWS Command Line Interface?

Akun

Untuk melakukan operasi blokir akses publik pada akun, gunakan AWS CLI layanan `s3control`. Operasi tingkat akun yang menggunakan layanan ini adalah sebagai berikut:

- `PutPublicAccessBlock`(untuk akun)
- `GetPublicAccessBlock`(untuk akun)

- `DeletePublicAccessBlock`(untuk akun)

Untuk informasi dan contoh tambahan, lihat put-public-access-blockdi AWS CLI Referensi.

Menggunakan AWS SDKs

Java

Contoh berikut menunjukkan cara menggunakan Amazon S3 Block Public Access dengan AWS SDK untuk Java untuk menempatkan konfigurasi blok akses publik pada akun Amazon S3.

```
AWSS3ControlClientBuilder controlClientBuilder =
 AWSS3ControlClientBuilder.standard();
controlClientBuilder.setRegion(<region>);
controlClientBuilder.setCredentials(<credentials>);

AWSS3Control client = controlClientBuilder.build();
client.putPublicAccessBlock(new PutPublicAccessBlockRequest()
  .withAccountId(<account-id>)
  .withPublicAccessBlockConfiguration(new PublicAccessBlockConfiguration()
    .withIgnorePublicAcls(<value>)
    .withBlockPublicAcls(<value>)
    .withBlockPublicPolicy(<value>)
    .withRestrictPublicBuckets(<value>)));
```

Important

Contoh ini hanya berkaitan dengan operasi tingkat akun, yang menggunakan kelas pelanggan `AWSS3Control`. Untuk operasi tingkat bucket, lihat contoh sebelumnya.

Other SDKs

Untuk informasi tentang menggunakan yang lain AWS SDKs, lihat Mengembangkan dengan Amazon S3 menggunakan Referensi API Amazon S3 AWS SDKs di Amazon S3.

Penggunaan API REST

Untuk informasi tentang menggunakan Amazon S3 Blokir Akses Publik melalui REST APIs, lihat topik berikut di Referensi API Amazon Simple Storage Service.

- Operasi tingkat akun
  - PutPublicAccessBlock
  - GetPublicAccessBlock
  - DeletePublicAccessBlock

Mengonfigurasi pengaturan blokir akses publik untuk bucket S3 Anda

Fitur Blokir Akses Publik Amazon S3 menyediakan pengaturan untuk titik akses, bucket, dan akun untuk membantu Anda mengelola akses publik ke sumber daya Amazon S3. Secara default, bucket, titik akses, dan objek yang baru tidak mengizinkan akses publik.

Untuk informasi selengkapnya, lihat Melakukan blok akses publik ke penyimpanan Amazon S3 Anda.

Anda dapat menggunakan konsol S3,, AWS CLI AWS SDKs, dan REST API untuk memberikan akses publik ke satu atau beberapa bucket. Anda juga dapat memblokir akses publik ke bucket yang sudah publik. Untuk informasi selengkapnya, lihat bagian di bawah ini.

Untuk mengonfigurasi pengaturan blokir akses publik untuk setiap bucket di akun Anda, lihat Mengonfigurasi pengaturan blokir akses publik untuk akun Anda. Untuk informasi tentang konfigurasi blokir akses publik untuk titik akses, lihat Melakukan operasi akses publik blok pada titik akses.

Menggunakan konsol S3

Blokir akses publik Amazon S3 mencegah aplikasi dari pengaturan apa pun yang memungkinkan akses publik ke data dalam S3 bucket. Bagian ini menjelaskan cara mengedit pengaturan Blokir Akses Publik untuk satu bucket S3 atau lebih. Untuk informasi tentang memblokir akses publik menggunakan AWS CLI, AWS SDKs, dan Amazon S3 REST APIs, lihat. Melakukan blok akses publik ke penyimpanan Amazon S3 Anda

Anda dapat melihat apakah bucket dapat diakses publik dari daftar Bucket, di kolom IAM Access Analyzer. Untuk informasi selengkapnya, lihat Meninjau akses bucket menggunakan IAM Access Analyzer untuk S3.

Jika Anda melihat `Error` saat Anda mencantumkan bucket dan pengaturan akses publiknya, Anda mungkin tidak memiliki izin yang diperlukan. Periksa untuk memastikan Anda memiliki izin berikut yang ditambahkan ke pengguna atau kebijakan peran Anda:

```
s3:GetAccountPublicAccessBlock
s3:GetBucketPublicAccessBlock
s3:GetBucketPolicyStatus
s3:GetBucketLocation
s3:GetBucketAcl
s3:ListAccessPoints
s3:ListAllMyBuckets
```

Dalam beberapa kasus yang jarang terjadi, permintaan juga dapat gagal karena gangguan Wilayah AWS .

Untuk mengedit pengaturan blokir akses publik Amazon S3 untuk satu bucket S3

Ikuti langkah-langkah ini jika Anda perlu mengubah pengaturan akses publik untuk bucket S3 tunggal.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di daftar Bucket, memilih nama bucket yang Anda inginkan.
3. Pilih Izin.
4. Pilih Edit di samping Blokir akses publik (pengaturan bucket) untuk mengubah pengaturan akses publik untuk bucket. Untuk informasi selengkapnya tentang empat Peraturan Blokir Akses Publik Amazon S3, lihat Pengaturan blokir akses publik.
5. Pilih salah satu pengaturan, lalu pilih Simpan perubahan.
6. Saat Anda diminta untuk mengonfirmasi, masukkan **`confirm`**. Kemudian, pilih Konfirmasi untuk menyimpan perubahan Anda.

Anda juga dapat mengubah setelan Akses Publik Blok Amazon S3 saat membuat bucket. Untuk informasi selengkapnya, lihat Membuat ember tujuan umum.

Menggunakan AWS CLI

Untuk memblokir akses publik pada bucket atau untuk menghapus blok akses publik, gunakan AWS CLI layanan ini`s3api`. Operasi tingkat bucket yang menggunakan layanan ini adalah sebagai berikut:

- `PutPublicAccessBlock`(untuk ember)
- `GetPublicAccessBlock`(untuk ember)
- `DeletePublicAccessBlock`(untuk ember)
- `GetBucketPolicyStatus`

Untuk informasi dan contoh selengkapnya, lihat put-public-access-block di Referensi AWS CLI .

Menggunakan AWS SDKs

Java

```
AmazonS3 client = AmazonS3ClientBuilder.standard()
   .withCredentials(<credentials>)
   .build();

client.setPublicAccessBlock(new SetPublicAccessBlockRequest()
  .withBucketName(<bucket-name>)
  .withPublicAccessBlockConfiguration(new PublicAccessBlockConfiguration()
    .withBlockPublicAcls(<value>)
    .withIgnorePublicAcls(<value>)
    .withBlockPublicPolicy(<value>)
    .withRestrictPublicBuckets(<value>)));
```

Important

Contoh ini hanya berkaitan dengan operasi tingkat bucket, yang menggunakan kelas pelanggan `AmazonS3`. Untuk operasi tingkat akun, lihat contoh berikut.

Other SDKs

Untuk informasi tentang menggunakan yang lain AWS SDKs, lihat Mengembangkan dengan Amazon S3 menggunakan Referensi API Amazon S3 AWS SDKs di Amazon S3.

Penggunaan API REST

Untuk informasi tentang menggunakan Amazon S3 Blokir Akses Publik melalui REST APIs, lihat topik berikut di Referensi API Amazon Simple Storage Service.

- Operasi tingkat bucket
  - PutPublicAccessBlock
  - GetPublicAccessBlock
  - DeletePublicAccessBlock
  - GetBucketPolicyStatus

Meninjau akses bucket menggunakan IAM Access Analyzer untuk S3

IAM Access Analyzer for S3 memberi tahu Anda tentang bucket S3 yang dikonfigurasi untuk memungkinkan akses ke siapa pun di internet atau lainnya Akun AWS, termasuk di luar organisasi Anda. Akun AWS Untuk setiap bucket publik atau kelompok bersama, Anda menerima temuan ke dalam sumber dan tingkat akses publik atau bersama. Misalnya, IAM Access Analyzer untuk S3 mungkin menunjukkan bahwa suatu bucket telah membaca atau menulis akses yang disediakan melalui daftar kontrol akses (ACL) bucket, kebijakan titik akses Multi-Wilayah, atau kebijakan titik akses. Dengan temuan ini, Anda dapat segera mengambil tindakan korektif yang tepat untuk memulihkan akses bucket ke apa yang Anda inginkan.

Saat meninjau bucket berisiko dalam Penganalisis Akses IAM untuk S3, Anda dapat memblokir semua akses publik ke bucket dengan sekali klik. Kami menyarankan agar Anda memblokir semua akses ke bucket Anda, kecuali jika Anda memerlukan akses publik untuk mendukung kasus penggunaan tertentu. Sebelum memblokir semua akses publik, pastikan bahwa aplikasi Anda akan terus berfungsi dengan benar tanpa akses publik. Untuk informasi selengkapnya, lihat Melakukan blok akses publik ke penyimpanan Amazon S3 Anda.

Anda juga dapat menelusuri pengaturan izin tingkat bucket untuk mengonfigurasi tingkat akses granular. Untuk kasus penggunaan kustom dan terverifikasi yang memerlukan akses publik, seperti hosting situs web statis, unduhan publik, atau berbagi akun silang, Anda dapat mengenali dan mencatat maksud Anda agar bucket tetap publik atau berbagi dengan mengarsipkan temuan untuk bucket. Anda dapat mempertahankan dan memodifikasi konfigurasi bucket ini kapan saja. Anda juga dapat mengunduh temuan Anda dalam bentuk laporan CSV untuk keperluan audit.

IAM Access Analyzer untuk S3 tersedia tanpa biaya tambahan, yang berada di konsol Amazon S3. Penganalisis Akses IAM untuk S3 didukung oleh Penganalisis Akses IAM (IAM) AWS Identity and Access Management . Untuk menggunakan IAM Access Analyzer untuk S3 di konsol Amazon S3, Anda harus mengunjungi konsol IAM dan mengaktifkan IAM Access Analyzer pada basis Per-wilayah.

Untuk informasi selengkapnya tentang IAM Access Analyzer, lihat Apa itu IAM Access Analyzer? di Panduan Pengguna IAM. Untuk informasi lebih lanjut tentang IAM Access Analyzer untuk S3, tinjau bagian berikut.

Important

- IAM Access Analyzer untuk S3 membutuhkan penganalisis tingkat akun. Untuk menggunakan IAM Access Analyzer untuk S3, Anda harus mengunjungi IAM Access Analyzer dan membuat analyzer yang memiliki akun sebagai zona kepercayaan. Untuk informasi selengkapnya, lihat Mengaktifkan IAM Access Analyzer di Panduan Pengguna IAM.
- IAM Access Analyzer untuk S3 tidak menganalisis kebijakan titik akses yang dilampirkan ke titik akses lintas akun. Perilaku ini terjadi karena titik akses dan kebijakannya berada di luar zona kepercayaan, yaitu akun. Bucket yang mendelegasikan akses ke titik akses lintas akun tercantum di bawah Bucket dengan akses publik jika Anda belum menerapkan pengaturan blokir akses publik `RestrictPublicBuckets` ke bucket atau akun. Saat Anda menerapkan setelan `RestrictPublicBuckets` blokir akses publik, bucket dilaporkan di bawah Bucket dengan akses dari orang lain Akun AWS — termasuk pihak ketiga Akun AWS.
- Ketika kebijakan bucket atau ACL bucket ditambahkan atau diubah, IAM Access Analyzer menghasilkan dan memperbarui temuan berdasarkan perubahan dalam waktu 30 menit. Temuan terkait dengan pengaturan akses publik blok tingkat akun tidak dapat dibuat atau diperbarui hingga 6 jam setelah Anda mengubah pengaturan. Temuan yang terkait dengan Titik Akses Multi-Wilayah mungkin tidak dibuat atau diperbarui hingga enam jam setelah Titik Akses Multi-Wilayah dibuat, dihapus, atau Anda mengubah kebijakannya.

Topik

- Informasi apa yang disediakan IAM Access Analyzer untuk S3?
- Mengaktifkan IAM Access Analyzer untuk S3

- Memblokir semua akses publik
- Meninjau dan mengubah akses bucket
- Mengarsipkan temuan bucket
- Mengaktifkan temuan bucket yang diarsipkan
- Melihat detail temuan
- Mengunduh IAM Access Analyzer untuk laporan S3

Informasi apa yang disediakan IAM Access Analyzer untuk S3?

Penganalisis Akses untuk S3 menyediakan temuan untuk bucket yang dapat diakses di luar Akun AWS Anda. Bucket yang terdaftar Bucket dengan akses publik dapat diakses oleh siapa pun di internet. Jika Access Analyzer untuk S3 mengidentifikasi bucket publik, Anda juga melihat peringatan di bagian atas halaman yang menunjukkan jumlah bucket publik di Wilayah Anda. Bucket yang terdaftar di bawah Bucket dengan akses dari orang lain Akun AWS — termasuk pihak ketiga Akun AWS dibagikan secara kondisional dengan orang lain Akun AWS, termasuk akun di luar organisasi Anda.

Untuk setiap bucket, IAM Access Analyzer untuk S3 memberikan informasi berikut:

- Nama Bucket
- Ditemukan oleh Access Analyzer - Saat IAM Access Analyzer untuk S3 menemukan akses bucket publik atau bersama.
- Dibagikan melalui - Cara bucket dibagikan—melalui kebijakan bucket, ACL bucket, kebijakan Titik Akses Multi-Wilayah, atau kebijakan titik akses. Titik Akses Multi-Wilayah dan titik akses lintas akun ditampilkan di bawah titik akses. Bucket dapat dibagikan melalui kebijakan dan ACLs. Jika ingin menemukan dan meninjau sumber untuk akses bucket Anda, Anda dapat menggunakan informasi dalam kolom ini sebagai titik awal untuk mengambil tindakan korektif yang cepat dan tepat.
- Status - Status temuan bucket. IAM Access Analyzer untuk S3 menampilkan temuan untuk semua bucket publik dan bersama.
  - Aktif - Temuan belum ditinjau.
  - Diarsipkan - Temuan telah ditinjau dan dikonfirmasi sesuai tujuan.
  - Semua - Semua temuan untuk ember yang bersifat publik atau dibagikan dengan orang lain Akun AWS, termasuk Akun AWS di luar organisasi Anda.
- Tingkat akses - Izin akses diberikan untuk bucket:

- Daftar - Cantumkan sumber daya.
- Baca - Membaca, tetapi tidak mengedit konten dan atribut sumber daya.
- Tulis - Membuat, menghapus, atau memodifikasi sumber daya.
- Izin - Berikan atau ubah izin sumber daya.
- Penandaan - Perbarui tag yang terkait dengan sumber daya.

Mengaktifkan IAM Access Analyzer untuk S3

Untuk menggunakan IAM Access Analyzer untuk S3, Anda harus menyelesaikan langkah prasyarat berikut.

1. Berikan izin yang diperlukan.

   Untuk informasi lebih lanjut, lihat Izin yang Diperlukan untuk menggunakan Access Analyzer dalam Panduan Pengguna IAM.
2. Kunjungi IAM untuk membuat penganalisis level akun untuk setiap Wilayah tempat Anda ingin menggunakan Access Analyzer.

   IAM Access Analyzer untuk S3 membutuhkan penganalisis tingkat akun. Untuk menggunakan Access Analyzer untuk S3, Anda harus membuat penganalisis yang memiliki akun sebagai zona kepercayaan. Untuk informasi selengkapnya, lihat Mengaktifkan IAM Access Analyzer di Panduan Pengguna IAM.

Memblokir semua akses publik

Jika Anda ingin memblokir semua akses ke bucket dalam satu klik, Anda dapat menggunakan Blokir semua akses publik dalam Access Analyzer untuk S3. Ketika Anda memblokir semua akses publik ke bucket, akses publik tidak diberikan. Kami menyarankan agar Anda memblokir semua akses publik ke bucket Anda kecuali jika Anda memerlukan akses publik untuk mendukung kasus penggunaan tertentu yang terverifikasi. Sebelum memblokir semua akses publik, pastikan bahwa aplikasi Anda akan terus berfungsi dengan benar tanpa akses publik.

Jika tidak ingin memblokir semua akses publik ke bucket Anda, Anda dapat mengedit pengaturan akses publik blok di konsol Amazon S3 untuk mengonfigurasi tingkat akses granular ke bucket Anda. Untuk informasi selengkapnya, lihat Melakukan blok akses publik ke penyimpanan Amazon S3 Anda.

Dalam kasus yang jarang terjadi, IAM Access Analyzer untuk S3 dan Amazon S3 memblokir evaluasi akses publik mungkin berbeda pada apakah bucket bersifat publik. Perilaku ini terjadi karena Amazon S3 memblokir akses publik melakukan validasi pada keberadaan tindakan selain mengevaluasi akses publik. Misalkan kebijakan bucket berisi `Action` pernyataan yang mengizinkan akses publik untuk tindakan yang tidak didukung oleh Amazon S3 (misalnya,`s3:NotASupportedAction`). Dalam hal ini, Amazon S3 memblokir akses publik mengevaluasi bucket sebagai publik karena pernyataan seperti itu berpotensi membuat bucket publik jika tindakan tersebut kemudian didukung. Jika Amazon S3 memblokir akses publik dan IAM Access Analyzer untuk S3 berbeda dalam evaluasinya, kami sarankan untuk meninjau kebijakan bucket dan menghapus tindakan yang tidak didukung.

Untuk memblokir semua akses publik ke bucket menggunakan Access Analyzer IAM untuk S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi di sebelah kiri, di bawah Dasbor, pilih Access analyzer untuk S3.
3. Di IAM Access Analyzer untuk S3, pilih bucket.
4. Pilih Blokir semua akses publik.
5. Untuk mengonfirmasi maksud Anda untuk memblokir semua akses publik ke bucket, di Blokir semua akses publik (pengaturan bucket), masukkan **`confirm`**.

   Amazon S3 memblokir semua akses publik ke bucket Anda. Status pembaruan temuan bucket untuk diselesaikan, dan bucket menghilang dari daftar Access Analyzer untuk S3. Jika Anda ingin meninjau bucket yang diselesaikan, buka IAM Access Analyzer di Konsol IAM.

Meninjau dan mengubah akses bucket

Jika Anda tidak bermaksud memberikan akses ke publik atau lainnya Akun AWS, termasuk akun di luar organisasi, Anda dapat mengubah ACL bucket, kebijakan bucket, kebijakan Titik Akses Multi-Wilayah, atau kebijakan jalur akses untuk menghapus akses ke bucket. Kolom Dibagikan melalui menampilkan semua sumber akses bucket: kebijakan bucket, bucket ACL, kebijakan titik and/or akses. Titik Akses Multi-Wilayah dan titik akses lintas akun ditampilkan di bawah titik akses.

Untuk meninjau dan mengubah kebijakan bucket, bucket ACL, Titik Akses Multi-Wilayah, atau kebijakan titik akses

1. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi, pilih Access analyzer untuk S3.

3. Untuk melihat apakah akses publik atau akses bersama diberikan melalui kebijakan bucket, kebijakan Titik Akses Multi-Wilayah, atau kebijakan titik akses, lihat di kolom Dibagikan melalui.
4. Di bagian bawah Bucket, pilih nama bucket dengan kebijakan bucket, kebijakan bucket ACL, atau kebijakan titik akses yang ingin Anda ubah atau tinjau.
5. Jika Anda ingin mengubah atau melihat bucket ACL:

   a. Pilih Izin.

   b. Pilih Daftar Kontrol Akses.

   c. Tinjau bucket ACL Anda, dan buat perubahan sesuai kebutuhan.

   Untuk informasi selengkapnya, lihat Mengkonfigurasi ACLs.
6. Jika Anda ingin mengubah atau meninjau kebijakan bucket:

   a. Pilih Izin.

   b. Pilih Kebijakan Bucket.

   c. Tinjau atau ubah kebijakan bucket sesuai kebutuhan.

   Untuk informasi selengkapnya, lihat Menambahkan kebijakan bucket dengan menggunakan konsol Amazon S3.
7. Jika Anda ingin mengubah atau melihat kebijakan Titik Akses Multi-Wilayah:

   a. Pilih Titik Akses Multi-Wilayah.

   b. Pilih nama Titik Akses Multi-Wilayah.

   c. Tinjau atau ubah kebijakan Titik Akses Multi-Wilayah sesuai kebutuhan.

   Untuk informasi selengkapnya, lihat Izin.
8. Jika Anda ingin meninjau atau mengubah kebijakan titik akses:

   a. Memilih titik akses.

   b. Pilih nama titik akses.

   c. Tinjau atau ubah akses sesuai kebutuhan.

   Untuk informasi selengkapnya, lihat Mengelola jalur akses Amazon S3 Anda untuk bucket tujuan umum.

Jika Anda mengedit atau menghapus Bacbu ACL, kebijakan bucket, atau kebijakan titik akses untuk menghapus akses publik atau akses bersama, status pembaruan temuan bucket untuk diselesaikan. Temuan bucket yang diselesaikan menghilang dari IAM Access Analyzer untuk daftar S3, tetapi Anda dapat melihatnya di IAM Access Analyzer.

Mengarsipkan temuan bucket

Jika bucket memberikan akses ke publik atau lainnya Akun AWS, termasuk akun di luar organisasi Anda, untuk mendukung kasus penggunaan tertentu (misalnya, situs web statis, unduhan publik, atau berbagi lintas akun), Anda dapat mengarsipkan temuan untuk bucket. Saat Anda mengarsipkan temuan bucket, Anda menyatakan dan mencatat maksud Anda agar bucket tetap bersifat publik, atau berbagi. Temuan bucket yang diarsipkan tetap ada dalam daftar Access Analyzer untuk S3 sehingga Anda selalu mengetahui bucket mana yang publik atau dibagikan.

Untuk mengarsipkan temuan bucket dalam IAM Access Analyzer untuk S3

1. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi, pilih Access analyzer untuk S3.
3. Access Analyzer IAM untuk S3, pilih bucket aktif.
4. Untuk mengetahui maksud Anda agar bucket ini dapat diakses oleh publik atau lainnya Akun AWS, termasuk akun di luar organisasi Anda, pilih Arsip.
5. Masukkan **`confirm`**, dan pilih Arsipkan.

Mengaktifkan temuan bucket yang diarsipkan

Setelah mengarsipkan temuan, Anda dapat selalu mempertahankan temuan tersebut dan mengubah status kembali menjadi aktif, yang menunjukkan bahwa bucket memerlukan peninjauan lain.

Untuk mengaktifkan temuan bucket yang diarsipkan di IAM Access Analyzer untuk S3

1. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi, pilih Access analyzer untuk S3.
3. Pilih bucket findings yang diarsipkan.
4. Pilih Tandai sebagai aktif.

Melihat detail temuan

Jika Anda perlu melihat informasi selengkapnya tentang bucket, Anda dapat membuka bucket untuk menemukan detail di IAM Access Analyzer di Konsol IAM.

Untuk melihat detail temuan dalam IAM Access Analyzer untuk S3

1. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi, pilih Access analyzer untuk S3.
3. Di IAM Access Analyzer untuk S3, pilih bucket.
4. Pilih Lihat detail.

   Detail temuan terbuka di IAM Access Analyzer di Konsol IAM.

Mengunduh IAM Access Analyzer untuk laporan S3

Anda dapat mengunduh temuan bucket Anda sebagai laporan CSV yang dapat Anda gunakan untuk tujuan audit. Laporan mencakup informasi yang sama yang Anda lihat dalam Access Analyzer untuk S3 pada konsol Amazon S3.

Untuk mengunduh laporan

1. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi di sebelah kiri, pilih Access analyzer untuk S3.
3. Pada filter Area, pilih Area.

   Access Analyzer IAM untuk pembaruan S3 untuk menunjukkan bucket untuk Area yang dipilih.
4. Pilih Unduh laporan.

   Laporan CSV dibuat dan disimpan ke komputer Anda.

Memverifikasi kepemilikan bucket dengan syarat pemilik bucket

Kondisi pemilik bucket Amazon S3 memastikan bahwa bucket yang Anda gunakan dalam operasi S3 adalah milik yang Anda harapkan. Akun AWS

Sebagian besar operasi S3 membaca atau menulis ke bucket S3 tertentu. Operasi ini meliputi pengunggahan, penyalinan, dan pengunduhan objek, pengambilan atau modifikasi konfigurasi

bucket, dan pengambilan atau modifikasi konfigurasi objek. Saat Anda melakukan operasi ini, Anda menentukan bucket yang ingin Anda gunakan dengan menyertakan nama dengan permintaan. Misalnya, untuk mengambil objek dari S3, Anda membuat permintaan yang menentukan nama bucket dan kunci objek untuk diambil dari bucket tersebut.

Karena Amazon S3 mengidentifikasi bucket berdasarkan namanya, aplikasi yang menggunakan nama bucket yang salah dalam permintaan dapat secara tidak sengaja melakukan operasi terhadap bucket yang berbeda dari yang diharapkan. Untuk membantu menghindari interaksi bucket yang tidak disengaja dalam situasi seperti ini, Anda dapat menggunakan ketentuan pemilik bucket. Syarat pemilik bucket mengaktifkan Anda memverifikasi bahwa bucket target dimiliki oleh Akun AWS yang diharapkan, sehingga memberikan lapisan jaminan tambahan bahwa operasi S3 Anda memiliki efek yang Anda inginkan.

Topik

- Kapan harus menggunakan ketentuan pemilik bucket
- Memverifikasi pemilik bucket
- Contoh
- Pembatasan dan batasan

Kapan harus menggunakan ketentuan pemilik bucket

Kami merekomendasikan penggunaan ketentuan pemilik bucket setiap kali Anda melakukan operasi S3 yang didukung dan mengetahui ID akun pemilik bucket yang diharapkan. Ketentuan pemilik bucket tersedia untuk semua operasi S3 Object dan sebagian besar operasi bucket S3. Untuk daftar operasi S3 yang tidak mendukung ketentuan pemilik bucket, lihat Pembatasan dan batasan.

Untuk melihat manfaat menggunakan kondisi pemilik bucket, pertimbangkan skenario berikut yang melibatkan AWS pelanggan Bea:

1. Bea mengembangkan aplikasi yang menggunakan Amazon S3. Selama pengembangan, Bea menggunakan pengujiannya hanya Akun AWS untuk membuat bucket bernama`bea-data-test`, dan mengonfigurasi aplikasinya untuk membuat permintaan. `bea-data-test`
2. Bea men-deploy aplikasinya, tetapi lupa mengonfigurasi ulang aplikasi untuk menggunakan bucket dalam Akun AWS produksinya.
3. Dalam produksi, aplikasi Bea membuat permintaan ke `bea-data-test`, yang berhasil. Hal ini menyebabkan data produksi ditulis ke bucket dalam akun pengujian Bea.

Bea dapat membantu melindungi dari situasi seperti ini dengan menggunakan ketentuan pemilik bucket. Dengan kondisi pemilik bucket, Bea dapat menyertakan Akun AWS ID pemilik bucket yang diharapkan dalam permintaannya. Amazon S3 kemudian memeriksa ID akun pemilik bucket sebelum memproses setiap permintaan. Apabila pemilik bucket yang sesungguhnya tidak sesuai dengan pemilik bucket yang diharapkan, permintaan gagal.

Jika Bea menggunakan ketentuan pemilik bucket, skenario yang dijelaskan sebelumnya tidak akan mengakibatkan aplikasi Bea secara tidak sengaja menulis ke bucket pengujian. Sebagai gantinya, permintaan yang dibuat aplikasinya pada langkah 3 akan gagal dengan pesan kesalahan `Access Denied`. Dengan menggunakan syarat pemilik bucket, Bea membantu menghilangkan risiko interaksi yang tidak disengaja dengan bucket dalam Akun AWS yang salah.

Memverifikasi pemilik bucket

Untuk menggunakan ketentuan pemilik bucket, Anda menyertakan parameter dengan permintaan Anda yang menentukan pemilik bucket yang diharapkan. Sebagian besar operasi S3 hanya melibatkan satu bucket, dan hanya memerlukan satu parameter ini untuk menggunakan ketentuan pemilik bucket. Untuk operasi `CopyObject`, parameter pertama ini menentukan pemilik bucket tujuan yang diharapkan, dan Anda menyertakan parameter kedua untuk menentukan pemilik bucket sumber yang diharapkan.

Saat Anda membuat permintaan yang menyertakan parameter ketentuan pemilik bucket, S3 memeriksa ID akun pemilik bucket terhadap parameter yang ditentukan sebelum memproses permintaan. Jika parameter cocok dengan ID akun pemilik bucket, S3 akan memproses permintaan. Jika parameter tidak cocok dengan ID akun pemilik bucket, permintaan akan gagal dengan pesan kesalahan `Access Denied`.

Anda dapat menggunakan kondisi pemilik bucket dengan AWS Command Line Interface (AWS CLI) AWS SDKs, dan Amazon S3 REST. APIs Saat menggunakan kondisi pemilik bucket dengan Amazon S3 REST AWS CLI dan Amazon S3 APIs, gunakan nama parameter berikut.

Metode akses	Parameter untuk operasi non penyalinan	Parameter sumber operasi penyalinan	Parameter tujuan operasi penyalinan
AWS CLI	`--expected-bucket-owner`	`--expected-source-bucket-owner`	`--expected-bucket-owner`

Metode akses	Parameter untuk operasi non penyalinan	Parameter sumber operasi penyalinan	Parameter tujuan operasi penyalinan
Amazon S3 REST APIs	Header `x-amz-expected-bucket-owner`	Header `x-amz-source-expected-bucket-owner`	Header `x-amz-expected-bucket-owner`

Nama parameter yang diperlukan untuk menggunakan kondisi pemilik bucket dengan AWS SDKs bervariasi tergantung pada bahasa. Untuk menentukan parameter yang diperlukan, lihat dokumentasi SDK untuk bahasa yang Anda inginkan. Anda dapat menemukan dokumentasi SDK di Alat untuk Membangun di AWS.

Contoh

Contoh berikut menunjukkan bagaimana Anda dapat menerapkan kondisi pemilik bucket di Amazon S3 menggunakan AWS CLI atau. AWS SDK for Java 2.x

Example

Contoh: Mengunggah sebuah objek

Contoh berikut menunjukkan cara unggah sebuah objek ke bucket S3 *amzn-s3-demo-bucket1*, menggunakan syarat pemilik bucket untuk memastikan bahwa *amzn-s3-demo-bucket1* dimiliki oleh Akun AWS 111122223333.

AWS CLI

```
aws s3api put-object \
                --bucket amzn-s3-demo-bucket1 --key exampleobject --
body example_file.txt \
                --expected-bucket-owner 111122223333
```

AWS SDK for Java 2.x

```
public void putObjectExample() {
    S3Client s3Client = S3Client.create();;
    PutObjectRequest request = PutObjectRequest.builder()
            .bucket("amzn-s3-demo-bucket1")
```

```
            .key("exampleobject")
            .expectedBucketOwner("111122223333")
            .build();
    Path path = Paths.get("example_file.txt");
    s3Client.putObject(request, path);
}
```

Example

Contoh: Menyalin sebuah objek

Contoh berikut menyalin objek `object1` dari bucket S3 *amzn-s3-demo-bucket1* ke bucket S3 *amzn-s3-demo-bucket2*. Itu menggunakan syarat pemilik bucket untuk memastikan bahwa bucket dimiliki oleh akun yang diharapkan sesuai dengan tabel berikut.

Bucket	Pemilik yang diharapkan
amzn-s3-demo-bucket1	111122223333
amzn-s3-demo-bucket2	444455556666

AWS CLI

```
aws s3api copy-object --copy-source amzn-s3-demo-bucket1/object1 \
                          --bucket amzn-s3-demo-bucket2 --key object1copy \
                          --expected-source-bucket-owner 111122223333 --expected-
bucket-owner 444455556666
```

AWS SDK for Java 2.x

```
public void copyObjectExample() {
        S3Client s3Client = S3Client.create();
        CopyObjectRequest request = CopyObjectRequest.builder()
                .copySource("amzn-s3-demo-bucket1/object1")
                .destinationBucket("amzn-s3-demo-bucket2")
                .destinationKey("object1copy")
                .expectedSourceBucketOwner("111122223333")
                .expectedBucketOwner("444455556666")
                .build();
```

```
        s3Client.copyObject(request);
    }
```

Example

Contoh: Mengambil kebijakan bucket

Contoh berikut ini mengambil kebijakan akses untuk bucket S3 *amzn-s3-demo-bucket1*, menggunakan syarat pemilik bucket untuk memastikan bahwa *amzn-s3-demo-bucket1* dimiliki oleh Akun AWS 111122223333.

AWS CLI

```
aws s3api get-bucket-policy --bucket amzn-s3-demo-bucket1 --expected-bucket-
owner 111122223333
```

AWS SDK for Java 2.x

```
public void getBucketPolicyExample() {
    S3Client s3Client = S3Client.create();
    GetBucketPolicyRequest request = GetBucketPolicyRequest.builder()
            .bucket("amzn-s3-demo-bucket1")
            .expectedBucketOwner("111122223333")
            .build();
    try {
        GetBucketPolicyResponse response = s3Client.getBucketPolicy(request);
    }
    catch (S3Exception e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    }
}
```

Pembatasan dan batasan

Ketentuan pemilik bucket Amazon S3 memiliki pembatasan dan batasan berikut:

- Nilai parameter kondisi pemilik bucket harus berupa Akun AWS ID (nilai numerik 12 digit). Pengguna utama layanan tidak didukung.

- Kondisi pemilik bucket tidak tersedia untuk CreateBucket, ListBuckets, atau operasi apa pun yang termasuk dalam Kontrol AWS S3. Amazon S3 mengabaikan parameter ketentuan pemilik bucket yang disertakan dengan permintaan ke operasi ini.
- Ketentuan pemilik bucket hanya memverifikasi bahwa akun yang ditentukan dalam parameter verifikasi memiliki bucket. Ketentuan pemilik bucket tidak memeriksa konfigurasi bucket. Ini juga tidak menjamin bahwa konfigurasi bucket memenuhi ketentuan tertentu atau sesuai untuk status sebelumnya.

Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda

S3 Object Ownership adalah setelan tingkat ember Amazon S3 yang dapat Anda gunakan untuk mengontrol kepemilikan objek yang diunggah ke bucket Anda dan untuk menonaktifkan atau mengaktifkan daftar kontrol akses (). ACLs Secara default, Kepemilikan Objek disetel ke setelan diberlakukan pemilik Bucket dan semuanya ACLs dinonaktifkan. Saat ACLs dinonaktifkan, pemilik bucket memiliki semua objek di bucket dan mengelola akses ke data secara eksklusif menggunakan kebijakan manajemen akses.

Mayoritas kasus penggunaan modern di Amazon S3 tidak lagi memerlukan penggunaan ACLs, dan kami menyarankan Anda tetap ACLs dinonaktifkan kecuali dalam keadaan yang tidak biasa di mana Anda harus mengontrol akses untuk setiap objek secara individual. Dengan ACLs dinonaktifkan, Anda dapat menggunakan kebijakan untuk mengontrol akses ke setiap objek di bucket dengan lebih mudah, terlepas dari siapa yang mengunggah objek di bucket Anda.

Object Ownership memiliki tiga pengaturan yang dapat Anda gunakan untuk mengontrol kepemilikan objek yang diunggah ke bucket Anda dan untuk menonaktifkan atau mengaktifkan ACLs:

ACLs dinonaktifkan

- Pemilik bucket diberlakukan (default) — ACLs dinonaktifkan, dan pemilik bucket secara otomatis memiliki dan memiliki kontrol penuh atas setiap objek di bucket. ACLs tidak lagi memengaruhi izin ke data di bucket S3. Bucket tersebut menggunakan kebijakan untuk menentukan kontrol akses.

ACLs diaktifkan

- Pemilik bucket yang dipilih—Pemilik bucket memiliki dan diberikan kendali penuh atas objek baru yang ditulis akun lain ke bucket dengan ACL `bucket-owner-full-control` yang dibatasi.

- Penulis objek — Akun AWS Yang mengunggah objek memiliki objek, memiliki kontrol penuh atasnya, dan dapat memberikan pengguna lain akses ke sana melalui. ACLs

Untuk sebagian besar kasus penggunaan modern di S3, sebaiknya Anda tetap ACLs dinonaktifkan dengan menerapkan pengaturan yang diberlakukan oleh pemilik Bucket dan menggunakan kebijakan bucket Anda untuk berbagi data dengan pengguna di luar akun Anda sesuai kebutuhan. Pendekatan ini menyederhanakan manajemen izin. Anda dapat menonaktifkan ACLs pada bucket yang baru dibuat dan yang sudah ada. Untuk bucket yang baru dibuat, dinonaktifkan ACLs secara default. Dalam kasus bucket yang sudah ada yang sudah memiliki objek di dalamnya, setelah Anda menonaktifkan ACLs, objek dan bucket ACLs tidak lagi menjadi bagian dari evaluasi akses, dan akses diberikan atau ditolak berdasarkan kebijakan. Untuk bucket yang ada, Anda dapat mengaktifkan kembali ACLs kapan saja setelah menonaktifkannya, dan bucket serta objek ACLs yang sudah ada sebelumnya dipulihkan.

Sebelum menonaktifkan ACLs, kami sarankan Anda meninjau kebijakan bucket untuk memastikan kebijakan tersebut mencakup semua cara yang ingin Anda berikan akses ke bucket di luar akun Anda. Setelah Anda menonaktifkan ACLs, bucket Anda hanya menerima PUT permintaan yang tidak menentukan ACL atau PUT permintaan dengan kontrol penuh pemilik bucket ACLs, misalnya, ACL `bucket-owner-full-control` kalengan atau bentuk setara ACL ini yang dinyatakan dalam XHTML. Aplikasi yang ada yang mendukung kontrol penuh pemilik bucket ACLs tidak akan berdampak. PUTpermintaan yang berisi lainnya ACLs (misalnya, hibah khusus untuk tertentu Akun AWS) gagal dan mengembalikan `400` kesalahan dengan kode `AccessControlListNotSupported` kesalahan.

Sebaliknya, ember dengan pengaturan pilihan pemilik Bucket terus menerima dan menghormati ember dan objek ACLs. Dengan pengaturan ini, objek baru yang ditulis dengan ACL terekam `bucket-owner-full-control` secara otomatis dimiliki oleh pemilik bucket daripada penulis objek. Semua perilaku ACL lainnya tetap sama. Agar semua operasi PUT Amazon S3 menyertakan ACL `bucket-owner-full-control` yang terekam, Anda dapat menambahkan kebijakan bucket yang hanya mengizinkan unggahan objek menggunakan ACL ini.

Untuk melihat pengaturan Kepemilikan Objek yang diterapkan ke bucket, Anda dapat menggunakan metrik Lensa Penyimpanan Amazon S3. Lensa Penyimpanan S3 adalah fitur analisis penyimpanan cloud yang dapat Anda gunakan untuk mendapatkan visibilitas seluruh organisasi ke dalam penggunaan dan aktivitas penyimpanan objek. Untuk informasi lebih lanjut, lihat Menggunakan Lensa Penyimpanan S3 untuk menemukan pengaturan Kepemilikan Objek.

> **Note**
>
> Untuk informasi selengkapnya tentang menggunakan kelas penyimpanan Amazon S3 Express One Zone dengan bucket direktori, lihat S3 Express One Zone dan Bekerja dengan bucket direktori.

Pengaturan Kepemilikan Objek

Tabel ini menunjukkan dampak yang dimiliki setiap setelan Kepemilikan Objek pada ACLs, objek, kepemilikan objek, dan unggahan objek.

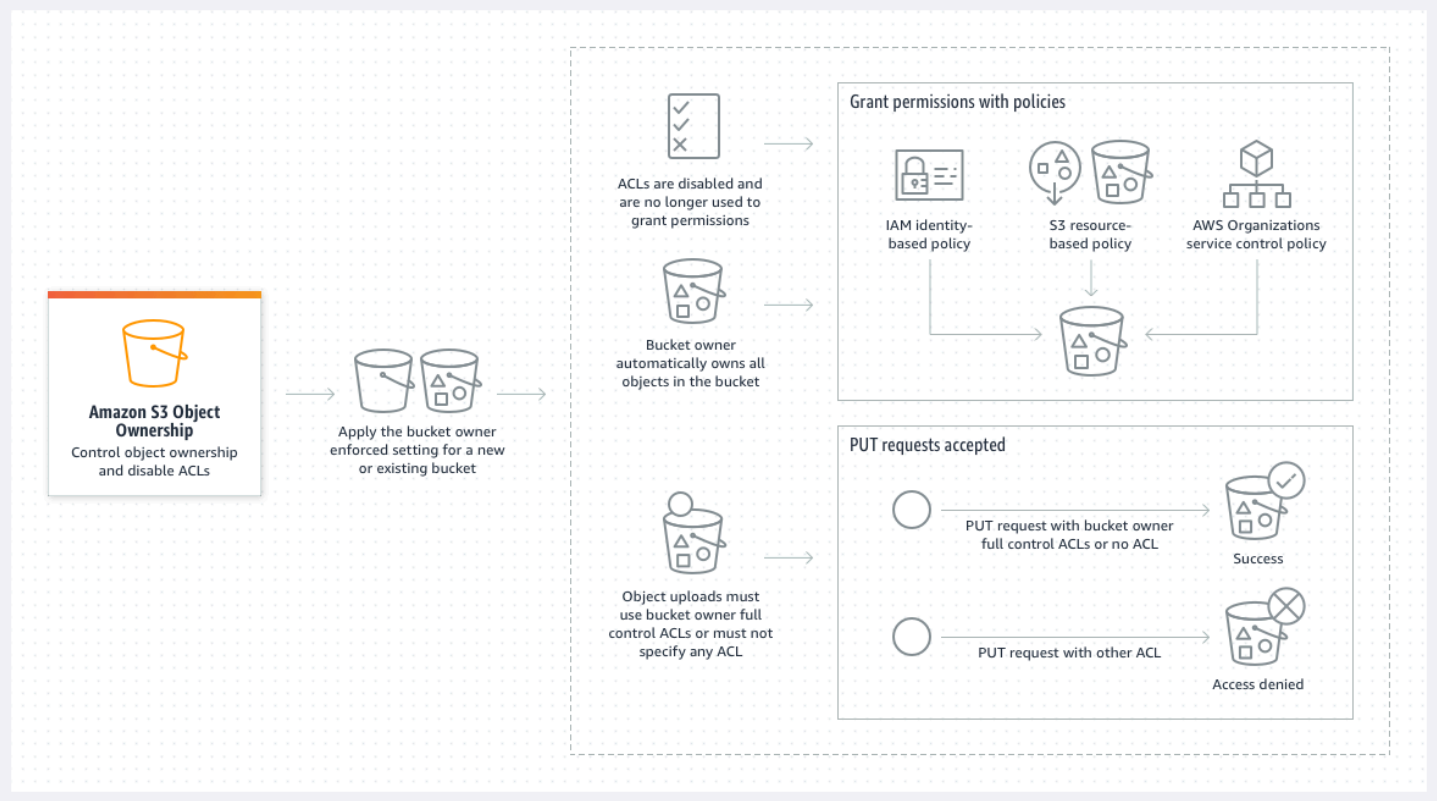
Pengaturan	Berlaku untuk	Efek pada kepemilikan objek	Efek pada ACLs	Unggahan diterima
Pemilik bucket diberlakukan (default)	Semua objek baru dan yang sudah ada	Pemilik bucket memiliki setiap objek.	ACLs dinonaktifkan dan tidak lagi memengaruhi izin akses ke bucket Anda. Permintaan untuk mengatur atau memperbarui ACLs gagal. Namun, permintaan untuk membaca ACLs didukung. Pemilik bucket memiliki kepemilikan dan kontrol penuh. Penulis objek tidak lagi memiliki	Unggahan dengan kontrol penuh pemilik bucket ACLs atau unggahan yang tidak menentukan ACL

Pengaturan	Berlaku untuk	Efek pada kepemilikan objek	Efek pada ACLs	Unggahan diterima
			kepemilikan dan kontrol penuh.	
bucket lebih disukai pemilik bucket	Objek baru	Jika unggahan objek menyertakan ACL `bucket-owner-full-control` terekam, pemilik bucket memiliki objek tersebut. Objek yang diunggah dengan ACLs yang lain dimiliki oleh akun penulisan.	ACLs dapat diperbarui dan dapat memberikan izin. Jika unggahan objek menyertakan ACL `bucket-owner-full-control` terekam, pemilik bucket memiliki akses kontrol penuh, dan penulis objek tidak lagi memiliki akses kontrol penuh.	Semua unggahan
Penulis objek	Objek baru	Penulis objek memiliki objek.	ACLs dapat diperbarui dan dapat memberikan izin. Penulis objek memiliki akses kontrol penuh.	Semua unggahan

Perubahan diperkenalkan dengan menonaktifkan ACLs

Ketika pengaturan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek ACLs diterapkan, akan dinonaktifkan dan Anda secara otomatis memiliki dan mengambil kendali penuh atas setiap objek di bucket tanpa mengambil tindakan tambahan apa pun. Pemilik bucket diberlakukan adalah pengaturan default untuk semua bucket yang baru dibuat. Setelah pengaturan diberlakukan pemilik Bucket diterapkan, Anda akan melihat tiga perubahan:

- Semua bucket ACLs dan objek ACLs dinonaktifkan, yang memberikan akses penuh kepada Anda, sebagai pemilik bucket. Saat Anda melakukan permintaan ACL baca pada bucket atau objek Anda, Anda akan melihat bahwa akses penuh hanya diberikan kepada pemilik bucket.
- Anda, sebagai pemilik bucket, secara otomatis memiliki dan memiliki kendali penuh atas setiap objek di bucket Anda.
- ACLs tidak lagi memengaruhi izin akses ke bucket Anda. Akibatnya, kontrol akses untuk data Anda didasarkan pada kebijakan, seperti kebijakan berbasis identitas AWS Identity and Access Management (IAM), kebijakan bucket Amazon S3, kebijakantitik akhir VPC, dan kebijakankontrol layanan Organizations () atau kebijakan SCPs kontrol sumber daya (). RCPs



Jika Anda menggunakan Pembuatan Penentuan Versi S3, pemilik bucket memiliki dan memiliki kontrol penuh atas semua versi objek di bucket Anda. Menerapkan pengaturan yang diberlakukan pemilik Bucket tidak akan menambahkan versi baru objek.

Objek baru dapat diunggah ke bucket Anda hanya jika mereka menggunakan kontrol penuh pemilik bucket ACLs atau tidak menentukan ACL. Unggahan objek gagal jika mereka menentukan ACL lainnya. Untuk informasi selengkapnya, lihat Pemecahan Masalah.

Karena contoh `PutObject` operasi berikut menggunakan AWS Command Line Interface (AWS CLI) menyertakan ACL `bucket-owner-full-control` kalengan, objek dapat diunggah ke bucket dengan dinonaktifkan. ACLs

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key key-name --body path-to-file --
acl bucket-owner-full-control
```

Karena `PutObject` operasi berikut tidak menentukan ACL, operasi ini juga berhasil untuk bucket dengan dinonaktifkan. ACLs

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key key-name --body path-to-file
```

> **Note**
>
> Jika orang lain Akun AWS memerlukan akses ke objek setelah mengunggah, Anda harus memberikan izin tambahan ke akun tersebut melalui kebijakan bucket. Untuk informasi selengkapnya, lihat Panduan yang menggunakan kebijakan untuk mengelola akses ke sumber daya Amazon S3.

Mengaktifkan kembali ACLs

Anda dapat mengaktifkan kembali ACLs dengan mengubah dari setelan yang diberlakukan pemilik Bucket ke setelan Kepemilikan Objek lain kapan saja. Jika Anda menggunakan objek ACLs untuk manajemen izin sebelum menerapkan setelan diberlakukan pemilik Bucket dan Anda tidak memigrasikan izin ACL objek ini ke kebijakan bucket, setelah Anda mengaktifkan kembali ACLs, izin ini akan dipulihkan. Selain itu, objek yang ditulis ke bucket saat pengaturan yang diberlakukan pemilik Bucket diterapkan masih dimiliki oleh pemilik bucket.

Misalnya, jika Anda mengubah dari pengaturan yang diberlakukan pemilik Bucket kembali ke pengaturan Object writer, Anda, sebagai pemilik bucket, tidak lagi memiliki dan memiliki

kontrol penuh atas objek yang sebelumnya dimiliki oleh orang lain Akun AWS. Sebaliknya, akun pengunggahan kembali memiliki objek-objek ini. Objek yang dimiliki oleh akun lain digunakan ACLs untuk izin, sehingga Anda tidak dapat menggunakan kebijakan untuk memberikan izin ke objek tersebut. Namun, Anda, sebagai pemilik bucket, masih memiliki objek apa pun yang ditulis ke bucket saat pengaturan yang diberlakukan oleh pemilik Bucket diterapkan. Objek-objek ini tidak dimiliki oleh penulis objek, bahkan jika Anda mengaktifkan kembali ACLs.

Untuk petunjuk tentang mengaktifkan dan mengelola ACLs menggunakan AWS Management Console, AWS Command Line Interface (CLI), REST API, AWS SDKs atau, lihat. Mengkonfigurasi ACLs

Prasyarat untuk menonaktifkan ACLs

Sebelum Anda menonaktifkan ACLs bucket yang sudah ada, selesaikan prasyarat berikut.

- Tinjau bucket dan objek ACLs dan migrasi izin ACL
- Identifikasi permintaan yang memerlukan ACL untuk otorisasi
- Meninjau dan memperbarui kebijakan bucket yang menggunakan kunci kondisi terkait ACL

Izin Kepemilikan Objek

Untuk menerapkan, memperbarui, atau menghapus pengaturan Kepemilikan Objek untuk bucket, Anda memerlukan izin `s3:PutBucketOwnershipControls` tersebut. Untuk mengembalikan pengaturan Kepemilikan Objek untuk bucket, Anda memerlukan izin `s3:GetBucketOwnershipControls`. Untuk informasi selengkapnya, lihat Mengatur Kepemilikan Objek saat membuat bucket dan Melihat pengaturan Kepemilikan Objek untuk bucket S3.

Menonaktifkan ACLs untuk semua bucket baru

Secara default, semua bucket baru dibuat dengan pengaturan diberlakukan pemilik Bucket diterapkan dan ACLs dinonaktifkan. Kami merekomendasikan agar tetap ACLs dinonaktifkan. Sebagai aturan umum, sebaiknya gunakan kebijakan berbasis sumber daya S3 (kebijakan bucket dan kebijakan jalur akses) atau kebijakan IAM untuk kontrol akses, bukan. ACLs Kebijakan adalah opsi kontrol akses yang disederhanakan dan lebih fleksibel. Dengan kebijakan bucket dan kebijakan titik akses, Anda dapat menentukan aturan yang berlaku secara luas di semua permintaan ke sumber daya Amazon S3 Anda.

Replikasi dan Kepemilikan Object

Bila Anda menggunakan replikasi S3 dan bucket sumber dan tujuan dimiliki oleh berbeda Akun AWS, Anda dapat menonaktifkan ACLs (dengan pengaturan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek) untuk mengubah kepemilikan replika menjadi bucket Akun AWS yang memiliki tujuan. Pengaturan ini meniru perilaku override pemilik yang ada tanpa memerlukan izin `s3:ObjectOwnerOverrideToBucketOwner`. Semua objek yang direplikasi ke bucket tujuan dengan pengaturan yang diberlakukan pemilik Bucket dimiliki oleh pemilik bucket tujuan. Untuk informasi selengkapnya tentang opsi penggantian pemilik untuk konfigurasi replikasi, lihat Mengubah pemilik replika.

Mengatur Kepemilikan Objek

Anda dapat menerapkan setelan Kepemilikan Objek dengan menggunakan konsol Amazon S3, AWS CLI AWS SDKs, Amazon S3 REST API, atau. AWS CloudFormation API REST dan AWS CLI perintah berikut mendukung Kepemilikan Objek:

API REST	AWS CLI	Deskripsi
PutBucketOwnershipControls	put-bucket-ownership-controls	Membuat atau memodifikasi pengaturan Kepemilikan Objek untuk bucket S3 yang ada.
CreateBucket	create-bucket	Membuat bucket menggunakan header permintaan `x-amz-object-ownership` untuk menentukan pengaturan Kepemilikan Objek.
GetBucketOwnershipControls	get-bucket-ownership-controls	Mengambil pengaturan Kepemilikan Objek untuk bucket Amazon S3.
DeleteBucketOwnershipControls	delete-bucket-ownership-controls	Menghapus pengaturan Kepemilikan Objek untuk bucket Amazon S3.

Untuk informasi tentang menerapkan dan bekerja dengan pengaturan Kepemilikan Objek, lihat topik-topik berikut.

Topik

- Prasyarat untuk menonaktifkan ACLs
- Mengatur Kepemilikan Objek saat membuat bucket
- Menyetel Kepemilikan Objek pada bucket yang ada
- Melihat pengaturan Kepemilikan Objek untuk bucket S3
- Menonaktifkan ACLs semua bucket baru dan menegakkan Kepemilikan Objek
- Pemecahan Masalah

Prasyarat untuk menonaktifkan ACLs

Daftar kontrol akses bucket (ACL) di Amazon S3 adalah mekanisme yang memungkinkan Anda menentukan izin granular untuk objek individual dalam bucket S3, yang menentukan akun atau grup AWS mana yang dapat mengakses dan memodifikasi objek tersebut. Mayoritas kasus penggunaan modern di Amazon S3 tidak lagi memerlukan penggunaan. ACLs Kami menyarankan Anda menggunakan AWS Identity and Access Management (IAM) dan kebijakan bucket untuk mengelola akses, dan untuk tetap ACLs dinonaktifkan, kecuali dalam keadaan yang tidak biasa di mana Anda perlu mengontrol akses untuk setiap objek secara individual.

Jika Anda telah ACLs mengaktifkan bucket, sebelum menonaktifkan ACLs, lengkapi prasyarat berikut:

Topik

- Tinjau bucket dan objek ACLs dan migrasi izin ACL
- Identifikasi permintaan yang memerlukan ACL untuk otorisasi
- Meninjau dan memperbarui kebijakan bucket yang menggunakan kunci kondisi terkait ACL
- Contoh kasus penggunaan

Tinjau bucket dan objek ACLs dan migrasi izin ACL

Saat Anda menonaktifkan ACLs, izin yang diberikan oleh bucket dan objek ACLs tidak lagi memengaruhi akses. Sebelum menonaktifkan ACLs, tinjau bucket dan objek Anda ACLs.

Setiap bucket dan objek Anda yang ada ACLs memiliki ekuivalen dalam kebijakan IAM. Contoh kebijakan bucket berikut menunjukkan cara `READ` dan `WRITE` izin untuk bucket dan ACLs peta objek ke izin IAM. Untuk informasi selengkapnya tentang bagaimana setiap ACL menerjemahkan ke izin IAM, lihat Pemetaan izin ACL dan izin kebijakan akses.

Sebelum Anda menonaktifkan ACLs:

- Jika bucket ACL memberikan akses di luar AWS akun, pertama-tama, Anda harus memigrasikan izin ACL bucket ke kebijakan bucket Anda.
- Selanjutnya, setel ulang bucket ACL Anda ke ACL pribadi default.
- Kami juga menyarankan Anda meninjau izin ACL tingkat objek dan memigrasikannya ke kebijakan bucket Anda.

Jika bucket Anda ACLs memberikan izin baca atau tulis kepada orang lain di luar akun Anda, sebelum Anda dapat menonaktifkan ACLs, Anda harus memigrasikan izin ini ke kebijakan bucket Anda. Setelah memigrasikan izin ini, Anda dapat menyetel Kepemilikan Objek ke setelan yang diberlakukan pemilik Bucket. Jika Anda tidak memigrasikan bucket ACLs yang memberikan akses baca atau tulis di luar akun, permintaan Anda untuk menerapkan setelan yang diberlakukan pemilik Bucket gagal dan mengembalikan kode InvalidBucketAclWithObjectOwnershipkesalahan.

Jika bucket ACL memberikan akses di luar Akun AWS, sebelum menonaktifkan ACLs, Anda harus memigrasikan izin ACL bucket ke kebijakan bucket dan mengatur ulang ACL bucket Anda ke ACL pribadi default. Jika Anda tidak melakukan migrasi dan mengatur ulang, permintaan Anda untuk menerapkan pengaturan yang diberlakukan pemilik Bucket untuk menonaktifkan ACLs gagal dan mengembalikan kode InvalidBucketAclWithObjectOwnershipkesalahan. Kami juga menyarankan Anda meninjau izin ACL objek dan memigrasikannya ke kebijakan bucket Anda.

Untuk meninjau dan memigrasi izin ACL ke kebijakan bucket, lihat topik berikut.

Topik

- Contoh kebijakan bucket
- Menggunakan konsol S3 untuk meninjau dan memigrasikan izin ACL
- Menggunakan AWS CLI untuk meninjau dan memigrasi izin ACL

Contoh kebijakan bucket

Contoh kebijakan bucket ini menunjukkan cara memigrasi `READ` dan `WRITE` meng-bucket serta objek izin ACL untuk pihak ketiga Akun AWS ke kebijakan bucket. `READ_ACP`dan kurang `WRITE_ACP` ACLs relevan untuk kebijakan karena mereka memberikan izin terkait ACL (`s3:GetBucketAcl`,, `s3:GetObjectAcls3:PutBucketAcl`, dan). `s3:PutObjectAcl`

Example – **READ** ACL untuk bucket

Jika bucket Anda memiliki `READ` ACL yang memberikan Akun AWS *111122223333* izin untuk mencantumkan konten bucket, Anda dapat menulis kebijakan bucket yang memberikan `s3:ListBucket` `s3:ListBucketMultipartUploads` izin untuk bucket Anda. `s3:ListBucketVersions`

```
{
  "Version": "2012-10-17",
  "Statement": [
   {
    "Sid": "Permission to list the objects in a bucket",
    "Effect": "Allow",
    "Principal": {
     "AWS": [

      "arn:aws:iam::111122223333:root"
     ]
    },
    "Action": [
     "s3:ListBucket",
     "s3:ListBucketVersions",
     "s3:ListBucketMultipartUploads"
    ],
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket"
   }
  ]
 }
```

Example — **READ** ACLs untuk setiap objek dalam ember

Jika setiap objek di bucket Anda memiliki `READ` ACL yang memberikan akses ke Akun AWS *111122223333*, Anda dapat menulis kebijakan bucket yang memberikan `s3:GetObject` dan `s3:GetObjectVersion` izin ke akun ini untuk setiap objek di bucket Anda.

```
{
  "Version": "2012-10-17",
  "Statement": [
   {
    "Sid": "Read permission for every object in a bucket",
    "Effect": "Allow",
    "Principal": {
     "AWS": [
      "arn:aws:iam::111122223333:root"
     ]
    },
    "Action": [
     "s3:GetObject",
     "s3:GetObjectVersion"
    ],
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
   }
  ]
 }
```

Contoh elemen sumber daya ini memberikan akses ke objek tertentu.

```
"Resource": "arn:aws:s3:::amzn-s3-demo-bucket/OBJECT-KEY"
```

Example – ACL **`WRITE`** yang memberikan izin untuk menulis objek ke bucket

Jika bucket Anda memiliki `WRITE` ACL yang memberikan Akun AWS *111122223333* izin untuk menulis objek ke bucket, Anda dapat menulis kebijakan bucket yang memberikan `s3:PutObject` izin untuk bucket Anda.

```
{
  "Version": "2012-10-17",
  "Statement": [
   {
    "Sid": "Permission to write objects to a bucket",
    "Effect": "Allow",
    "Principal": {
     "AWS": [
      "arn:aws:iam::111122223333:root"
     ]
    },
    "Action": [
```

```
      "s3:PutObject"
     ],
     "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
    }
   ]
}
```

Menggunakan konsol S3 untuk meninjau dan memigrasikan izin ACL

Untuk meninjau izin ACL bucket

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di dalam daftar Bucket, pilih nama bucket.
3. Pilih tab Izin.
4. Di bagian bawah Daftar kontrol akses (ACL), tinjau izin ACL bucket Anda.

Untuk meninjau izin ACL objek

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di daftar Bucket, pilih nama bucket yang berisi objek Anda.
3. Di daftar Objek, pilih nama objek Anda.
4. Pilih tab Izin.
5. Di bagian bawah Daftar kontrol akses (ACL), tinjau izin ACL objek Anda.

Untuk memigrasi izin ACL dan memperbarui bucket ACL

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di dalam daftar Bucket, pilih nama bucket.
3. Pada tab Izin, di bawah Kebijakan bucket, pilih Edit.
4. Di kotak Kebijakan, tambahkan atau perbarui kebijakan bucket Anda.

   Misalnya kebijakan bucket, lihat Contoh kebijakan bucket danContoh kasus penggunaan.
5. Pilih Simpan perubahan.

6. Perbarui bucket ACL Anda untuk menghapus pemberian ACL ke grup lain atau. Akun AWS
7. Terapkan pengaturan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek.

Menggunakan AWS CLI untuk meninjau dan memigrasi izin ACL

1. Untuk mengembalikan bucket ACL untuk bucket Anda, gunakan get-bucket-acl AWS CLI perintah:

```
aws s3api get-bucket-acl --bucket amzn-s3-demo-bucket
```

Misalnya, bucket ACL ini memberikan `WRITE` dan `READ` akses ke akun pihak ketiga. Dalam ACL ini, akun pihak ketiga diidentifikasi oleh ID pengguna kanonik. Untuk menerapkan pengaturan yang diberlakukan dan menonaktifkan pemilik Bucket ACLs, Anda harus memigrasikan izin ini untuk akun pihak ketiga ke kebijakan bucket.

```
{
  "Owner": {
   "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
   "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID"
  },
  "Grants": [
   {
    "Grantee": {
     "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
     "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID",
     "Type": "CanonicalUser"
    },
    "Permission": "FULL_CONTROL"
   },
   {
    "Grantee": {
     "DisplayName": "THIRD-PARTY-EXAMPLE-ACCOUNT",
     "ID":
 "72806de9d1ae8b171cca9e2494a8d1335dfced4ThirdPartyAccountCanonicalUserID",
     "Type": "CanonicalUser"
    },
    "Permission": "READ"
   },
   {
    "Grantee": {
     "DisplayName": "THIRD-PARTY-EXAMPLE-ACCOUNT",
```

```
    "ID":
"72806de9d1ae8b171cca9e2494a8d1335dfced4ThirdPartyAccountCanonicalUserID",
    "Type": "CanonicalUser"
   },
   "Permission": "WRITE"
  }
 ]
}
```

Untuk contoh lain ACLs, lihatContoh kasus penggunaan.

2. Migrasikan izin ACL bucket Anda ke kebijakan bucket:

   Contoh kebijakan bucket ini memberikan izin `s3:PutObject` dan `s3:ListBucket` untuk akun pihak ketiga. Dalam kebijakan bucket, akun pihak ketiga diidentifikasi oleh Akun AWS ID (*111122223333*).

```
aws s3api put-bucket-policy --bucket amzn-s3-demo-bucket --policy file://
policy.json

 policy.json:
 {
  "Version": "2012-10-17",
  "Statement": [
   {
    "Sid": "PolicyForCrossAccountAllowUpload",
    "Effect": "Allow",
    "Principal": {
     "AWS": [
      "arn:aws:iam::111122223333:root"
     ]
    },
    "Action": [
     "s3:PutObject",
     "s3:ListBucket"
    ],
    "Resource": [
     "arn:aws:s3:::amzn-s3-demo-bucket",
     "arn:aws:s3:::amzn-s3-demo-bucket/*"
   }
  ]
 }
```

Untuk contoh kebijakan bucket lainnya, lihat Contoh kebijakan bucket dan Contoh kasus penggunaan.

3. Untuk mengembalikan ACL untuk objek tertentu, gunakan get-object-acl AWS CLI perintah.

```
aws s3api get-object-acl --bucket amzn-s3-demo-bucket --key EXAMPLE-OBJECT-KEY
```

4. Jika diperlukan, migrasi izin ACL objek ke kebijakan bucket Anda.

   Contoh elemen sumber daya ini memberikan akses ke objek tertentu dalam kebijakan bucket.

```
"Resource": "arn:aws:s3:::amzn-s3-demo-bucket/EXAMPLE-OBJECT-KEY"
```

5. Setel ulang ACL untuk bucket Anda ke ACL default.

```
aws s3api put-bucket-acl --bucket amzn-s3-demo-bucket --acl private
```

6. Terapkan pengaturan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek.

Identifikasi permintaan yang memerlukan ACL untuk otorisasi

Untuk mengidentifikasi permintaan Amazon S3 yang diperlukan ACLs untuk otorisasi, Anda dapat menggunakan `aclRequired` nilai di log akses server Amazon S3 atau. AWS CloudTrail Jika permintaan memerlukan ACL untuk otorisasi atau jika Anda memiliki permintaan PUT yang menentukan ACL, string adalah `Yes`. Jika tidak ACLs diperlukan, atau jika Anda menyetel ACL `bucket-owner-full-control` kalengan, atau jika permintaan diizinkan oleh kebijakan bucket Anda, string `aclRequired` nilainya adalah "`-`" di log akses server Amazon S3 dan tidak ada di dalamnya. CloudTrail Untuk informasi selengkapnya tentang nilai `aclRequired` yang diharapkan, lihat Nilai `aclRequired` untuk permintaan Amazon S3 umum.

Jika Anda memiliki `PutBucketAcl` atau `PutObjectAcl` meminta dengan header yang memberikan izin berbasis ACL, kecuali ACL yang `bucket-owner-full-control` dikalengkan, Anda harus menghapus header tersebut sebelum dapat menonaktifkannya. ACLs Jika tidak, permintaan Anda akan gagal.

Untuk semua permintaan lain yang memerlukan ACL untuk otorisasi, migrasi izin ACL tersebut ke kebijakan bucket. Kemudian, lepaskan bucket apa pun ACLs sebelum Anda mengaktifkan pengaturan yang diberlakukan pemilik bucket.

> **Note**
>
> Jangan hapus objek ACLs. Jika tidak, aplikasi yang mengandalkan objek ACLs untuk izin akan kehilangan akses.

Jika Anda melihat bahwa tidak ada permintaan yang memerlukan ACL untuk otorisasi, Anda dapat melanjutkan untuk menonaktifkan. ACLs Untuk informasi selengkapnya tentang mengidentifikasi permintaan, lihat Menggunakan log akses server Amazon S3 untuk mengidentifikasi permintaan dan Mengidentifikasi permintaan Amazon S3 menggunakan CloudTrail.

Meninjau dan memperbarui kebijakan bucket yang menggunakan kunci kondisi terkait ACL

Setelah Anda menerapkan pengaturan diberlakukan pemilik Bucket untuk menonaktifkan ACLs, objek baru dapat diunggah ke bucket Anda hanya jika permintaan menggunakan kontrol penuh pemilik bucket ACLs atau tidak menentukan ACL. Sebelum menonaktifkan ACLs, tinjau kebijakan bucket Anda untuk kunci kondisi terkait ACL.

Jika kebijakan bucket Anda menggunakan kunci kondisi terkait ACL untuk mewajibkan ACL yang terekam `bucket-owner-full-control` (misalnya, `s3:x-amz-acl`), Anda tidak perlu memperbarui kebijakan bucket. Kebijakan bucket berikut menggunakan `s3:x-amz-acl` untuk mewajibkan ACL yang terekam `bucket-owner-full-control` untuk permintaan `PutObject` S3. Kebijakan ini masih mengharuskan penulis objek untuk menentukan ACL terekam `bucket-owner-full-control`. Namun, bucket dengan ACLs dinonaktifkan masih menerima ACL ini, sehingga permintaan terus berhasil tanpa perlu perubahan sisi klien.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Only allow writes to my bucket with bucket owner full control",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:user/ExampleUser"
                ]
            },
            "Action": [
```

```
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control"
                }
            }
        }
    ]
}
```

Namun, jika kebijakan bucket Anda menggunakan kunci kondisi terkait ACL yang memerlukan ACL berbeda, Anda harus menghapus kunci kondisi ini. Contoh kebijakan bucket ini memerlukan `public-read` ACL untuk `PutObject` permintaan S3 dan oleh karena itu harus diperbarui sebelum menonaktifkan. ACLs

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Only allow writes to my bucket with public read access",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:user/ExampleUser"                ]
            },
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "public-read"
                }
            }
        }
    ]
}
```

Contoh kasus penggunaan

Contoh berikut ini menunjukkan cara memigrasikan izin ACL ke kebijakan bucket untuk kasus penggunaan tertentu.

Topik

- Berikan akses ke grup pengiriman log S3 untuk pencatatan akses server
- Berikan akses baca publik ke objek dalam bucket
- Berikan Amazon ElastiCache (Redis OSS) akses ke bucket S3 Anda

Berikan akses ke grup pengiriman log S3 untuk pencatatan akses server

Jika Anda ingin menerapkan pengaturan yang diberlakukan pemilik Bucket ACLs untuk menonaktifkan bucket tujuan pencatatan akses server (juga dikenal sebagai bucket target), Anda harus memigrasikan izin ACL bucket untuk grup pengiriman log S3 ke prinsipal layanan logging (`logging.s3.amazonaws.com`) dalam kebijakan bucket. Untuk informasi lebih lanjut tentang izin pengiriman log, lihat Izin untuk pengiriman log.

ACL bucket ini memberikan akses `WRITE` dan `READ_ACP` ke grup pengiriman log S3:

```
{
    "Owner": {
        "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
        "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID"
    },
    "Grants": [
        {
            "Grantee": {
                "Type": "CanonicalUser",
                "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
                "ID":
 "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID"
            },
            "Permission": "FULL_CONTROL"
        },
        {
            "Grantee": {
                "Type": "Group",
                "URI": "http://acs.amazonaws.com/groups/s3/LogDelivery"
            },
            "Permission": "WRITE"
```

```
        },
        {
            "Grantee": {
                "Type": "Group",
                "URI": "http://acs.amazonaws.com/groups/s3/LogDelivery"
            },
            "Permission": "READ_ACP"
        }
    ]
}
```

Untuk memigrasikan izin ACL bucket untuk grup pengiriman log S3 ke pengguna utama layanan logging dalam kebijakan bucket

1. Tambahkan kebijakan bucket berikut ke bucket tujuan Anda, ganti nilai contoh.

```
aws s3api put-bucket-policy --bucket amzn-s3-demo-bucket --policy file://
policy.json

policy.json:      {
    {
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3ServerAccessLogsPolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "logging.s3.amazonaws.com"
            },
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/EXAMPLE-LOGGING-PREFIX*",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:::SOURCE-BUCKET-NAME"
                },
                "StringEquals": {
                    "aws:SourceAccount": "SOURCE-AWS-ACCOUNT-ID"
                }
            }
        }
    ]
```

```
}
```

2. Setel ulang ACL untuk bucket tujuan Anda ke ACL default.

```
aws s3api put-bucket-acl --bucket amzn-s3-demo-bucket --acl private
```

3. Terapkan pengaturan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek ke bucket tujuan Anda.

Berikan akses baca publik ke objek dalam bucket

Jika objek Anda ACLs memberikan akses baca publik ke semua objek di bucket, Anda dapat memigrasikan izin ACL ini ke kebijakan bucket.

Objek ACL ini memberikan akses baca publik ke objek dalam bucket:

```
{
    "Owner": {
        "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
        "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID"
    },
    "Grants": [
        {
            "Grantee": {
                "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
                "ID":
 "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID",
                "Type": "CanonicalUser"
            },
            "Permission": "FULL_CONTROL"
        },
        {
            "Grantee": {
                "Type": "Group",
                "URI": "http://acs.amazonaws.com/groups/global/AllUsers"
            },
            "Permission": "READ"
        }
    ]
}
```

Untuk memigrasikan izin ACL baca publik ke kebijakan bucket

1. Untuk memberikan akses baca publik ke semua objek di bucket Anda, tambahkan kebijakan bucket berikut, ganti nilai contoh.

```
aws s3api put-bucket-policy --bucket amzn-s3-demo-bucket --policy
 file://policy.json

policy.json:
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

   Untuk memberikan akses publik ke objek tertentu dalam kebijakan bucket, gunakan format berikut untuk elemen `Resource` tersebut.

```
"Resource": "arn:aws:s3:::amzn-s3-demo-bucket/OBJECT-KEY"
```

   Untuk memberikan akses publik ke semua objek dengan prefiks tertentu, gunakan format berikut untuk elemen `Resource` tersebut.

```
"Resource": "arn:aws:s3:::amzn-s3-demo-bucket/PREFIX/*"
```

2. Terapkan pengaturan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek.

Berikan Amazon ElastiCache (Redis OSS) akses ke bucket S3 Anda

Anda dapat mengekspor cadangan ElastiCache (Redis OSS) Anda ke bucket S3, yang memberi Anda akses ke cadangan dari luar. ElastiCache Untuk mengekspor cadangan ke bucket S3, Anda harus memberikan ElastiCache izin untuk menyalin snapshot ke bucket. Jika Anda telah memberikan izin ke ElastiCache ACL bucket, Anda harus memigrasikan izin ini ke kebijakan bucket sebelum menerapkan pengaturan yang diberlakukan pemilik Bucket untuk menonaktifkan. ACLs Untuk informasi selengkapnya, lihat Memberikan ElastiCache akses ke bucket Amazon S3 Anda di ElastiCache Panduan Pengguna Amazon.

Contoh berikut menunjukkan izin ACL bucket yang memberikan izin. ElastiCache

```
{
    "Owner": {
        "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
        "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID"
    },
    "Grants": [
        {
            "Grantee": {
                "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
                "ID":
 "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID",
                "Type": "CanonicalUser"
            },
            "Permission": "FULL_CONTROL"
        },
        {
            "Grantee": {
                "DisplayName": "aws-scs-s3-readonly",
                "ID":
"540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353",
                "Type": "CanonicalUser"
            },
            "Permission": "READ"
        },
        {
            "Grantee": {
                "DisplayName": "aws-scs-s3-readonly",
                "ID":
"540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353",
                "Type": "CanonicalUser"
            },
```

```
            "Permission": "WRITE"
        },
        {
            "Grantee": {
                "DisplayName": "aws-scs-s3-readonly",
                "ID":
 "540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353",
                "Type": "CanonicalUser"
            },
            "Permission": "READ_ACP"
        }
    ]
}
```

Untuk memigrasikan izin ACL bucket untuk ElastiCache (Redis OSS) ke kebijakan bucket

1. Tambahkan kebijakan bucket berikut ke bucket Anda, ganti nilai contoh.

```
aws s3api put-bucket-policy --bucket amzn-s3-demo-bucket --policy
 file://policy.json

policy.json:
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Stmt15399483",
            "Effect": "Allow",
            "Principal": {
                "Service": "Region.elasticache-snapshot.amazonaws.com"
            },
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:GetBucketAcl",
                "s3:ListMultipartUploadParts",
                "s3:ListBucketMultipartUploads"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
```

```
        }
    ]
}
```

2. Setel ulang ACL untuk bucket Anda ke ACL default:

```
aws s3api put-bucket-acl --bucket amzn-s3-demo-bucket --acl private
```

3. Terapkan pengaturan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek.

Mengatur Kepemilikan Objek saat membuat bucket

Saat membuat bucket, Anda dapat mengonfigurasi Kepemilikan S3 Object. Untuk mengatur Kepemilikan Objek untuk bucket yang ada, lihat Menyetel Kepemilikan Objek pada bucket yang ada.

S3 Object Ownership adalah setelan tingkat ember Amazon S3 yang dapat Anda gunakan untuk menonaktifkan daftar kontrol akses ACLs () dan mengambil kepemilikan setiap objek di bucket Anda, menyederhanakan manajemen akses untuk data yang disimpan di Amazon S3. Secara default, Kepemilikan Objek S3 disetel ke pengaturan yang diberlakukan pemilik Bucket, dan ACLs dinonaktifkan untuk bucket baru. Dengan ACLs dinonaktifkan, pemilik bucket memiliki setiap objek di bucket dan mengelola akses ke data secara eksklusif dengan menggunakan kebijakan manajemen akses. Kami menyarankan agar Anda tetap ACLs dinonaktifkan, kecuali dalam keadaan yang tidak biasa di mana Anda harus mengontrol akses untuk setiap objek secara individual.

Object Ownership memiliki tiga pengaturan yang dapat Anda gunakan untuk mengontrol kepemilikan objek yang diunggah ke bucket Anda dan untuk menonaktifkan atau mengaktifkan ACLs:

ACLs dinonaktifkan

- Pemilik bucket diberlakukan (default) — ACLs dinonaktifkan, dan pemilik bucket secara otomatis memiliki dan memiliki kontrol penuh atas setiap objek di bucket. ACLs tidak lagi memengaruhi izin ke data di bucket S3. Bucket tersebut menggunakan kebijakan untuk menentukan kontrol akses.

ACLs diaktifkan

- Pemilik bucket yang dipilih—Pemilik bucket memiliki dan diberikan kendali penuh atas objek baru yang ditulis akun lain ke bucket dengan ACL `bucket-owner-full-control` yang dibatasi.
- Penulis objek — Akun AWS Yang mengunggah objek memiliki objek, memiliki kontrol penuh atasnya, dan dapat memberikan pengguna lain akses ke sana melalui. ACLs

Izin: Untuk menerapkan pengaturan yang diberlakukan pemilik Bucket atau pengaturan pilihan pemilik Bucket, Anda harus memiliki izin berikut: `s3:CreateBucket` dan `s3:PutBucketOwnershipControls`. Tidak diperlukan izin tambahan saat membuat bucket dengan pengaturan Object writer yang diterapkan. Untuk informasi selengkapnya tentang izin Amazon S3, lihat Kunci tindakan, sumber daya, dan kondisi untuk Amazon S3 di Referensi Otorisasi Layanan.

Untuk informasi selengkapnya tentang izin operasi S3 API menurut jenis sumber daya S3, lihat. Izin yang diperlukan untuk operasi API Amazon S3

Important

Mayoritas kasus penggunaan modern di Amazon S3 tidak lagi memerlukan penggunaan ACLs, dan kami menyarankan Anda menonaktifkan ACLs kecuali dalam keadaan yang tidak biasa di mana Anda perlu mengontrol akses untuk setiap objek secara individual. Dengan Object Ownership, Anda dapat menonaktifkan ACLs dan mengandalkan kebijakan untuk kontrol akses. Saat menonaktifkan ACLs, Anda dapat dengan mudah memelihara ember dengan objek yang diunggah oleh AWS akun yang berbeda. Anda, sebagai pemilik bucket, memiliki semua objek di bucket dan dapat mengelola akses ke mereka menggunakan kebijakan.

Menggunakan konsol S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi di bagian atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah tempat Anda ingin membuat ember.

Note

- Setelah membuat bucket, Anda tidak dapat mengubah Region.
- Untuk meminimalkan latensi dan biaya serta memenuhi persyaratan regulasi, pilih Wilayah yang dekat dengan Anda. Objek yang disimpan di Wilayah tidak pernah keluar dari Wilayah kecuali Anda secara tegas mentransfer atau mereplikasinya ke Wilayah lain. Untuk daftar Amazon S3 Wilayah AWS, lihat Layanan AWS titik akhir di. Referensi Umum Amazon Web Services

3. Di panel navigasi kiri, pilih Bucket tujuan umum.
4. Pilih Buat bucket. Halaman Buat bucket terbuka.
5. Untuk Nama bucket, masukkan nama untuk bucket Anda.

   Nama bucket harus:

   - Unik dalam partisi. Partisi adalah pengelompokan Wilayah. AWS saat ini memiliki tiga partisi: `aws` (Kawasan komersial), `aws-cn` (Wilayah Tiongkok), dan `aws-us-gov` (AWS GovCloud (US) Regions).
   - Panjangnya antara 3 hingga 63 karakter.
   - Hanya terdiri dari huruf kecil, angka, titik (`.`), dan tanda hubung (). `-` Untuk kompatibilitas terbaik, sebaiknya hindari penggunaan period (`.`) dalam nama bucket, kecuali bucket yang hanya digunakan untuk hosting situs statis.
   - Dimulai dan diakhiri dengan huruf atau angka.
   - Untuk daftar lengkap aturan penamaan ember, lihat. Aturan penamaan bucket tujuan umum

   > **Important**
   >
   > - Setelah membuat bucket, Anda tidak dapat mengubah namanya.
   > - Jangan sertakan informasi sensitif dalam nama bucket. Nama bucket terlihat dalam URLs yang menunjuk objek dalam bucket.

6. (Opsional) Di bawah Konfigurasi umum, Anda dapat memilih untuk menyalin setelan bucket yang ada ke bucket baru Anda. Jika Anda tidak ingin menyalin pengaturan bucket yang ada, lewati ke langkah berikutnya.

   > **Note**
   >
   > Opsi ini:
   >
   > - Tidak tersedia di AWS CLI dan hanya tersedia di konsol Amazon S3
   > - Tidak menyalin kebijakan bucket dari bucket yang ada ke bucket baru

   Untuk menyalin setelan bucket yang ada, di bagian Salin setelan dari bucket yang ada, pilih Pilih bucket. Jendela Pilih ember terbuka. Temukan bucket dengan pengaturan yang ingin Anda

salin, lalu pilih Pilih bucket. Jendela Choose bucket ditutup, dan jendela Create bucket terbuka kembali.

Di bawah Salin pengaturan dari bucket yang ada, Anda sekarang melihat nama bucket yang Anda pilih. Pengaturan bucket baru Anda sekarang cocok dengan pengaturan bucket yang Anda pilih. Jika Anda ingin menghapus pengaturan yang disalin, pilih Kembalikan default. Tinjau setelan bucket yang tersisa di halaman Buat bucket. Jika Anda tidak ingin membuat perubahan apa pun, Anda dapat melompat ke langkah terakhir.

7. Di bawah Kepemilikan Objek, untuk menonaktifkan atau mengaktifkan ACLs dan mengontrol kepemilikan objek yang diunggah di bucket Anda, pilih salah satu setelan berikut:

   ACLs dinonaktifkan

   - Pemilik bucket diberlakukan (default) — ACLs dinonaktifkan, dan pemilik bucket secara otomatis memiliki dan memiliki kontrol penuh atas setiap objek di bucket tujuan umum. ACLs tidak lagi memengaruhi izin akses ke data di bucket tujuan umum S3. Bucket menggunakan kebijakan secara eksklusif untuk menentukan kontrol akses.

     Secara default, ACLs dinonaktifkan. Mayoritas kasus penggunaan modern di Amazon S3 tidak lagi memerlukan penggunaan. ACLs Kami menyarankan agar Anda tetap ACLs dinonaktifkan, kecuali dalam keadaan yang tidak biasa di mana Anda harus mengontrol akses untuk setiap objek secara individual. Untuk informasi selengkapnya, lihat Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.

   ACLs diaktifkan

   - Pemilik bucket yang dipilih—Pemilik bucket memiliki dan diberikan kendali penuh atas objek baru yang ditulis akun lain ke bucket dengan ACL `bucket-owner-full-control` yang dibatasi.

     Jika Anda menerapkan pengaturan Pemilik bucket yang dipilih, agar semua unggahan Amazon S3 menyertakan ACL `bucket-owner-full-control` yang terekam, Anda dapat menambahkan kebijakan bucket yang hanya mengizinkan unggahan objek yang menggunakan ACL ini.
   - Penulis objek — Akun AWS Yang mengunggah objek memiliki objek, memiliki kontrol penuh atasnya, dan dapat memberikan pengguna lain akses ke sana melalui. ACLs

> **Note**
>
> Pengaturan default-nya adalah Pemilik Bucket yang diberlakukan. Untuk menerapkan pengaturan default dan tetap ACLs dinonaktifkan, hanya `s3:CreateBucket` izin yang diperlukan. Untuk mengaktifkan ACLs, Anda harus memiliki `s3:PutBucketOwnershipControls` izin.

8. Di bawah Pengaturan Blokir Akses Publik untuk bucket ini, pilih pengaturan Blokir Akses Publik yang ingin Anda terapkan ke bucket.

   Secara default, semua pengaturan Blokir Akses Publik untuk bucket direktori diaktifkan. Kami menyarankan Anda tetap mengaktifkan semua pengaturan, kecuali Anda tahu bahwa Anda perlu menonaktifkan satu atau beberapa pengaturan untuk kasus penggunaan spesifik Anda. Untuk informasi lebih lanjut tentang pemblokiran akses publik, lihat Melakukan blok akses publik ke penyimpanan Amazon S3 Anda.

   > **Note**
   >
   > Untuk mengaktifkan semua pengaturan Blokir Akses Publik, hanya izin `s3:CreateBucket` yang diperlukan. Untuk mematikan pengaturan Blokir Akses Publik, Anda harus memiliki izin `s3:PutBucketPublicAccessBlock`.

9. (Opsional) Secara default, Bucket Versioning dinonaktifkan. Penentuan Versi adalah cara menyimpan beberapa varian objek dalam bucket yang sama. Anda dapat menggunakan Penentuan Versi untuk menyimpan, mengambil, dan memulihkan setiap versi dari setiap objek yang disimpan dalam bucket Anda. Dengan versioning, Anda dapat lebih mudah memulihkan dari tindakan pengguna yang tidak diinginkan dan kegagalan aplikasi. Untuk informasi selengkapnya tentang penentuan versi, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.

   Untuk mengaktifkan pembuatan versi di bucket Anda, pilih Aktifkan.

10. (Opsional) Di bawah Tanda, Anda dapat memilih untuk menambahkan tanda ke bucket Anda. Dengan alokasi AWS biaya, Anda dapat menggunakan tag bucket untuk membubuhi keterangan penagihan penggunaan bucket. Tanda adalah pasangan nilai kunci yang mewakili label yang Anda tetapkan ke bucket. Untuk informasi selengkapnya, lihat the section called “Menggunakan tanda alokasi biaya”.

Untuk menambahkan tanda bucket, masukkan Kunci dan secara opsional Nilai, lalu pilih Tambahkan Tanda.

11. Untuk mengonfigurasi enkripsi Default, di bawah Jenis enkripsi, pilih salah satu dari berikut ini:

    - Enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3)
    - Enkripsi sisi server dengan AWS Key Management Service kunci (SSE-KMS)
    - Enkripsi sisi server dua lapis dengan kunci () (DSSE-KMS AWS Key Management Service)AWS KMS

    > **Important**
    >
    > Jika Anda menggunakan opsi SSE-KMS atau DSSE-KMS untuk konfigurasi enkripsi default Anda, Anda tunduk pada kuota permintaan per detik (RPS) sebesar. AWS KMSUntuk informasi selengkapnya tentang AWS KMS kuota dan cara meminta kenaikan kuota, lihat Kuota di Panduan Pengembang AWS Key Management Service .

    Bucket dan objek baru dienkripsi dengan menggunakan enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3) sebagai tingkat dasar konfigurasi enkripsi. Untuk informasi selengkapnya tentang enkripsi default, lihat Mengatur perilaku enkripsi sisi server default untuk bucket Amazon S3. Untuk informasi selengkapnya tentang SSE-S3, lihat. Menggunakan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3)

    Untuk informasi selengkapnya tentang penggunaan enkripsi sisi server untuk mengenkripsi data Anda, lihat. the section called “Enkripsi data”

12. Jika Anda memilih enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3) atau enkripsi sisi server dual-layer dengan kunci () (DSSE-KMS), lakukan hal berikut: AWS Key Management Service AWS KMS

    a. Di bawah AWS KMS kunci, tentukan kunci KMS Anda dengan salah satu cara berikut ini:

        - Untuk memilih dari daftar kunci KMS yang tersedia, pilih Pilih dari Anda AWS KMS keys, dan pilih kunci KMS Anda dari daftar kunci yang tersedia.

          Kunci Kunci yang dikelola AWS (`aws/s3`) dan kunci terkelola pelanggan Anda muncul dalam daftar ini. Untuk informasi selengkapnya tentang CMK, lihat Kunci pelanggan dan AWS kunci di AWS Key Management Service Panduan Pengembang.

- Untuk memasukkan ARN kunci KMS, pilih Masukkan AWS KMS key ARN, dan masukkan ARN kunci KMS Anda di bidang yang muncul.
- Untuk membuat kunci terkelola pelanggan baru di AWS KMS konsol, pilih Buat kunci KMS.

  Untuk informasi selengkapnya tentang membuat AWS KMS key, lihat Membuat kunci di Panduan AWS Key Management Service Pengembang.

> **Important**
>
> Anda hanya dapat menggunakan tombol KMS yang tersedia Wilayah AWS sama dengan bucket. Konsol Amazon S3 hanya mencantumkan kunci 100 KMS pertama di Wilayah yang sama dengan bucket. Untuk menggunakan kunci KMS yang tidak terdaftar, Anda harus memasukkan ARN kunci KMS Anda. Jika Anda ingin menggunakan kunci KMS yang dimiliki oleh akun yang berbeda, Anda harus terlebih dahulu memiliki izin untuk menggunakan kunci tersebut, dan kemudian Anda harus memasukkan ARN kunci KMS. Untuk informasi selengkapnya tentang izin lintas akun untuk kunci KMS, lihat Membuat kunci KMS yang dapat digunakan akun lain di Panduan Pengembang AWS Key Management Service . Untuk informasi lebih lanjut tentang SSE-KMS, lihat Menentukan enkripsi di sisi server dengan AWS KMS (SSE-KMS). Untuk informasi lebih lanjut tentang DSSE-KMS, lihat the section called “Enkripsi di sisi server dua lapis (DSSE-KMS)”.
> Saat Anda menggunakan enkripsi sisi server AWS KMS key untuk Amazon S3, Anda harus memilih kunci KMS enkripsi simetris. Amazon S3 hanya mendukung kunci KMS enkripsi simetris dan tidak mendukung kunci KMS asimetris. Untuk informasi selengkapnya, lihat Mengidentifikasi tombol KMS simetris dan asimetris dalam Panduan Pengembang AWS Key Management Service .

b. Saat mengonfigurasi bucket untuk menggunakan enkripsi default dengan SSE-KMS, Anda juga dapat menggunakan S3 Bucket Keys. S3 Bucket Keys menurunkan biaya enkripsi dengan mengurangi lalu lintas permintaan dari Amazon S3 ke. AWS KMS Untuk informasi selengkapnya, lihat Mengurangi biaya SSE-KMS dengan Kunci Bucket Amazon S3. Kunci Bucket S3 tidak didukung untuk DSSE-KMS.

   Secara default, Kunci Bucket S3 diaktifkan di konsol Amazon S3. Sebaiknya biarkan Kunci Bucket S3 diaktifkan untuk menurunkan biaya Anda. Untuk menonaktifkan Kunci Bucket S3 untuk bucket Anda, di bawah Kunci Bucket, pilih Nonaktifkan.

13. (Opsional) Kunci Objek S3 membantu melindungi objek baru agar tidak dihapus atau ditimpa. Untuk informasi selengkapnya, lihat Mengunci objek dengan Object Lock. Jika Anda ingin mengaktifkan S3 Object Lock, lakukan hal berikut:

    a. Pilih Pengaturan lanjutan.

    > **Important**
    >
    > Mengaktifkan Object Lock secara otomatis memungkinkan pembuatan versi untuk bucket. Setelah mengaktifkan dan berhasil membuat bucket, Anda juga harus mengonfigurasi pengaturan penyimpanan default Object Lock dan penahanan legal pada tab Properties bucket.

    b. Jika Anda ingin mengaktifkan Kunci Objek, pilih Aktifkan, baca peringatan yang muncul, lalu setujui.

    > **Note**
    >
    > Untuk membuat bucket yang diaktifkan Object Lock, Anda harus memiliki izin berikut:`s3:CreateBucket`,`s3:PutBucketVersioning`, dan`s3:PutBucketObjectLockConfiguration`.

14. Pilih Buat bucket.

Menggunakan AWS CLI

Untuk menyetel Object Ownership saat membuat bucket baru, gunakan `create-bucket` AWS CLI perintah dengan `--object-ownership` parameter.

Contoh ini menerapkan pengaturan yang diberlakukan pemilik Bucket untuk bucket baru menggunakan: AWS CLI

```
aws s3api create-bucket --bucket  amzn-s3-demo-bucket --region us-east-1 --object-
ownership BucketOwnerEnforced
```

> **Important**
>
> Jika Anda tidak menyetel Kepemilikan Objek saat membuat bucket dengan menggunakan AWS CLI, pengaturan defaultnya adalah `ObjectWriter` (ACLs diaktifkan).

Menggunakan AWS SDK for Java

Contoh ini menyetel pengaturan yang diberlakukan pemilik Bucket untuk bucket baru menggunakan: AWS SDK untuk Java

```
    // Build the ObjectOwnership for CreateBucket
    CreateBucketRequest createBucketRequest = CreateBucketRequest.builder()
            .bucket(bucketName)
            .objectOwnership(ObjectOwnership.BucketOwnerEnforced)
            .build()

     // Send the request to Amazon S3
     s3client.createBucket(createBucketRequest);
```

Menggunakan AWS CloudFormation

Untuk menggunakan `AWS::S3::Bucket` AWS CloudFormation sumber daya untuk menyetel Kepemilikan Objek saat membuat bucket baru, lihat AWS::S3::Bucket di OwnershipControlsdalam Panduan AWS CloudFormation Pengguna.

Penggunaan API REST

Untuk menerapkan pengaturan diberlakukan pemilik Bucket untuk Kepemilikan S3 Object, gunakan operasi `CreateBucket` API dengan header `x-amz-object-ownership` permintaan disetel ke `BucketOwnerEnforced`. Untuk informasi dan contoh, lihat CreateBucketReferensi API Amazon Simple Storage Service.

Langkah selanjutnya: Setelah menerapkan pengaturan pilihan pemilik Bucket yang diberlakukan atau pemilik bucket untuk Kepemilikan Objek, Anda dapat mengambil langkah-langkah berikut:

- Pemilik bucket diberlakukan — Mengharuskan semua bucket baru dibuat dengan ACLs dinonaktifkan menggunakan kebijakan IAM atau Organizations.
- Pemilik bucket yang dipilih–Menambahkan kebijakan bucket S3 untuk ACL terekam `bucket-owner-full-control` untuk semua unggahan objek ke bucket Anda.

Menyetel Kepemilikan Objek pada bucket yang ada

Anda dapat mengonfigurasi Kepemilikan S3 Object pada bucket S3 yang ada. Untuk menerapkan Kepemilikan Objek saat membuat bucket, lihat Mengatur Kepemilikan Objek saat membuat bucket.

S3 Object Ownership adalah setelan tingkat ember Amazon S3 yang dapat Anda gunakan untuk menonaktifkan daftar kontrol akses ACLs () dan mengambil kepemilikan setiap objek di bucket Anda, menyederhanakan manajemen akses untuk data yang disimpan di Amazon S3. Secara default, Kepemilikan Objek S3 disetel ke pengaturan yang diberlakukan pemilik Bucket, dan ACLs dinonaktifkan untuk bucket baru. Dengan ACLs dinonaktifkan, pemilik bucket memiliki setiap objek di bucket dan mengelola akses ke data secara eksklusif dengan menggunakan kebijakan manajemen akses. Kami menyarankan agar Anda tetap ACLs dinonaktifkan, kecuali dalam keadaan yang tidak biasa di mana Anda harus mengontrol akses untuk setiap objek secara individual.

Object Ownership memiliki tiga pengaturan yang dapat Anda gunakan untuk mengontrol kepemilikan objek yang diunggah ke bucket Anda dan untuk menonaktifkan atau mengaktifkan ACLs:

ACLs dinonaktifkan

- Pemilik bucket diberlakukan (default) — ACLs dinonaktifkan, dan pemilik bucket secara otomatis memiliki dan memiliki kontrol penuh atas setiap objek di bucket. ACLs tidak lagi memengaruhi izin ke data di bucket S3. Bucket tersebut menggunakan kebijakan untuk menentukan kontrol akses.

ACLs diaktifkan

- Pemilik bucket yang dipilih—Pemilik bucket memiliki dan diberikan kendali penuh atas objek baru yang ditulis akun lain ke bucket dengan ACL `bucket-owner-full-control` yang dibatasi.
- Penulis objek — Akun AWS Yang mengunggah objek memiliki objek, memiliki kontrol penuh atasnya, dan dapat memberikan pengguna lain akses ke sana melalui. ACLs

Prasyarat: Sebelum menerapkan pengaturan yang diberlakukan pemilik Bucket untuk menonaktifkan ACLs, Anda harus memigrasikan izin ACL bucket ke kebijakan bucket dan mengatur ulang bucket Anda ke ACL pribadi default. ACLs Kami juga menyarankan Anda memigrasikan izin ACL objek ke kebijakan bucket dan mengedit kebijakan bucket yang memerlukan kontrol penuh ACLs selain pemilik bucket. ACLs Untuk informasi selengkapnya, lihat Prasyarat untuk menonaktifkan ACLs.

Izin: Untuk menggunakan operasi ini, Anda harus memiliki `s3:PutBucketOwnershipControls` izin. Untuk informasi selengkapnya tentang izin Amazon S3, lihat Kunci tindakan, sumber daya, dan kondisi untuk Amazon S3 di Referensi Otorisasi Layanan.

Untuk informasi selengkapnya tentang izin operasi S3 API menurut jenis sumber daya S3, lihat. Izin yang diperlukan untuk operasi API Amazon S3

Menggunakan konsol S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di daftar Bucket, pilih nama bucket yang ingin Anda terapkan pengaturan Kepemilikan S3 Object.
3. Pilih tab Izin.
4. Di bawah Kepemilikan Objek, pilih Edit.
5. Di bawah Kepemilikan Objek, untuk menonaktifkan atau mengaktifkan ACLs dan mengontrol kepemilikan objek yang diunggah di bucket Anda, pilih salah satu setelan berikut:

   ACLs dinonaktifkan

   - Pemilik bucket diberlakukan — ACLs dinonaktifkan, dan pemilik bucket secara otomatis memiliki dan memiliki kontrol penuh atas setiap objek di ember. ACLs tidak lagi memengaruhi izin ke data di bucket S3. Bucket tersebut menggunakan kebijakan untuk menentukan kontrol akses.

     Untuk mewajibkan semua bucket baru dibuat dengan ACLs dinonaktifkan menggunakan IAM atau AWS Organizations kebijakan, lihat. Menonaktifkan ACLs semua bucket baru (pemilik bucket diberlakukan)

   ACLs diaktifkan

   - Pemilik bucket yang dipilih—Pemilik bucket memiliki dan diberikan kendali penuh atas objek baru yang ditulis akun lain ke bucket dengan ACL `bucket-owner-full-control` yang dibatasi.

     Jika Anda menerapkan pengaturan pemilik bucket yang dipilih, agar semua unggahan Amazon S3 menyertakan ACL `bucket-owner-full-control` yang terekam, Anda

dapat menambahkan kebijakan bucket yang hanya mengizinkan unggahan objek yang menggunakan ACL ini.

- Penulis objek — Akun AWS Yang mengunggah objek memiliki objek, memiliki kontrol penuh atasnya, dan dapat memberikan pengguna lain akses ke sana melalui. ACLs

6. Pilih Simpan.

Menggunakan AWS CLI

Untuk menerapkan pengaturan Kepemilikan Objek untuk bucket yang ada, gunakan perintah `put-bucket-ownership-controls` dengan parameter `--ownership-controls`. Nilai yang valid untuk kepemilikan adalah`BucketOwnerEnforced`, `BucketOwnerPreferred`, atau `ObjectWriter`.

Contoh ini menerapkan pengaturan yang diberlakukan pemilik Bucket untuk bucket yang ada dengan menggunakan: AWS CLI

```
aws s3api put-bucket-ownership-controls --bucket amzn-s3-demo-bucket --ownership-
controls="Rules=[{ObjectOwnership=BucketOwnerEnforced}]"
```

Untuk informasi selengkapnya tentang `put-bucket-ownership-controls`, lihat put-bucket-ownership-controls di Panduan Pengguna AWS Command Line Interface .

Menggunakan AWS SDK for Java

Contoh ini menerapkan pengaturan `BucketOwnerEnforced` untuk Kepemilikan Objek pada bucket yang ada dengan menggunakan AWS SDK untuk Java:

```
        // Build the ObjectOwnership for BucketOwnerEnforced
        OwnershipControlsRule rule = OwnershipControlsRule.builder()
               .objectOwnership(ObjectOwnership.BucketOwnerEnforced)
               .build();

        OwnershipControls ownershipControls = OwnershipControls.builder()
                  .rules(rule)
                  .build()

         // Build the PutBucketOwnershipControlsRequest
         PutBucketOwnershipControlsRequest putBucketOwnershipControlsRequest =
               PutBucketOwnershipControlsRequest.builder()
```

```
                    .bucket(BUCKET_NAME)
                    .ownershipControls(ownershipControls)
                    .build();

        // Send the request to Amazon S3
        s3client.putBucketOwnershipControls(putBucketOwnershipControlsRequest);
```

Menggunakan AWS CloudFormation

AWS CloudFormation Untuk menggunakan pengaturan Kepemilikan Objek untuk bucket yang ada, lihat AWS::S3::Bucket OwnershipControlsdi Panduan AWS CloudFormation Pengguna.

Penggunaan API REST

Untuk menggunakan API REST untuk menerapkan pengaturan Kepemilikan Objek ke bucket S3 yang ada, gunakan `PutBucketOwnershipControls`. Untuk informasi selengkapnya, lihat PutBucketOwnershipControls dalam Referensi API Amazon Simple Storage Service.

Langkah selanjutnya: Setelah menerapkan pengaturan pilihan pemilik Bucket yang diberlakukan atau pemilik bucket untuk Kepemilikan Objek, Anda dapat mengambil langkah-langkah berikut:

- Pemilik bucket diberlakukan — Mengharuskan semua bucket baru dibuat dengan ACLs dinonaktifkan menggunakan kebijakan IAM atau Organizations.
- Pemilik bucket yang dipilih–Menambahkan kebijakan bucket S3 untuk ACL terekam `bucket-owner-full-control` untuk semua unggahan objek ke bucket Anda.

Melihat pengaturan Kepemilikan Objek untuk bucket S3

S3 Object Ownership adalah setelan tingkat ember Amazon S3 yang dapat Anda gunakan untuk menonaktifkan daftar kontrol akses ACLs () dan mengambil kepemilikan setiap objek di bucket Anda, menyederhanakan manajemen akses untuk data yang disimpan di Amazon S3. Secara default, Kepemilikan Objek S3 disetel ke pengaturan yang diberlakukan pemilik Bucket, dan ACLs dinonaktifkan untuk bucket baru. Dengan ACLs dinonaktifkan, pemilik bucket memiliki setiap objek di bucket dan mengelola akses ke data secara eksklusif dengan menggunakan kebijakan manajemen akses. Kami menyarankan agar Anda tetap ACLs dinonaktifkan, kecuali dalam keadaan yang tidak biasa di mana Anda harus mengontrol akses untuk setiap objek secara individual.

Object Ownership memiliki tiga pengaturan yang dapat Anda gunakan untuk mengontrol kepemilikan objek yang diunggah ke bucket Anda dan untuk menonaktifkan atau mengaktifkan ACLs:

ACLs dinonaktifkan

- Pemilik bucket diberlakukan (default) — ACLs dinonaktifkan, dan pemilik bucket secara otomatis memiliki dan memiliki kontrol penuh atas setiap objek di bucket. ACLs tidak lagi memengaruhi izin ke data di bucket S3. Bucket tersebut menggunakan kebijakan untuk menentukan kontrol akses.

ACLs diaktifkan

- Pemilik bucket yang dipilih—Pemilik bucket memiliki dan diberikan kendali penuh atas objek baru yang ditulis akun lain ke bucket dengan ACL `bucket-owner-full-control` yang dibatasi.
- Penulis objek — Akun AWS Yang mengunggah objek memiliki objek, memiliki kontrol penuh atasnya, dan dapat memberikan pengguna lain akses ke sana melalui. ACLs

Anda dapat melihat pengaturan Kepemilikan S3 Object untuk bucket Amazon S3. Untuk menyetel Kepemilikan Objek untuk bucket baru, lihatMengatur Kepemilikan Objek saat membuat bucket. Untuk mengatur Kepemilikan Objek untuk bucket yang ada, lihat Menyetel Kepemilikan Objek pada bucket yang ada.

Izin: Untuk menggunakan operasi ini, Anda harus memiliki izin `s3:GetBucketOwnershipControls`. Untuk informasi selengkapnya tentang izin Amazon S3, lihat Kunci tindakan, sumber daya, dan kondisi untuk Amazon S3 di Referensi Otorisasi Layanan.

Untuk informasi selengkapnya tentang izin operasi S3 API menurut jenis sumber daya S3, lihat. Izin yang diperlukan untuk operasi API Amazon S3

Menggunakan konsol S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di daftar Bucket, pilih nama bucket yang ingin Anda terapkan pengaturan Kepemilikan Objek.
3. Pilih tab Izin.
4. Di bawah Kepemilikan Objek, Anda dapat melihat pengaturan Kepemilikan Objek untuk bucket Anda.

Menggunakan AWS CLI

Untuk mengambil pengaturan Kepemilikan Objek S3 untuk bucket S3, gunakan perintah. get-bucket-ownership-controls AWS CLI

```
aws s3api get-bucket-ownership-controls --bucket amzn-s3-demo-bucket
```

Penggunaan API REST

Untuk mengambil pengaturan Kepemilikan Objek untuk bucket S3, gunakan operasi API `GetBucketOwnershipControls`. Untuk informasi selengkapnya, lihat GetBucketOwnershipControls.

Menonaktifkan ACLs semua bucket baru dan menegakkan Kepemilikan Objek

Kami menyarankan Anda menonaktifkan ACLs bucket Amazon S3 Anda. Anda dapat melakukannya dengan menerapkan pengaturan yang diberlakukan pemilik Bucket untuk Kepemilikan S3 Object. Ketika Anda menerapkan pengaturan ini, ACLs dinonaktifkan dan Anda secara otomatis memiliki dan memiliki kontrol penuh atas semua objek di bucket Anda. Untuk mewajibkan semua bucket baru dibuat dengan ACLs dinonaktifkan, gunakan kebijakan AWS Identity and Access Management (IAM) atau kebijakan kontrol AWS Organizations layanan (SCPs), seperti yang dijelaskan di bagian berikutnya.

Untuk menerapkan kepemilikan objek untuk objek baru tanpa menonaktifkan ACLs, Anda dapat menerapkan pengaturan pilihan pemilik Bucket. Saat menerapkan pengaturan ini, kami sangat menyarankan agar Anda memperbarui kebijakan bucket untuk `bucket-owner-full-control` mewajibkan ACL yang terekam untuk semua permintaan PUT ke bucket Anda. Pastikan Anda juga memperbarui klien Anda untuk mengirim ACL terekam `bucket-owner-full-control` ke bucket Anda dari akun lain.

Topik

- Menonaktifkan ACLs semua bucket baru (pemilik bucket diberlakukan)
- Memerlukan ACL bucket-owner-full-control kalengan untuk operasi Amazon PUT S3 (lebih disukai pemilik ember)

Menonaktifkan ACLs semua bucket baru (pemilik bucket diberlakukan)

Contoh berikut kebijakan IAM menolak izin `s3:CreateBucket` untuk pengguna atau peran IAM tertentu kecuali pengaturan yang diberlakukan pemilik Bucket diterapkan untuk Kepemilikan Objek. Pasangan kunci-nilai di dalam blok `Condition` menentukan `s3:x-amz-object-ownership`

sebagai kunci, dan pengaturan `BucketOwnerEnforced` sebagai nilainya. Dengan kata lain, pengguna IAM hanya dapat membuat bucket jika mereka menyetel pengaturan yang diterapkan pemilik Bucket untuk Kepemilikan Objek dan menonaktifkan. ACLs Anda juga dapat menggunakan kebijakan ini sebagai SCP batas untuk organisasi Anda. AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RequireBucketOwnerFullControl",
            "Action": "s3:CreateBucket",
            "Effect": "Deny",
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "s3:x-amz-object-ownership": "BucketOwnerEnforced"
                }
            }
        }
    ]
}
```

Memerlukan ACL bucket-owner-full-control kalengan untuk operasi Amazon **PUT** S3 (lebih disukai pemilik ember)

Dengan pengaturan pilihan pemilik Bucket untuk Kepemilikan Objek, Anda, sebagai pemilik bucket, memiliki dan memiliki kendali penuh atas objek baru yang ditulis akun lain ke bucket Anda dengan ACL terekam `bucket-owner-full-control`. Namun, jika akun lain menulis objek ke bucket Anda tanpa ACL terekam`bucket-owner-full-control`, penulis objek mempertahankan akses kontrol penuh. Anda, sebagai pemilik bucket, dapat menerapkan kebijakan bucket yang mengizinkan penulisan hanya jika mereka menentukan ACL terekam `bucket-owner-full-control`.

> **Note**
>
> Jika Anda telah ACLs menonaktifkan pengaturan yang diberlakukan pemilik Bucket, Anda, sebagai pemilik bucket, secara otomatis memiliki dan memiliki kendali penuh atas semua objek di bucket Anda. Anda tidak perlu menggunakan bagian ini untuk memperbarui kebijakan bucket guna menegakkan kepemilikan objek bagi pemilik bucket.

Kebijakan bucket berikut ini menentukan akun tersebut *111122223333* dapat unggah objek ke *amzn-s3-demo-bucket* hanya ketika objek ACL diatur ke `bucket-owner-full-control`. Pastikan untuk mengganti *111122223333* dengan akun Anda, dan *amzn-s3-demo-bucket* dengan nama bucket Anda.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "Only allow writes to my bucket with bucket owner full control",
         "Effect": "Allow",
         "Principal": {
            "AWS": [
               "arn:aws:iam::111122223333:user/ExampleUser"
            ]
         },
         "Action": [
            "s3:PutObject"
         ],
         "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
         "Condition": {
            "StringEquals": {
               "s3:x-amz-acl": "bucket-owner-full-control"
            }
         }
      }
   ]
}
```

Berikut ini adalah contoh operasi penyalinan yang menyertakan `bucket-owner-full-control` ACL terekam dengan menggunakan AWS Command Line Interface (AWS CLI).

```
aws s3 cp file.txt s3://amzn-s3-demo-bucket --acl bucket-owner-full-control
```

Setelah kebijakan bucket diberlakukan, jika klien tidak menyertakan ACL terekam `bucket-owner-full-control`, maka operasi akan gagal, dan pengunggah menerima kesalahan berikut:

Terjadi kesalahan (AccessDenied) saat memanggil PutObject operasi: Akses Ditolak.

> **Note**
>
> Jika klien perlu mengakses objek setelah mengunggah, maka Anda harus memberikan izin tambahan untuk akun pengunggah. Untuk informasi tentang bagaimana memberikan akun akses ke sumber daya Anda, lihat Panduan yang menggunakan kebijakan untuk mengelola akses ke sumber daya Amazon S3.

Pemecahan Masalah

Saat Anda menerapkan pengaturan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek S3, daftar kontrol akses (ACLs) dinonaktifkan dan Anda, sebagai pemilik bucket, secara otomatis memiliki semua objek di bucket Anda. ACLs tidak lagi memengaruhi izin untuk objek di bucket Anda. Anda dapat menggunakan kebijakan untuk memberikan izin. Semua permintaan PUT S3 harus menentukan ACL terekam `bucket-owner-full-control` atau tidak menentukan ACL sama sekali, atau permintaan ini akan gagal. Untuk informasi selengkapnya, lihat Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.

Jika ACL yang tidak valid ditentukan atau izin ACL bucket memberikan akses di luar Akun AWS Anda, Anda mungkin melihat respons kesalahan berikut.

AccessControlListNotSupported

Setelah Anda menerapkan pengaturan diberlakukan pemilik Bucket untuk Kepemilikan Objek, akan ACLs dinonaktifkan. Permintaan untuk mengatur ACLs atau memperbarui ACLs gagal dengan `400` kesalahan dan mengembalikan kode AccessControlListNotSupportedkesalahan. Permintaan untuk membaca ACLs masih didukung. Permintaan untuk membaca ACLs selalu menampilkan respons yang menunjukkan kontrol penuh untuk pemilik bucket. Dalam PUT operasi Anda, Anda harus menentukan kontrol penuh pemilik bucket ACLs atau tidak menentukan ACL. Jika tidak, operasi PUT Anda gagal.

Contoh `put-object` AWS CLI perintah berikut termasuk ACL `public-read` kalengan.

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key object-key-name --body doc-example-body --acl public-read
```

Jika bucket menggunakan setelan yang diberlakukan pemilik Bucket untuk menonaktifkan ACLs, operasi ini gagal, dan pengunggah menerima pesan galat berikut:

Terjadi kesalahan (AccessControlListNotSupported) saat memanggil PutObject operasi: Bucket tidak mengizinkan ACLs

InvalidBucketAclWithObjectOwnership

Jika Anda ingin menerapkan pengaturan yang diberlakukan pemilik Bucket untuk menonaktifkan ACLs, ACL bucket Anda harus memberikan kontrol penuh hanya kepada pemilik bucket. Bucket ACL Anda tidak dapat memberikan akses ke grup eksternal Akun AWS atau grup lainnya. Misalnya, jika `CreateBucket` permintaan Anda menetapkan diberlakukan pemilik Bucket dan menentukan ACL bucket yang menyediakan akses ke eksternal Akun AWS, permintaan Anda gagal dengan `400` kesalahan dan mengembalikan kode kesalahan. InvalidBucketAclWithObjectOwnership Demikian pula, jika permintaan `PutBucketOwnershipControls` Anda menetapkan pemilik Bucket yang diberlakukan pada bucket yang memiliki bucket ACL yang memberikan izin kepada orang lain, permintaan tersebut gagal.

Example : Bucket ACL yang ada memberikan akses baca publik

Misalnya, jika ACL bucket yang ada memberikan akses baca publik, Anda tidak dapat menerapkan pengaturan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek hingga Anda memigrasikan izin ACL ini ke kebijakan bucket dan mengatur ulang ACL bucket Anda ke ACL pribadi default. Untuk informasi selengkapnya, lihat Prasyarat untuk menonaktifkan ACLs.

Contoh bucket ACL ini memberikan akses baca publik:

```
{
    "Owner": {
        "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID"
    },
    "Grants": [
        {
            "Grantee": {
                "ID":
 "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID",
                "Type": "CanonicalUser"
            },
            "Permission": "FULL_CONTROL"
        },
        {
            "Grantee": {
                "Type": "Group",
```

```
                "URI": "http://acs.amazonaws.com/groups/global/AllUsers"
            },
            "Permission": "READ"
        }
    ]
}
```

`put-bucket-ownership-controls` AWS CLI Perintah contoh berikut menerapkan setelan diberlakukan pemilik Bucket untuk Kepemilikan Objek:

```
aws s3api put-bucket-ownership-controls --bucket amzn-s3-demo-bucket --ownership-
controls Rules=[{ObjectOwnership=BucketOwnerEnforced}]
```

Karena bucket ACL memberikan akses baca publik, permintaan gagal dan mengembalikan kode kesalahan berikut:

Terjadi kesalahan (InvalidBucketAclWithObjectOwnership) saat memanggil PutBucketOwnershipControls operasi: Bucket tidak dapat ACLs disetel dengan ObjectOwnership BucketOwnerEnforced pengaturan

Keamanan di Amazon S3

Keamanan cloud di AWS adalah prioritas tertinggi. 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 AWS dan Anda. Model tanggung jawab bersama menjelaskan hal ini sebagai keamanan cloud dan keamanan dalam cloud:

Keamanan cloud

AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Efektivitas keamanan kami diuji dan diverifikasi secara rutin oleh auditor pihak ketiga sebagai bagian dari program kepatuhan AWS. Untuk mempelajari tentang program kepatuhan yang berlaku untuk Amazon S3, lihat AWS Layanan dalam Cakupan Program Kepatuhan.

Keamanan cloud

Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor-faktor lain termasuk sensitivitas data, kebutuhan organisasi, serta undang-undang dan peraturan yang berlaku. Untuk Amazon S3, tanggung jawab Anda mencakup area berikut:

- Mengelola data Anda, termasuk kepemilikan objek dan enkripsi.
- Mengklasifikasikan aset Anda.
- Mengelola akses ke data Anda menggunakan peran IAM dan konfigurasi layanan lainnya untuk menerapkan izin yang sesuai.
- Mengaktifkan kontrol detektif seperti atau Amazon GuardDuty untuk AWS CloudTrailAmazon S3.

Dokumentasi ini akan membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan Amazon S3. Topik berikut menunjukkan cara mengonfigurasi Amazon S3 untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga akan mempelajari cara menggunakan AWS layanan lain yang dapat membantu Anda memantau dan mengamankan sumber daya Amazon S3 Anda.

> **Note**
>
> Untuk informasi selengkapnya tentang menggunakan kelas penyimpanan Amazon S3 Express One Zone dengan bucket direktori, lihat S3 Express One Zone dan Bekerja dengan bucket direktori.

Topik

- Praktik Terbaik Keamanan untuk Amazon S3
- Perlindungan data dalam Amazon S3
- Melindungi data dengan enkripsi
- Privasi lalu lintas antarjaringan
- Validasi kepatuhan untuk Amazon S3
- Ketahanan dalam Amazon S3
- Keamanan infrastruktur di Amazon S3
- Analisis konfigurasi dan kerentanan di Amazon S3
- Manajemen akses

Praktik Terbaik Keamanan untuk Amazon S3

Amazon S3 menyediakan sejumlah fitur keamanan untuk dipertimbangkan ketika Anda mengembangkan dan menerapkan kebijakan keamanan Anda sendiri. Praktik terbaik berikut adalah pedoman umum dan tidak mewakili solusi keamanan yang lengkap. Karena praktik terbaik ini mungkin tidak sesuai atau tidak memadai untuk lingkungan Anda, perlakukan itu sebagai rekomendasi yang bermanfaat, bukan sebagai resep.

Topik

- Praktik terbaik keamanan Amazon S3
- Praktik Terbaik Pemantauan dan Audit Amazon S3
- Memantau keamanan data dengan layanan AWS keamanan terkelola

Praktik terbaik keamanan Amazon S3

Praktik terbaik berikut untuk Amazon S3 dapat membantu mencegah insiden keamanan.

Nonaktifkan daftar kontrol akses (ACLs)

S3 Object Ownership adalah setelan tingkat ember Amazon S3 yang dapat Anda gunakan untuk mengontrol kepemilikan objek yang diunggah ke bucket Anda dan untuk menonaktifkan atau mengaktifkan. ACLs Secara default, Kepemilikan Objek disetel ke setelan diberlakukan pemilik Bucket dan semuanya ACLs dinonaktifkan. Saat ACLs dinonaktifkan, pemilik bucket memiliki semua objek di bucket dan mengelola akses ke data secara eksklusif menggunakan kebijakan manajemen akses.

Mayoritas kasus penggunaan modern di Amazon S3 tidak lagi memerlukan penggunaan daftar kontrol akses () ACLs. Kami menyarankan Anda menonaktifkan ACLs, kecuali dalam keadaan yang tidak biasa di mana Anda harus mengontrol akses untuk setiap objek secara individual. Untuk menonaktifkan ACLs dan mengambil kepemilikan setiap objek di bucket Anda, terapkan pengaturan yang diberlakukan pemilik bucket untuk Kepemilikan Objek S3. Saat Anda menonaktifkan ACLs, Anda dapat dengan mudah memelihara ember dengan objek yang diunggah oleh berbeda Akun AWS.

Kapan ACLs dinonaktifkan kontrol akses untuk data Anda didasarkan pada kebijakan, seperti berikut ini:

- AWS Identity and Access Management (IAM) kebijakan pengguna

- Kebijakan bucket S3
- Kebijakan titik akhir cloud privat virtual (VPC)
- AWS Organizations kebijakan kontrol layanan (SCPs)
- AWS Organizations kebijakan kontrol sumber daya (RCPs)

Menonaktifkan ACLs menyederhanakan manajemen izin dan audit. ACLs dinonaktifkan untuk ember baru secara default. Anda juga dapat menonaktifkan ACLs untuk ember yang ada. Jika Anda memiliki bucket yang sudah memiliki objek di dalamnya, setelah Anda menonaktifkan ACLs, objek dan bucket ACLs tidak lagi menjadi bagian dari proses evaluasi akses. Sebaliknya, akses diberikan atau ditolak berdasarkan kebijakan.

Sebelum Anda menonaktifkan ACLs, pastikan Anda melakukan hal berikut:

- Tinjau kebijakan bucket Anda untuk memastikan bahwa kebijakan tersebut mencakup semua cara yang ingin Anda berikan akses ke bucket di luar akun Anda.
- Setel ulang ACL bucket Anda ke default (kontrol penuh ke pemilik bucket).

Setelah Anda menonaktifkan ACLs, perilaku berikut terjadi:

- Bucket Anda hanya menerima PUT permintaan yang tidak menentukan ACL atau PUT permintaan dengan kontrol penuh pemilik bucket. ACLs Ini ACLs termasuk ACL `bucket-owner-full-control` kalengan atau bentuk setara dari ACL ini yang diekspresikan dalam XML.
- Aplikasi yang ada yang mendukung kontrol penuh pemilik bucket ACLs tidak akan berdampak.
- PUTpermintaan yang berisi lainnya ACLs (misalnya, hibah khusus untuk tertentu Akun AWS) gagal dan mengembalikan kode status HTTP `400 (Bad Request)` dengan kode `AccessControlListNotSupported` kesalahan.

Untuk informasi selengkapnya, lihat Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.

Pastikan bucket Amazon S3 Anda menggunakan kebijakan yang benar dan tidak dapat diakses publik

Kecuali jika Anda secara tegas meminta siapa pun di internet untuk dapat membaca atau menulis ke bucket S3 Anda, pastikan bucket S3 Anda tidak tersedia untuk umum. Berikut ini adalah beberapa langkah yang dapat Anda lakukan untuk memblokir akses publik:

- Gunakan Blokir Akses Publik S3. Dengan Blokir Akses Publik S3, Anda dapat dengan mudah menyiapkan kontrol terpusat untuk membatasi akses publik ke sumber daya Amazon S3 Anda. Kontrol terpusat ini ditegakkan terlepas dari bagaimana sumber daya dibuat. Untuk informasi selengkapnya, lihat Melakukan blok akses publik ke penyimpanan Amazon S3 Anda.
- Identifikasi kebijakan bucket Amazon S3 yang mengizinkan identitas wildcard seperti `"Principal": "*"` (yang secara efektif berarti “siapa pun”). Cari juga kebijakan yang memungkinkan tindakan wildcard `"*"` (yang secara efektif memungkinkan pengguna melakukan tindakan apa pun di bucket Amazon S3).
- Demikian pula, cari daftar kontrol akses bucket Amazon S3 (ACLs) yang menyediakan baca, tulis, atau akses penuh ke “Semua Orang” atau “Setiap pengguna yang diautentikasi AWS .”
- Menggunakan operasi API `ListBuckets` untuk memindai semua bucket Amazon S3. Kemudian, gunakan `GetBucketAcl`, `GetBucketWebsite`, dan `GetBucketPolicy` untuk menentukan apakah setiap bucket memiliki kontrol akses yang sesuai dan konfigurasi yang sesuai.
- Gunakan AWS Trusted Advisor untuk memeriksa implementasi Amazon S3.
- Pertimbangkan untuk menerapkan kontrol detektif yang sedang berlangsung dengan menggunakan s3-bucket-public-read-prohibited dan s3-bucket-public-write-prohibited Aturan AWS Config yang dikelola.

Untuk informasi selengkapnya, lihat Identity and Access Management untuk Amazon S3.

Terapkan akses hak akses paling rendah

Saat memberikan izin, Anda memutuskan siapa yang mendapatkan izin yang menjadi sumber daya Amazon S3. Anda mengaktifkan tindakan tertentu yang ingin Anda izinkan pada sumber daya tersebut. Oleh karena itu, kami menyarankan Anda hanya memberikan izin yang diperlukan untuk melaksanakan tugas. Menerapkan akses hak akses paling rendah adalah hal mendasar dalam mengurangi risiko keamanan dan dampak yang dapat diakibatkan oleh kesalahan atau niat jahat.

Alat bantu berikut tersedia untuk menerapkan akses hak akses paling rendah:

- Tindakan kebijakan untuk Amazon S3 dan Batasan Izin untuk Entitas IAM
- Bagaimana Amazon S3 bekerja dengan IAM
- Gambaran umum daftar kontrol akses (ACL)

Untuk panduan tentang apa yang harus dipertimbangkan ketika memilih satu atau beberapa mekanisme sebelumnya, lihat Identity and Access Management untuk Amazon S3.

Gunakan peran IAM untuk aplikasi dan Layanan AWS yang memerlukan akses Amazon S3

Agar aplikasi yang berjalan di Amazon EC2 atau lainnya dapat Layanan AWS mengakses sumber daya Amazon S3, mereka harus menyertakan AWS kredensi yang valid dalam permintaan API mereka. AWS Sebaiknya jangan menyimpan AWS kredensialnya secara langsung di aplikasi atau instans Amazon EC2 . Ini adalah kredensial jangka panjang yang tidak dirotasi secara otomatis dan dapat menimbulkan dampak bisnis yang signifikan jika dibobol.

Sebaliknya, gunakan peran IAM untuk mengelola kredensial sementara untuk aplikasi atau layanan yang perlu mengakses Amazon S3. Saat Anda menggunakan peran, Anda tidak perlu mendistribusikan kredensi jangka panjang (seperti nama pengguna dan kata sandi atau kunci akses) ke EC2 instans Amazon atau Layanan AWS, seperti. AWS Lambda Peran ini menyediakan izin sementara yang dapat digunakan aplikasi saat mereka melakukan panggilan ke AWS sumber daya lain.

Untuk informasi selengkapnya, lihat topik berikut di Panduan Pengguna IAM:

- Peran IAM
- Skenario Umum untuk Peran: Pengguna, Aplikasi, dan Layanan

Pertimbangkan untuk mengenkripsi data diam

Anda memiliki opsi berikut untuk melindungi data diam di Amazon S3:

- Enkripsi sisi server — Semua bucket Amazon S3 memiliki enkripsi yang dikonfigurasi secara default, dan semua objek baru yang diunggah ke bucket S3 secara otomatis dienkripsi saat istirahat. Enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3) adalah konfigurasi enkripsi default untuk setiap bucket di Amazon S3. Untuk menggunakan jenis enkripsi yang berbeda, Anda dapat menentukan jenis enkripsi di sisi server yang akan digunakan dalam permintaan PUT S3, atau Anda dapat mengatur konfigurasi enkripsi default di bucket tujuan.

  Amazon S3 juga menyediakan opsi enkripsi sisi server ini:

  - Enkripsi sisi server dengan kunci AWS Key Management Service (AWS KMS) (SSE-KMS)
  - Enkripsi sisi server dua lapis dengan kunci () (DSSE-KMS AWS Key Management Service)AWS KMS
  - Enkripsi di sisi server dengan kunci yang disediakan pelanggan (SSE-C)

  Untuk informasi selengkapnya, lihat Melindungi data dengan enkripsi di sisi klien.
- Enkripsi di sisi klien–Enkripsikan data sisi klien dan unggah data yang dienkripsi ke Amazon S3. Dalam hal ini, Anda mengelola proses enkripsi, kunci enkripsi, dan alat terkait. Seperti enkripsi

di sisi server, enkripsi di sisi klien dapat membantu mengurangi risiko dengan mengenkripsi data dengan kunci yang disimpan dalam mekanisme yang berbeda dari mekanisme yang menyimpan data itu sendiri.

Amazon S3 menyediakan beberapa opsi enkripsi di sisi klien. Untuk informasi selengkapnya, lihat Melindungi data menggunakan enkripsi di sisi klien.

Menerapkan enkripsi data bergerak

Anda dapat menggunakan HTTPS (TLS) untuk membantu mencegah penyerang potensial menguping atau memanipulasi lalu lintas jaringan dengan menggunakan atau serangan serupa. person-in-the-middle Sebaiknya izinkan koneksi terenkripsi melalui HTTPS (TLS) dengan menggunakan kondisi aws:SecureTransport dalam kebijakan bucket Amazon S3 Anda. Untuk informasi selengkapnya, lihat contoh kebijakan bucket S3 Mengelola akses berdasarkan permintaan HTTP atau HTTPS. Selain menolak permintaan HTTP, kami menyarankan Anda mengatur CloudWatch alarm Amazon pada peringatan `tlsDetails.tlsVersion NOT EXISTS` itu jika upaya akses HTTP dilakukan pada konten Anda. Untuk informasi selengkapnya tentang cara mengonfigurasi CloudWatch alarm Amazon, lihat Membuat CloudWatch alarm untuk CloudTrail peristiwa: contoh dan CloudTrail merekam konten di AWS CloudTrail Panduan Pengguna.

> **Important**
>
> Kami menyarankan agar aplikasi Anda tidak menyematkan sertifikat Amazon S3 TLS karena AWS tidak mendukung penyematan sertifikat tepercaya publik. S3 secara otomatis memperbarui sertifikat dan perpanjangan dapat terjadi kapan saja sebelum sertifikat kedaluwarsa. Memperbarui sertifikat menghasilkan key pair public-private baru. Jika Anda telah menyematkan sertifikat S3 yang baru saja diperbarui dengan kunci publik baru, Anda tidak akan dapat terhubung ke S3 hingga aplikasi Anda menggunakan sertifikat baru.

Lalu, pertimbangkan untuk menerapkan kontrol detektif yang sedang berlangsung dengan menggunakan aturan s3-bucket-ssl-requests-only AWS Config terkelola.

Pertimbangkan untuk menggunakan Kunci Objek S3

Dengan Kunci Objek S3, Anda dapat menyimpan objek dengan menggunakan model “Tulis Sekali Baca Banyak” (WORM). Kunci Objek S3 dapat membantu mencegah penghapusan data

yang tidak disengaja atau tidak tepat. Misalnya, Anda dapat menggunakan S3 Object Lock untuk membantu melindungi AWS CloudTrail log Anda.

Untuk informasi selengkapnya, lihat Mengunci objek dengan Object Lock.

Aktifkan Penentuan Versi S3

Penentuan Versi S3 adalah cara menyimpan beberapa varian objek dalam bucket yang sama. Anda dapat menggunakan Penentuan Versi untuk menyimpan, mengambil, dan memulihkan setiap versi dari setiap objek yang disimpan dalam bucket Anda. Dengan Penentuan Versi, Anda dapat dengan mudah memulihkan dari tindakan pengguna yang tidak diinginkan, serta kegagalan aplikasi.

Lalu, pertimbangkan untuk menerapkan kontrol detektif yang sedang berlangsung dengan menggunakan aturan s3-bucket-versioning-enabled AWS Config terkelola.

Untuk informasi selengkapnya, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.

Pertimbangkan untuk menggunakan Replikasi Lintas Wilayah S3

Meskipun Amazon S3 menyimpan data Anda di berbagai Zona Ketersediaan yang berbeda secara geografis secara default, persyaratan kepatuhan mungkin mengatur bahwa Anda menyimpan data dengan jarak lebih jauh. Dengan S3 Cross-Region Replication (CRR), Anda dapat mereplikasi data antara jarak jauh Wilayah AWS untuk membantu memenuhi persyaratan ini. CRR memungkinkan penyalinan objek secara otomatis dan asinkron di seluruh ember dalam berbagai jenis. Wilayah AWS Untuk informasi selengkapnya, lihat Mereplikasi objek di dalam dan di seluruh Wilayah.

> **Note**
>
> CRR mewajibkan bucket S3 sumber dan tujuan agar Penentuan Versi diaktifkan.

Lalu, pertimbangkan untuk menerapkan kontrol detektif yang sedang berlangsung dengan menggunakan aturan s3-bucket-replication-enabled AWS Config terkelola.

Pertimbangkan untuk menggunakan titik akhir VPC untuk akses Amazon S3

Titik akhir cloud privat virtual (VPC) untuk Amazon S3 adalah entitas logis dalam sebuah VPC yang mengizinkan konektivitas hanya ke Amazon S3. Titik akhir VPC dapat membantu mencegah lalu lintas melintasi internet terbuka.

Titik akhir VPC untuk Amazon S3 menyediakan banyak cara untuk mengontrol akses ke data Amazon S3 Anda:

- Anda dapat mengontrol permintaan, pengguna, atau grup yang diizinkan melalui titik akhir VPC tertentu dengan menggunakan kebijakan bucket S3.
- Anda dapat mengontrol titik akhir VPCs atau VPC mana yang memiliki akses ke bucket S3 Anda dengan menggunakan kebijakan bucket S3.
- Anda dapat membantu mencegah eksfiltrasi data dengan menggunakan VPC yang tidak memiliki gateway internet.

Untuk informasi selengkapnya, lihat Mengendalikan akses dari titik akhir VPC dengan kebijakan bucket.

Menggunakan layanan AWS keamanan terkelola untuk memantau keamanan data

Beberapa layanan AWS keamanan terkelola dapat membantu Anda mengidentifikasi, menilai, dan memantau risiko keamanan dan kepatuhan untuk data Amazon S3 Anda. Layanan ini juga dapat membantu Anda melindungi data Anda dari risiko tersebut. Layanan ini mencakup kemampuan deteksi, pemantauan, dan perlindungan otomatis yang dirancang untuk menskalakan dari sumber daya Amazon S3 untuk sumber daya tunggal Akun AWS untuk organisasi yang mencakup ribuan akun.

Untuk informasi selengkapnya, lihat Memantau keamanan data dengan layanan AWS keamanan terkelola.

Praktik Terbaik Pemantauan dan Audit Amazon S3

Praktik terbaik berikut untuk Amazon S3 dapat membantu mendeteksi potensi kelemahan dan insiden keamanan.

Identifikasi dan audit semua bucket Amazon S3 Anda

Identifikasi aset IT Anda adalah aspek penting dari tata kelola dan keamanan. Anda perlu memiliki visibilitas semua sumber daya Amazon S3 Anda untuk menilai postur keamanan dan mengambil tindakan di area kelemahan potensial. Untuk mengaudit sumber daya Anda, kami sarankan melakukan hal berikut ini:

- Gunakan Editor Tag untuk mengidentifikasi dan menandai sumber daya yang sensitif terhadap keamanan atau audit, kemudian gunakan tag tersebut saat Anda perlu mencari sumber daya

ini. Untuk informasi selengkapnya, lihat Mencari Sumber Daya untuk Ditandai di Panduan Pengguna AWS Sumber Daya Penandaan.

- Gunakan Inventaris S3 untuk mengaudit dan melaporkan replikasi dan status enkripsi objek Anda untuk kebutuhan bisnis, kepatuhan, dan regulasi. Untuk informasi selengkapnya, lihat Membuat katalog dan menganalisis data Anda dengan S3 Inventory.
- Buat grup sumber daya untuk sumber daya Amazon S3 Anda. Untuk informasi selengkapnya, lihat Apa yang dimaksud dengan grup sumber daya? dalam AWS Resource Groups Panduan Pengguna.

Melaksanakan pemantauan dengan menggunakan alat AWS pemantauan

Pemantauan adalah bagian penting dalam menjaga keandalan, keamanan, ketersediaan, dan kinerja Amazon S3 dan solusi Anda AWS . AWS menyediakan beberapa alat dan layanan untuk membantu Anda memantau Amazon S3 dan yang lain. Layanan AWS Misalnya, Anda dapat memantau CloudWatch metrik Amazon untuk Amazon S3, khususnya,, `PutRequests` dan `GetRequests` metrik. `4xxErrors` `DeleteRequests` Untuk informasi selengkapnya, lihat Memantau metrik dengan Amazon CloudWatch dan Pencatatan log dan pemantauan di Amazon S3.

Untuk contoh kedua, lihat Contoh: Aktivitas Bucket Amazon S3. Contoh ini menjelaskan cara membuat CloudWatch alarm yang dipicu saat panggilan API Amazon S3 dibuat `PUT` atau `DELETE` kebijakan bucket, siklus hidup bucket, atau konfigurasi replikasi bucket, atau ke ACL bucket. `PUT`

Aktifkan log akses server Amazon S3

Pencatatan akses server memberikan catatan detail permintaan yang dibuat ke bucket. Log akses server dapat membantu Anda dalam keamanan dan audit akses, membantu Anda mempelajari basis pelanggan Anda, dan memahami tagihan Amazon S3. Untuk petunjuk tentang mengaktifkan log akses server, lihat Pencatatan permintaan dengan pencatatan akses server.

Juga pertimbangkan untuk menerapkan kontrol detektif yang sedang berlangsung dengan menggunakan aturan s3-bucket-logging-enabled AWS Config terkelola.

Gunakan AWS CloudTrail

AWS CloudTrail menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau Layanan AWS di Amazon S3. Anda dapat menggunakan informasi yang dikumpulkan oleh CloudTrail untuk menentukan hal-hal berikut:

- Permintaan yang diajukan ke Amazon S3
- Alamat IP dari mana permintaan itu dibuat

- Siapa yang membuat permintaan
- Kapan permintaan dibuat
- Detail tambahan tentang permintaan

Misalnya, Anda dapat mengidentifikasi CloudTrail entri untuk PUT tindakan yang memengaruhi akses data, khususnya`PutBucketAcl`, `PutObjectAclPutBucketPolicy`, dan`PutBucketWebsite`.

Saat Anda mengatur Akun AWS, CloudTrail diaktifkan secara default. Anda dapat melihat peristiwa terbaru di CloudTrail konsol. Untuk membuat catatan aktivitas dan peristiwa yang sedang berlangsung untuk bucket Amazon S3, Anda dapat membuat jejak di konsol CloudTrail . Untuk informasi selengkapnya, lihat Mencatat peristiwa data dalam AWS CloudTrail Panduan Pengguna.

Saat membuat jejak, Anda dapat mengonfigurasi CloudTrail untuk mencatat peristiwa data. Peristiwa data adalah catatan operasi sumber daya yang dilakukan pada atau di dalam sumber daya. Di Amazon S3, peristiwa data merekam aktivitas API tingkat objek untuk masing-masing bucket. CloudTrail mendukung subset operasi API tingkat objek Amazon S3, sepertiGetObject,, dan. DeleteObject PutObject Untuk informasi selengkapnya tentang cara CloudTrail bekerja dengan Amazon S3, lihat. Mencatat panggilan API Amazon S3 menggunakan AWS CloudTrail Di konsol Amazon S3, Anda juga dapat mengonfigurasi bucket S3 Anda ke Mengaktifkan pencatatan CloudTrail peristiwa untuk bucket dan objek S3.

AWS Config menyediakan aturan terkelola (`cloudtrail-s3-dataevents-enabled`) yang dapat Anda gunakan untuk mengonfirmasi bahwa setidaknya satu CloudTrail jejak mencatat peristiwa data untuk bucket S3 Anda. Untuk informasi lebih lanjut, lihat cloudtrail-s3-dataevents-enabled dalam Panduan Pengembang AWS Config .

Aktifkan AWS Config

Beberapa praktik terbaik yang tercantum dalam topik ini menyarankan untuk membuat AWS Config aturan. AWS Config membantu Anda menilai, mengaudit, dan mengevaluasi konfigurasi AWS sumber daya Anda. AWS Config memantau konfigurasi sumber daya sehingga Anda dapat mengevaluasi konfigurasi yang direkam terhadap konfigurasi aman yang diinginkan. Dengan AWS Config, Anda dapat melakukan hal berikut:

- Tinjau perubahan konfigurasi dan hubungan antar sumber daya AWS
- Selidiki riwayat konfigurasi sumber daya terperinci

- Tentukan kepatuhan Anda secara keseluruhan terhadap konfigurasi yang ditentukan dalam pedoman internal Anda

Menggunakan AWS Config dapat membantu Anda menyederhanakan audit kepatuhan, analisis keamanan, manajemen perubahan, dan pemecahan masalah operasional. Untuk informasi selengkapnya, lihat Menyiapkan AWS Config dengan Konsol di Panduan AWS Config Pengembang. Saat menentukan jenis sumber daya yang akan dicatat, pastikan Anda menyertakan sumber daya Amazon S3.

> **Important**
>
> AWS Config aturan terkelola hanya mendukung bucket tujuan umum saat mengevaluasi sumber daya Amazon S3. AWS Config tidak merekam perubahan konfigurasi untuk bucket direktori. Untuk informasi selengkapnya, lihat Aturan AWS Config Terkelola dan Daftar Aturan AWS Config Terkelola di Panduan AWS Config Pengembang.

Untuk contoh cara menggunakan AWS Config, lihat Cara Menggunakan AWS Config untuk Memantau dan Menanggapi Bucket Amazon S3 yang Mengizinkan Akses Publik di Blog Keamanan.AWS

Menggunakan Lensa Penyimpanan S3

Lensa Penyimpanan S3 adalah fitur analisis penyimpanan cloud yang dapat Anda gunakan untuk mendapatkan visibilitas seluruh organisasi ke dalam penggunaan dan aktivitas penyimpanan objek. Lensa Penyimpanan S3 juga menganalisis metrik untuk memberikan rekomendasi kontekstual yang dapat Anda gunakan untuk mengoptimalkan biaya penyimpanan dan menerapkan praktik terbaik untuk melindungi data Anda.

Dengan Lensa Penyimpanan S3, Anda dapat menggunakan metrik untuk menghasilkan wawasan ringkasan, seperti mencari tahu berapa banyak penyimpanan yang Anda miliki di seluruh organisasi atau bucket dan prefiks mana yang paling cepat berkembang. Anda juga dapat menggunakan metrik Lensa Penyimpanan S3 untuk mengidentifikasi peluang optimasi biaya, menerapkan praktik terbaik untuk perlindungan data dan keamanan, serta meningkatkan kinerja beban kerja aplikasi.

Misalnya, Anda dapat mengidentifikasi bucket yang tidak memiliki aturan Siklus Hidup S3 untuk membatalkan unggahan multibagian yang tidak lengkap yang berusia lebih dari 7 hari. Anda juga dapat mengidentifikasi bucket yang tidak mengikuti praktik terbaik perlindungan data, seperti

menggunakan Replikasi S3 atau Penentuan Versi S3. Untuk informasi selengkapnya, lihat Memahami Lensa Penyimpanan Amazon S3.

Memantau laporan keamanan AWS

Kami menyarankan Anda secara teratur memeriksa nasihat keamanan yang diposting dalam Trusted Advisor untuk Akun AWS Anda. Secara khusus, cari peringatan tentang bucket Amazon S3 dengan “buka izin akses.” Anda dapat melakukan ini secara terprogram dengan menggunakan describe-trusted-advisor-checks.

Selanjutnya, secara aktif memantau alamat email utama yang terdaftar untuk masing-masing Anda Akun AWS. AWS menggunakan alamat email ini untuk menghubungi Anda tentang masalah keamanan yang muncul yang mungkin memengaruhi Anda.

AWS masalah operasional dengan dampak luas diposting di AWS Health Dashboard - Layanan kesehatan. Masalah operasional juga diposting ke akun individu melalui AWS Health Dashboard. Untuk informasi lebih lanjut, lihat dokumentasi AWS Health.

Memantau keamanan data dengan layanan AWS keamanan terkelola

Beberapa layanan AWS keamanan terkelola dapat membantu Anda mengidentifikasi, menilai, dan memantau risiko keamanan dan kepatuhan untuk data Amazon S3 Anda. Mereka juga dapat membantu Anda melindungi data Anda dari risiko tersebut. Layanan ini mencakup kemampuan deteksi, pemantauan, dan perlindungan otomatis yang dirancang untuk menskalakan dari sumber daya Amazon S3 untuk sumber daya tunggal Akun AWS hingga sumber daya untuk organisasi yang mencakup ribuan. Akun AWS

AWS Layanan deteksi dan respons dapat membantu Anda mengidentifikasi potensi kesalahan konfigurasi keamanan, ancaman, atau perilaku tak terduga, sehingga Anda dapat dengan cepat merespons aktivitas yang berpotensi tidak sah atau berbahaya di lingkungan Anda. AWS Layanan perlindungan data dapat membantu Anda memantau dan melindungi data, akun, dan beban kerja Anda dari akses yang tidak sah. Mereka juga dapat membantu Anda menemukan data sensitif, seperti informasi pengenal pribadi (PII), di properti data Amazon S3 Anda.

Untuk membantu Anda mengidentifikasi dan mengevaluasi risiko keamanan dan kepatuhan data, layanan keamanan terkelola AWS menghasilkan temuan untuk memberi tahu Anda tentang potensi peristiwa keamanan atau masalah dengan data Amazon S3 Anda. Temuan ini memberikan rincian relevan yang dapat Anda gunakan untuk menyelidiki, menilai, dan menindaklanjuti risiko ini sesuai dengan alur kerja dan kebijakan respons insiden Anda. Anda dapat mengakses data temuan secara

langsung dengan menggunakan setiap layanan. Anda juga dapat mengirim data ke aplikasi, layanan, dan sistem lain, seperti insiden keamanan dan sistem manajemen peristiwa (SIEM).

Untuk memantau keamanan data Amazon S3 Anda, pertimbangkan untuk menggunakan layanan AWS keamanan terkelola ini.

Amazon GuardDuty

Amazon GuardDuty adalah layanan pendeteksi ancaman yang terus memantau beban kerja Anda Akun AWS dan beban kerja untuk aktivitas berbahaya dan memberikan temuan keamanan terperinci untuk visibilitas dan remediasi.

Dengan fitur perlindungan S3 di GuardDuty, Anda dapat mengonfigurasi GuardDuty untuk menganalisis AWS CloudTrail manajemen dan peristiwa data untuk sumber daya Amazon S3 Anda. GuardDuty kemudian memantau peristiwa tersebut untuk aktivitas berbahaya dan mencurigakan. Untuk menginformasikan analisis dan mengidentifikasi potensi risiko keamanan, GuardDuty gunakan umpan intelijen ancaman dan pembelajaran mesin.

GuardDuty dapat memantau berbagai jenis aktivitas untuk sumber daya Amazon S3 Anda. Misalnya, peristiwa CloudTrail manajemen untuk Amazon S3 mencakup operasi tingkat ember, seperti,, dan. `ListBuckets` `DeleteBucket` `PutBucketReplication` CloudTrail peristiwa data untuk Amazon S3 mencakup operasi tingkat objek, seperti`GetObject`,, dan. `ListObjects` `PutObject` Jika GuardDuty mendeteksi aktivitas anomali atau berpotensi berbahaya, itu menghasilkan temuan untuk memberi tahu Anda.

Untuk informasi selengkapnya, lihat Perlindungan Amazon S3 GuardDuty di Amazon di GuardDuty Panduan Pengguna Amazon.

Amazon Detective

Amazon Detective menyederhanakan proses investigasi dan membantu Anda melakukan investigasi keamanan yang lebih cepat dan efektif. Detective menyediakan agregasi data, ringkasan, dan konteks yang dapat membantu Anda menganalisis dan menilai sifat dan tingkat kemungkinan masalah keamanan.

Detective secara otomatis mengekstrak peristiwa berbasis waktu, seperti panggilan API dari dan AWS CloudTrail Amazon VPC Flow Logs untuk sumber daya Anda. AWS Ini juga menelan temuan yang dihasilkan oleh Amazon GuardDuty. Detective kemudian menggunakan machine learning, analisis statistik, dan teori grafik untuk menghasilkan visualisasi yang membantu Anda melakukan investigasi keamanan yang efektif dengan lebih cepat.

Visualisasi ini memberikan pandangan interaktif yang terpadu tentang perilaku sumber daya dan interaksi di antara mereka dari waktu ke waktu. Anda dapat menjelajahi grafik perilaku ini untuk memeriksa tindakan yang berpotensi berbahaya, seperti upaya login yang gagal atau panggilan API yang mencurigakan. Anda juga dapat melihat bagaimana tindakan ini memengaruhi sumber daya, seperti bucket dan S3 Object.

Untuk informasi selengkapnya, lihat Panduan Administrasi Amazon Detective.

IAM Access Analyzer

AWS Identity and Access Management Access Analyzer (IAM Access Analyzer) dapat membantu Anda mengidentifikasi sumber daya yang dibagikan dengan entitas eksternal. Anda juga dapat menggunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM terhadap tata bahasa kebijakan dan praktik terbaik, serta menghasilkan kebijakan IAM berdasarkan aktivitas akses di log Anda. AWS CloudTrail

IAM Access Analyzer menggunakan penalaran berbasis logika untuk menganalisis kebijakan sumber daya di AWS lingkungan Anda, seperti kebijakan bucket. Dengan IAM Access Analyzer untuk S3, Anda akan diberi tahu saat bucket S3 dikonfigurasi untuk memungkinkan akses ke siapa pun di internet atau lainnya Akun AWS, termasuk akun di luar organisasi Anda. Misalnya, Penganalisis Akses IAM untuk S3 dapat melaporkan bahwa suatu bucket telah membaca atau menulis akses yang disediakan melalui daftar kontrol akses (ACL) bucket, kebijakan bucket, kebijakan titik akses Multi-Wilayah atau kebijakan titik akses. Untuk setiap bucket publik atau bucket bersama, Anda akan menerima temuan yang menunjukkan sumber dan tingkat akses publik atau bersama. Dengan temuan ini, Anda dapat segera mengambil tindakan korektif yang tepat untuk memulihkan akses bucket ke apa yang Anda inginkan.

Untuk informasi selengkapnya, lihat Meninjau akses bucket menggunakan IAM Access Analyzer untuk S3.

Amazon Macie

Amazon Macie adalah layanan keamanan yang menemukan data sensitif dengan menggunakan machine learning dan pencocokan pola. Macie memberikan visibilitas ke dalam risiko keamanan data, dan memungkinkan perlindungan otomatis terhadap risiko tersebut. Dengan Macie, Anda dapat mengotomatisasi penemuan dan pelaporan data sensitif di properti data Amazon S3 untuk mendapatkan pemahaman yang lebih baik tentang data yang disimpan organisasi Anda di S3.

Untuk mendeteksi data sensitif dengan Macie, Anda dapat menggunakan kriteria dan teknik bawaan yang dirancang untuk mendeteksi daftar jenis data sensitif yang besar dan berkembang

untuk banyak negara dan wilayah. Jenis data sensitif ini mencakup beberapa jenis informasi pengenal pribadi (PII), data keuangan, dan data kredensial. Anda juga dapat menggunakan kriteria kustom yang Anda tetapkan, ekspresi reguler yang menentukan pola teks yang cocok dan, opsional, urutan karakter dan aturan kedekatan yang menyempurnakan hasil.

Jika Macie mendeteksi data sensitif dalam S3 Object, Macie akan membuat temuan keamanan untuk memberi tahu Anda. Temuan ini memberikan informasi tentang objek yang terpengaruh, jenis dan jumlah kemunculan data sensitif yang ditemukan Macie, dan detail tambahan untuk membantu Anda menyelidiki bucket dan S3 Object yang terpengaruh. Untuk informasi selengkapnya, lihat Panduan Pengguna Amazon Macie.

AWS Security Hub

AWS Security Hub adalah layanan manajemen postur keamanan yang melakukan pemeriksaan praktik terbaik keamanan, mengumpulkan peringatan dan temuan dari berbagai sumber ke dalam satu format, dan memungkinkan remediasi otomatis.

Security Hub mengumpulkan dan menyediakan data temuan keamanan dari solusi AWS Partner Network keamanan terintegrasi dan Layanan AWS, termasuk Amazon Detective, Amazon, IAM Access Analyzer, dan GuardDuty Amazon Macie. Ini juga menghasilkan temuannya sendiri dengan menjalankan pemeriksaan keamanan otomatis yang berkelanjutan berdasarkan praktik AWS terbaik dan standar industri yang didukung.

Security Hub kemudian mengkorelasikan dan mengkonsolidasikan temuan di seluruh penyedia untuk membantu Anda memprioritaskan dan memproses temuan yang paling signifikan. Ini juga menyediakan dukungan untuk tindakan kustom, yang dapat Anda gunakan untuk menginvokasi respons atau tindakan remediasi untuk kelas temuan tertentu.

Dengan Security Hub, Anda dapat menilai status keamanan dan kepatuhan sumber daya Amazon S3, dan Anda dapat melakukannya sebagai bagian dari analisis yang lebih luas tentang postur keamanan organisasi Anda di masing-masing Wilayah AWS dan di beberapa Wilayah. Ini termasuk menganalisis tren keamanan dan mengidentifikasi masalah keamanan prioritas tertinggi. Anda juga dapat mengumpulkan temuan dari beberapa Wilayah AWS, dan memantau serta memproses data temuan agregat dari satu Wilayah.

Untuk informasi selengkapnya, lihat Kontrol Amazon Simple Storage Service di dalam Panduan Pengguna AWS Security Hub .

Perlindungan data dalam Amazon S3

Amazon S3 menyediakan infrastruktur penyimpanan yang sangat tahan lama yang dirancang untuk penyimpanan data penting dan primer. S3 Standard, S3 Intelligent-Tiering, S3 Standard-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval, dan S3 Glacier Deep Archive menyimpan objek secara redundan di beberapa perangkat di minimal tiga Zona Ketersediaan dalam satu Wilayah AWS. Zona Ketersediaan adalah satu atau lebih pusat data diskret dengan daya redundan, jaringan, dan konektivitas dalam Wilayah AWS. Zona Ketersediaan secara fisik dipisahkan oleh jarak yang berarti, beberapa kilometer, dari Zona Ketersediaan lainnya, meskipun semuanya berada dalam jarak 100 km (60 mil) satu sama lain. Kelas penyimpanan S3 One Zone-IA menyimpan data secara redundan di beberapa perangkat dalam satu Zona Ketersediaan. Layanan ini dirancang untuk menangani kegagalan perangkat secara bersamaan dengan mendeteksi dan memperbaiki redundansi yang hilang secara cepat, dan layanan ini juga secara rutin memverifikasi integritas data Anda menggunakan checksum.

Penyimpanan Amazon S3 standard menawarkan fitur berikut:

- Didukung dengan Perjanjian Tingkat Layanan Amazon S3.
- Dirancang untuk memberikan daya tahan 99,999999999% dan ketersediaan objek 99,99% pada tahun tertentu.
- S3 Standard, S3 Intelligent-Tiering, S3 Standard-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval, dan S3 Glacier Deep Archive semuanya dirancang untuk mempertahankan data jika terjadi peristiwa kehilangan Zona Ketersediaan Amazon S3 secara keseluruhan.

Amazon S3 selanjutnya melindungi data Anda menggunakan Penentuan Versi. Anda dapat menggunakan Penentuan Versi untuk melestarikan, mengambil, dan memulihkan setiap versi dari setiap objek yang disimpan di dalam bucket Amazon S3 Anda. Dengan Penentuan Versi, Anda dapat dengan mudah memulihkan dari tindakan pengguna yang tidak diinginkan, serta kegagalan aplikasi. Secara default, permintaan mengambil versi terbaru yang ditulis. Anda dapat mengambil versi lama objek dengan menentukan versi objek dalam permintaan.

Selain S3 Penentuan Versi, Anda juga dapat menggunakan Kunci Objek Amazon S3 dan Replikasi S3 untuk melindungi data Anda. Untuk informasi selengkapnya, lihat Perlindungan data di Amazon S3.

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensyal dan mengatur akun pengguna individu dengan AWS Identity and Access Management, sehingga setiap pengguna hanya diberikan izin yang diperlukan untuk memenuhi tugas pekerjaan mereka.

Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-2 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Untuk informasi selengkapnya tentang titik akhir FIPS yang tersedia, lihat Federal Information Processing Standard (FIPS) 140-2.

Praktik terbaik keamanan berikut juga membahas perlindungan data di Amazon S3:

- Implement server-side encryption
- Enforce encryption of data in transit
- Consider using Macie with Amazon S3
- Identify and audit all your Amazon S3 buckets
- Monitor Amazon Web Services security advisories

Melindungi data dengan enkripsi

> **Important**
>
> Amazon S3 sudah menerapkan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3) sebagai tingkat dasar enkripsi bagi setiap bucket di Amazon S3. Mulai 5 Januari 2023, semua unggahan objek baru ke Amazon S3 secara otomatis akan dienkripsi tanpa biaya tambahan dan tidak akan berdampak pada kinerja. Status enkripsi otomatis untuk konfigurasi enkripsi default bucket S3 dan untuk unggahan objek baru tersedia di AWS CloudTrail log, S3 Inventory, S3 Storage Lens, konsol Amazon S3, dan sebagai header respons API Amazon S3 tambahan di dan. AWS Command Line Interface AWS SDKs Untuk informasi selengkapnya, lihat FAQ enkripsi default.

Perlindungan data mengacu pada perlindungan data bergerak (saat melakukan perjalanan ke dan dari Amazon S3) dan saat diam (saat data disimpan di dalam disk di pusat data Amazon S3). Anda dapat melindungi data dalam perjalanan dengan menggunakan Secure Socket Layer/Transport Layer Security (SSL/TLS) atau enkripsi sisi klien. Untuk melindungi data diam di Amazon S3, Anda memiliki opsi berikut:

- Enkripsi sisi server — Amazon S3 mengenkripsi objek Anda sebelum menyimpannya di disk di pusat AWS data dan kemudian mendekripsi objek saat Anda mengunduhnya.

  Semua bucket Amazon S3 memiliki enkripsi yang dikonfigurasi secara default, dan semua objek baru yang diunggah ke bucket S3 secara otomatis dienkripsi saat sedang tidak aktif. Enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3) adalah konfigurasi enkripsi default untuk setiap bucket di Amazon S3. Untuk menggunakan jenis enkripsi yang berbeda, Anda dapat menentukan jenis enkripsi sisi server yang akan digunakan dalam PUT permintaan S3, atau memperbarui konfigurasi enkripsi default di bucket tujuan.

  Jika Anda ingin menentukan jenis enkripsi yang berbeda dalam PUT permintaan Anda, Anda dapat menggunakan enkripsi sisi server dengan AWS Key Management Service () kunci (SSE-KMS AWS KMS), enkripsi sisi server dua lapis dengan kunci (DSSE-KMS), atau enkripsi sisi server dengan AWS KMS kunci yang disediakan pelanggan (SSE-C). Jika Anda ingin mengatur konfigurasi enkripsi default yang berbeda di dalam bucket tujuan, Anda dapat menggunakan SSE-KMS atau DSSE-KMS.

  Untuk informasi selengkapnya tentang mengubah konfigurasi enkripsi default untuk bucket tujuan umum Anda, lihatMengonfigurasi enkripsi default.

  Saat Anda mengubah konfigurasi enkripsi default bucket ke SSE-KMS, jenis enkripsi objek Amazon S3 yang ada di bucket tidak akan diubah. Untuk mengubah jenis enkripsi objek yang sudah ada sebelumnya setelah memperbarui konfigurasi enkripsi default ke SSE-KMS, Anda dapat menggunakan Operasi Batch Amazon S3. Anda menyediakan Operasi Batch S3 dengan daftar objek, dan Operasi Batch memanggil operasi API masing-masing. Anda dapat menggunakan Menyalin objek tindakan untuk menyalin objek yang ada, yang menuliskannya kembali ke bucket yang sama dengan objek terenkripsi SSE-KMS. Satu tugas Operasi Batch dapat melakukan operasi tertentu pada miliaran objek yang berisi data sebesar eksabita. Untuk informasi selengkapnya, lihat Melakukan operasi objek secara massal dengan Operasi Batch dan posting Blog AWS Penyimpanan Cara mengenkripsi objek yang ada secara surut di Amazon S3 menggunakan S3 Inventory, Amazon Athena, dan Operasi Batch S3.

  Untuk informasi selengkapnya tentang setiap opsi untuk enkripsi sisi server, lihat. Melindungi data dengan enkripsi di sisi klien

  Untuk mengonfigurasi enkripsi di sisi server, lihat:

  - Menentukan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3)
  - Menentukan enkripsi di sisi server dengan AWS KMS (SSE-KMS)

  - the section called “Menentukan SSE-KMS”
  - Menentukan enkripsi di sisi server dengan kunci yang disediakan pelanggan (SSE-C)

- Enkripsi di sisi klien–Anda mengenkripsi data sisi klien dan mengunggah data yang dienkripsi ke Amazon S3. Dalam hal ini, Anda mengelola proses enkripsi, kunci enkripsi, dan alat terkait.

  Untuk mengonfigurasi enkripsi di sisi klien, lihat Melindungi data menggunakan enkripsi di sisi klien.

Untuk melihat persentase byte penyimpanan yang dienkripsi, Anda dapat menggunakan metrik Lensa Penyimpanan Amazon S3. Lensa Penyimpanan S3 adalah fitur analisis penyimpanan cloud yang dapat Anda gunakan untuk mendapatkan visibilitas seluruh organisasi ke dalam penggunaan dan aktivitas penyimpanan objek. Untuk informasi selengkapnya, lihat Menilai aktivitas penyimpanan dan penggunaan Anda dengan Lensa Penyimpanan S3. Untuk daftar lengkap metrik, lihat Glosarium metrik Lensa Penyimpanan S3.

Untuk informasi selengkapnya tentang enkripsi di sisi server dan enkripsi di sisi klien, tinjau topik berikut.

Topik

- Melindungi data dengan enkripsi di sisi klien
- Melindungi data menggunakan enkripsi di sisi klien

Melindungi data dengan enkripsi di sisi klien

> **Important**
>
> Amazon S3 sudah menerapkan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3) sebagai tingkat dasar enkripsi bagi setiap bucket di Amazon S3. Mulai 5 Januari 2023, semua unggahan objek baru ke Amazon S3 secara otomatis akan dienkripsi tanpa biaya tambahan dan tidak akan berdampak pada kinerja. Status enkripsi otomatis untuk konfigurasi enkripsi default bucket S3 dan untuk unggahan objek baru tersedia di AWS CloudTrail log, S3 Inventory, S3 Storage Lens, konsol Amazon S3, dan sebagai header respons API Amazon S3 tambahan di dan. AWS Command Line Interface AWS SDKs Untuk informasi selengkapnya, lihat FAQ enkripsi default.

Enkripsi di sisi server adalah enkripsi data di tempat tujuan oleh aplikasi atau layanan yang menerimanya. Amazon S3 mengenkripsi data Anda pada tingkat objek saat menulisnya ke disk di pusat AWS data dan mendekripsi untuk Anda saat Anda mengaksesnya. Selama Anda mengautentikasi permintaan dan telah memiliki izin akses, tidak ada perbedaan dalam cara mengakses objek terenkripsi atau tidak terenkripsi. Misalnya, jika Anda berbagi objek menggunakan URL yang ditandatangani sebelumnya, URL tersebut bekerja dengan cara yang sama untuk objek terenkripsi maupun yang tidak terenkripsi. Selain itu, saat Anda mencantumkan objek di bucket, operasi daftar API menampilkan daftar semua objek, terlepas dari apakah objek tersebut dienkripsi atau tidak.

Semua bucket Amazon S3 memiliki enkripsi yang dikonfigurasi secara default, dan semua objek baru yang diunggah ke bucket S3 secara otomatis dienkripsi saat sedang tidak aktif. Enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3) adalah konfigurasi enkripsi default untuk setiap bucket di Amazon S3. Untuk menggunakan jenis enkripsi yang berbeda, Anda dapat menentukan jenis enkripsi sisi server yang akan digunakan dalam PUT permintaan S3, atau memperbarui konfigurasi enkripsi default di bucket tujuan.

Jika Anda ingin menentukan jenis enkripsi yang berbeda dalam PUT permintaan Anda, Anda dapat menggunakan enkripsi sisi server dengan AWS Key Management Service () kunci (SSE-KMS AWS KMS), enkripsi sisi server dua lapis dengan kunci (DSSE-KMS), atau enkripsi sisi server dengan AWS KMS kunci yang disediakan pelanggan (SSE-C). Jika Anda ingin mengatur konfigurasi enkripsi default yang berbeda di dalam bucket tujuan, Anda dapat menggunakan SSE-KMS atau DSSE-KMS.

Untuk informasi selengkapnya tentang mengubah konfigurasi enkripsi default untuk bucket tujuan umum Anda, lihatMengonfigurasi enkripsi default.

Saat Anda mengubah konfigurasi enkripsi default bucket ke SSE-KMS, jenis enkripsi objek Amazon S3 yang ada di bucket tidak akan diubah. Untuk mengubah jenis enkripsi objek yang sudah ada sebelumnya setelah memperbarui konfigurasi enkripsi default ke SSE-KMS, Anda dapat menggunakan Operasi Batch Amazon S3. Anda menyediakan Operasi Batch S3 dengan daftar objek, dan Operasi Batch memanggil operasi API masing-masing. Anda dapat menggunakan Menyalin objek tindakan untuk menyalin objek yang ada, yang menuliskannya kembali ke bucket yang sama dengan objek terenkripsi SSE-KMS. Satu tugas Operasi Batch dapat melakukan operasi tertentu pada miliaran objek yang berisi data sebesar eksabita. Untuk informasi selengkapnya, lihat Melakukan operasi objek secara massal dengan Operasi Batch dan posting Blog AWS Penyimpanan Cara mengenkripsi objek yang ada secara surut di Amazon S3 menggunakan S3 Inventory, Amazon Athena, dan Operasi Batch S3.

> **Note**
>
> Anda tidak dapat menerapkan berbagai jenis enkripsi di sisi server ke objek yang sama secara bersamaan.

Jika Anda perlu mengenkripsi objek yang ada, gunakan Operasi Batch S3 dan Inventaris S3. Untuk informasi selengkapnya, lihat Mengenkripsi objek dengan Operasi Batch Amazon S3 dan Melakukan operasi objek secara massal dengan Operasi Batch.

Saat menyimpan data di Amazon S3, Anda memiliki empat opsi yang saling eksklusif untuk enkripsi sisi server, tergantung pada bagaimana Anda memilih untuk mengelola kunci enkripsi dan jumlah lapisan enkripsi yang ingin Anda terapkan.

Enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3)

Semua bucket Amazon S3 memiliki enkripsi yang dikonfigurasi secara default. Opsi default untuk enkripsi di sisi server adalah dengan kunci terkelola Amazon S3 (SSE-S3). Setiap objek dienkripsi dengan kunci unik. Sebagai pelindung tambahan, SSE-S3 mengenkripsi kunci itu sendiri dengan kunci root yang secara rutin diputar. SSE-S3 menggunakan salah satu penyandian blok terkuat yang ada, Advanced Encryption Standard 256-bit (AES-256), untuk mengenkripsi data Anda. Untuk informasi selengkapnya, lihat Menggunakan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3).

Enkripsi sisi server dengan kunci AWS Key Management Service (AWS KMS) (SSE-KMS)

Enkripsi sisi server dengan AWS KMS keys (SSE-KMS) disediakan melalui integrasi layanan dengan Amazon S3. AWS KMS Dengan AWS KMS, Anda memiliki kontrol lebih besar atas kunci Anda. Misalnya, Anda dapat melihat kunci terpisah, mengedit kebijakan kontrol, dan mengikuti kunci di AWS CloudTrail. Selain itu, Anda dapat membuat dan mengelola CMK atau menggunakan Kunci yang dikelola AWS yang unik bagi Anda, layanan Anda, dan Wilayah Anda. Untuk informasi selengkapnya, lihat Menggunakan enkripsi sisi server dengan AWS KMS kunci (SSE-KMS).

Enkripsi sisi server dua lapis dengan kunci () (DSSE-KMS AWS Key Management Service)AWS KMS

Enkripsi sisi server dual-layer dengan AWS KMS keys (DSSE-KMS) mirip dengan SSE-KMS, tetapi DSSE-KMS menerapkan dua lapisan individual enkripsi tingkat objek, bukan satu lapisan. Karena kedua lapisan enkripsi diterapkan ke objek di sisi server, Anda dapat menggunakan berbagai alat

Layanan AWS dan untuk menganalisis data di S3 sambil menggunakan metode enkripsi yang dapat memenuhi persyaratan kepatuhan Anda. Untuk informasi selengkapnya, lihat Menggunakan enkripsi sisi server dual-layer dengan kunci (DSSE-KMS) AWS KMS.

Enkripsi di sisi server dengan kunci yang disediakan pelanggan (SSE-C)

Dengan enkripsi di sisi server dengan kunci yang disediakan pelanggan (SSE-C), Anda mengelola kunci enkripsi, dan Amazon S3 mengelola enkripsi saat menulis ke disk dan dekripsi saat Anda mengakses objek Anda. Untuk informasi selengkapnya, lihat Menggunakan enkripsi di sisi server dengan kunci yang disediakan pelanggan (SSE-C).

> **Note**
>
> Saat menggunakan titik akses untuk sistem FSx file Amazon menggunakan titik akses S3, Anda memiliki satu opsi untuk enkripsi sisi server.
>
> Enkripsi sisi server dengan Amazon FSx (SSE-FSX)
>
> Semua sistem FSx file Amazon memiliki enkripsi yang dikonfigurasi secara default dan dienkripsi saat istirahat dengan kunci yang dikelola menggunakan. AWS Key Management Service Data secara otomatis dienkripsi dan didekripsi oleh pada sistem file karena data sedang ditulis dan dibaca dari sistem file. Proses ini ditangani secara transparan oleh Amazon. FSx

Mengatur perilaku enkripsi sisi server default untuk bucket Amazon S3

> **Important**
>
> Amazon S3 sudah menerapkan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3) sebagai tingkat dasar enkripsi bagi setiap bucket di Amazon S3. Mulai 5 Januari 2023, semua unggahan objek baru ke Amazon S3 secara otomatis akan dienkripsi tanpa biaya tambahan dan tidak akan berdampak pada kinerja. Status enkripsi otomatis untuk konfigurasi enkripsi default bucket S3 dan untuk unggahan objek baru tersedia di AWS CloudTrail log, S3 Inventory, S3 Storage Lens, konsol Amazon S3, dan sebagai header respons API Amazon S3 tambahan di dan. AWS Command Line Interface AWS SDKs Untuk informasi selengkapnya, lihat FAQ enkripsi default.

Semua bucket Amazon S3 memiliki enkripsi yang dikonfigurasi secara default, dan objek secara otomatis dienkripsi dengan menggunakan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3). Pengaturan enkripsi ini berlaku untuk semua objek di dalam bucket Amazon S3 Anda.

Jika Anda memerlukan kontrol lebih besar atas kunci Anda, seperti mengelola rotasi kunci dan hibah kebijakan akses, Anda dapat memilih untuk menggunakan enkripsi sisi server dengan () kunci AWS Key Management Service (SSE-KMS AWS KMS), atau enkripsi sisi server dua lapis dengan kunci (DSSE-KMS). AWS KMS Untuk informasi selengkapnya tentang mengedit kunci KMS, lihat Mengedit kunci dalam Panduan Pengembang AWS Key Management Service .

> **Note**
>
> Kami telah mengubah bucket untuk mengenkripsi unggahan objek baru secara otomatis. Jika sebelumnya Anda membuat bucket tanpa enkripsi default, Amazon S3 akan mengaktifkan enkripsi secara default untuk bucket menggunakan SSE-S3. Tidak akan ada perubahan pada konfigurasi enkripsi default untuk bucket yang sudah ada yang sudah memiliki konfigurasi SSE-S3 atau SSE-KMS. Jika Anda ingin mengenkripsi objek Anda dengan SSE-KMS, Anda harus mengubah jenis enkripsi di pengaturan bucket Anda. Untuk informasi selengkapnya, lihat Menggunakan enkripsi sisi server dengan AWS KMS kunci (SSE-KMS).

Saat mengonfigurasi bucket untuk menggunakan enkripsi default dengan SSE-KMS, Anda juga dapat mengaktifkan Kunci Bucket S3 untuk mengurangi lalu lintas permintaan dari Amazon S3 ke AWS KMS dan mengurangi biaya enkripsi. Untuk informasi selengkapnya, lihat Mengurangi biaya SSE-KMS dengan Kunci Bucket Amazon S3.

Untuk mengidentifikasi bucket yang mengaktifkan SSE-KMS untuk enkripsi default, Anda dapat menggunakan metrik Lensa Penyimpanan Amazon S3. Lensa Penyimpanan S3 adalah fitur analisis penyimpanan cloud yang dapat Anda gunakan untuk mendapatkan visibilitas seluruh organisasi ke dalam penggunaan dan aktivitas penyimpanan objek. Untuk informasi selengkapnya, lihat Menggunakan Lensa Penyimpanan S3 untuk melindungi data Anda.

Saat Anda menggunakan enkripsi di sisi server, Amazon S3 mengenkripsi objek sebelum menyimpannya ke disk, dan mendekripsinya saat Anda mengunduh objek. Untuk informasi lebih lanjut tentang cara untuk melindungi data menggunakan enkripsi di sisi server dan manajemen kunci enkripsi, lihat Melindungi data dengan enkripsi di sisi klien.

Untuk informasi lebih lanjut tentang izin yang diperlukan untuk enkripsi default, lihat PutBucketEncryption dalam Referensi API Amazon Simple Storage Service.

Anda dapat mengonfigurasi perilaku enkripsi default Amazon S3 untuk bucket S3 dengan menggunakan konsol Amazon S3, API REST Amazon S3, dan Antarmuka Baris Perintah (). AWS SDKs AWS AWS CLI

Mengenkripsi objek yang ada

Untuk mengenkripsi objek Amazon S3 yang tidak terenkripsi, Anda dapat menggunakan Operasi Batch Amazon S3. Anda harus menyediakan Operasi Batch S3 dengan daftar objek untuk dioperasikan, dan Operasi Batch akan memanggil masing-masing API untuk melakukan operasi tertentu. Anda juga dapat menggunakan Operasi Salin Operasi Batch untuk menyalin objek tidak terenkripsi yang ada, dan menuliskannya kembali ke bucket yang sama sebagai objek terenkripsi. Satu tugas Operasi Batch dapat melakukan operasi tertentu pada miliaran objek yang berisi data sebesar eksabita. Untuk informasi selengkapnya, lihat Melakukan operasi objek secara massal dengan Operasi Batch dan postingan AWS Blog Penyimpanan Mengenkripsi objek dengan Operasi Batch Amazon S3.

Anda juga dapat mengenkripsi objek yang ada dengan menggunakan operasi `CopyObject` API atau `copy-object` AWS CLI perintah. Untuk informasi selengkapnya, lihat postingan AWS Blog Penyimpanan Mengenkripsi objek Amazon S3 yang sudah ada dengan AWS CLI.

> **Note**
>
> Bucket Amazon S3 dengan enkripsi bucket default yang diatur ke SSE-KMS tidak dapat digunakan sebagai bucket tujuan the section called “Pencatatan akses server”. Hanya enkripsi default SSE-S3 yang didukung untuk bucket tujuan log akses server.

Menggunakan enkripsi SSE-KMS untuk operasi lintas akun

Saat menggunakan enkripsi untuk operasi lintas akun, perhatikan hal berikut:

- Jika Nama Sumber Daya AWS KMS key Amazon (ARN) atau alias tidak disediakan pada waktu permintaan atau melalui konfigurasi enkripsi default bucket, Kunci yang dikelola AWS (`aws/s3`) akan digunakan.
- Jika Anda mengunggah atau mengakses objek S3 dengan menggunakan prinsipal AWS Identity and Access Management (IAM) yang sama Akun AWS dengan kunci KMS Anda, Anda dapat menggunakan (). Kunci yang dikelola AWS `aws/s3`
- Jika Anda ingin memberikan akses lintas akun ke objek S3 Anda, gunakan CMK. Anda dapat mengonfigurasi kebijakan CMK untuk mengizinkan akses dari akun lain.

- Jika Anda menentukan kunci KMS yang dikelola pelanggan, kami sarankan untuk menggunakan ARN kunci KMS yang memenuhi syarat sepenuhnya. Jika Anda menggunakan alias kunci KMS sebagai gantinya, AWS KMS selesaikan kunci dalam akun pemohon. Perilaku ini dapat menghasilkan data yang dienkripsi dengan kunci KMS milik pemohon, dan bukan pemilik bucket.
- Anda harus menentukan kunci yang kepada Anda (pemohon) telah diberikan izin `Encrypt`. Untuk informasi selengkapnya, lihat Mengizinkan pengguna di akun lain untuk menggunakan kunci KMS untuk operasi kriptografi di Panduan Pengembang AWS Key Management Service .

Untuk informasi selengkapnya tentang kapan menggunakan kunci terkelola pelanggan dan kunci KMS AWS terkelola, lihat Haruskah saya menggunakan Kunci yang dikelola AWS atau kunci yang dikelola pelanggan untuk mengenkripsi objek saya di Amazon S3?

Menggunakan enkripsi default menggunakan replikasi

Saat Anda mengaktifkan enkripsi default untuk bucket tujuan replikasi, perilaku enkripsi berikut berlaku:

- Jika objek dalam bucket sumber tidak dienkripsi, objek replika dalam bucket tujuan akan dienkripsi menggunakan pengaturan enkripsi default dari bucket tujuan. Akibatnya, tag entitas (ETags) dari objek sumber berbeda ETags dari objek replika. Jika Anda memiliki aplikasi yang menggunakan ETags, Anda harus memperbarui aplikasi tersebut untuk memperhitungkan perbedaan ini.
- Jika objek dalam bucket sumber dienkripsi menggunakan enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3), enkripsi sisi server dengan kunci () (SSE-KMS), atau enkripsi sisi server dua lapis dengan AWS Key Management Service kunci (DSSE-KMS), objek replika di bucket tujuan menggunakan jenis enkripsi yang sama dengan AWS KMS objek sumber.AWS KMS Pengaturan enkripsi default bucket tujuan tidak digunakan.

Untuk informasi lebih lanjut tentang penggunaan enkripsi default dengan SSE-KMS, lihat Mereplikasi objek terenkripsi.

Menggunakan Kunci Bucket Amazon S3 dengan enkripsi default

Saat mengonfigurasi bucket untuk menggunakan enkripsi default dengan SSE-KMS, Anda juga dapat mengaktifkan Kunci Bucket S3. S3 Bucket Keys mengurangi jumlah transaksi dari Amazon S3 AWS KMS untuk mengurangi biaya SSE-KMS.

Saat mengonfigurasi bucket untuk menggunakan Kunci Bucket S3 untuk SSE-KMS pada objek baru, AWS KMS buat kunci tingkat ember yang digunakan untuk membuat kunci data unik untuk objek di

bucket. Kunci Bucket S3 ini digunakan untuk jangka waktu terbatas dalam Amazon S3, mengurangi kebutuhan Amazon S3 untuk membuat permintaan AWS KMS untuk menyelesaikan operasi enkripsi.

Untuk informasi lebih lanjut tentang menggunakan Kunci Bucket S3, lihat Menggunakan Kunci Bucket Amazon S3.

Mengonfigurasi enkripsi default

> **Important**
>
> Amazon S3 sudah menerapkan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3) sebagai tingkat dasar enkripsi bagi setiap bucket di Amazon S3. Mulai 5 Januari 2023, semua unggahan objek baru ke Amazon S3 secara otomatis akan dienkripsi tanpa biaya tambahan dan tidak akan berdampak pada kinerja. Status enkripsi otomatis untuk konfigurasi enkripsi default bucket S3 dan untuk unggahan objek baru tersedia di AWS CloudTrail log, S3 Inventory, S3 Storage Lens, konsol Amazon S3, dan sebagai header respons API Amazon S3 tambahan di dan. AWS Command Line Interface AWS SDKs Untuk informasi selengkapnya, lihat FAQ enkripsi default.

Bucket Amazon S3 mengaktifkan enkripsi bucket secara default, dan objek baru secara otomatis dienkripsi dengan menggunakan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3). Enkripsi ini berlaku untuk semua objek baru di bucket Amazon S3 Anda, dan Anda tidak akan dikenakan biaya.

Jika Anda memerlukan kontrol lebih besar atas kunci enkripsi Anda, seperti mengelola rotasi kunci dan hibah kebijakan akses, Anda dapat memilih untuk menggunakan enkripsi sisi server dengan AWS Key Management Service () kunci (SSE-KMS AWS KMS), atau enkripsi sisi server dua lapis dengan kunci (DSSE-KMS). AWS KMS Untuk informasi lebih lanjut tentang SSE-KMS, lihat Menentukan enkripsi di sisi server dengan AWS KMS (SSE-KMS). Untuk informasi lebih lanjut tentang DSSE-KMS, lihat the section called “Enkripsi di sisi server dua lapis (DSSE-KMS)”.

Jika Anda ingin menggunakan kunci KMS yang dimiliki oleh akun lain, Anda harus memiliki izin untuk menggunakan kunci tersebut. Untuk informasi selengkapnya tentang izin lintas akun untuk kunci KMS, lihat Membuat kunci KMS yang dapat digunakan oleh akun lain di Panduan Pengembang AWS Key Management Service .

Saat Anda menyetel enkripsi bucket default ke SSE-KMS, Anda juga dapat mengonfigurasi Kunci Bucket S3 untuk mengurangi biaya permintaan. AWS KMS Untuk informasi selengkapnya, lihat Mengurangi biaya SSE-KMS dengan Kunci Bucket Amazon S3.

> **Note**
>
> Jika Anda menggunakan PutBucketEncryptionuntuk menyetel enkripsi bucket default ke SSE-KMS, Anda harus memverifikasi bahwa ID kunci KMS Anda sudah benar. Amazon S3 tidak memvalidasi ID kunci KMS yang disediakan dalam permintaan. PutBucketEncryption

Tidak ada biaya tambahan untuk menggunakan enkripsi default untuk bucket S3. Permintaan untuk mengonfigurasi perilaku enkripsi default dikenakan biaya permintaan standar Amazon S3. Untuk informasi tentang harga, lihat Harga Amazon S3. Untuk SSE-KMS dan DSSE-KMS, AWS KMS biaya berlaku dan tercantum pada harga.AWS KMS

Enkripsi di sisi server dengan kunci yang disediakan pelanggan (SSE-C) tidak didukung untuk enkripsi default.

Anda dapat mengonfigurasi enkripsi default Amazon S3 untuk bucket S3 dengan menggunakan konsol Amazon S3, API REST AWS SDKs Amazon S3, dan (). AWS Command Line Interface AWS CLI

Perubahan yang perlu diingat sebelum mengaktifkan enkripsi default

Setelah Anda mengaktifkan enkripsi default untuk bucket, perilaku enkripsi berikut ini akan berlaku:

- Tidak ada perubahan pada enkripsi objek yang ada dalam bucket sebelum enkripsi default-nya diaktifkan.
- Saat Anda mengunggah objek setelah mengaktifkan enkripsi default:
  - Jika header permintaan PUT tidak mencakup informasi enkripsi, Amazon S3 akan menggunakan pengaturan enkripsi default bucket untuk mengenkripsi objek.
  - Jika header permintaan PUT mencakup informasi enkripsi, Amazon S3 menggunakan informasi enkripsi dari permintaan PUT untuk mengenkripsi objek sebelum menyimpannya di Amazon S3.
- Jika Anda menggunakan opsi SSE-KMS untuk konfigurasi enkripsi default, Anda tunduk pada kuota permintaan per detik (RPS) AWS KMS. Untuk informasi selengkapnya tentang kuota AWS KMS dan cara meminta kenaikan kuota, lihat Kuota di Panduan Pengembang AWS Key Management Service .

> **Note**
>
> Objek yang diunggah sebelum enkripsi default diaktifkan tidak akan dienkripsi. Untuk informasi tentang mengenkripsi objek yang ada, lihat the section called “Mengatur enkripsi bucket default”.

Menggunakan konsol S3

Untuk mengonfigurasi enkripsi default di bucket Amazon S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Dari daftar Bucket, pilih nama bucket yang Anda inginkan.
4. Pilih tab Properti.
5. Di bagian bawah Enkripsi default, pilih Edit.
6. Untuk mengonfigurasi enkripsi default, di bawah Jenis enkripsi, pilih salah satu dari berikut ini:
   - Enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3)
   - Enkripsi sisi server dengan AWS Key Management Service kunci (SSE-KMS)
   - Enkripsi sisi server dua lapis dengan kunci (DSSE-KMS) AWS Key Management Service

   > **Important**
   >
   > Jika Anda menggunakan opsi SSE-KMS atau DSSE-KMS untuk konfigurasi enkripsi default, Anda tunduk pada kuota permintaan per detik (RPS) AWS KMS. Untuk informasi selengkapnya tentang AWS KMS kuota dan cara meminta kenaikan kuota, lihat Kuota di Panduan Pengembang AWS Key Management Service .

   Bucket dan objek baru dienkripsi secara default dengan SSE-S3, kecuali Anda menentukan jenis enkripsi default lain untuk bucket Anda. Untuk informasi selengkapnya tentang enkripsi default, lihat Mengatur perilaku enkripsi sisi server default untuk bucket Amazon S3.

Untuk informasi selengkapnya tentang penggunaan enkripsi di sisi server Amazon S3 guna mengenkripsi data Anda, lihat Menggunakan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3).

7. Jika Anda memilih enkripsi sisi server dengan AWS Key Management Service kunci (SSE-KMS) atau enkripsi sisi server Dual-layer dengan kunci (DSSE-KMS), lakukan hal berikut: AWS Key Management Service

   a. Di bawah AWS KMS kunci, tentukan kunci KMS Anda dengan salah satu cara berikut ini:

   - Untuk memilih dari daftar kunci KMS yang tersedia, pilih Pilih dari Anda AWS KMS keys, dan pilih kunci KMS Anda dari daftar kunci yang tersedia.

     Kunci Kunci yang dikelola AWS (`aws/s3`) dan kunci terkelola pelanggan Anda muncul dalam daftar ini. Untuk informasi selengkapnya tentang kunci yang dikelola pelanggan, lihat Kunci dan AWS kunci pelanggan di Panduan AWS Key Management Service Pengembang.

   - Untuk memasukkan ARN kunci KMS, pilih Masukkan AWS KMS key ARN, dan masukkan ARN kunci KMS Anda di bidang yang muncul.
   - Untuk membuat kunci terkelola pelanggan baru di AWS KMS konsol, pilih Buat kunci KMS.

     Untuk informasi selengkapnya tentang membuat AWS KMS key, lihat Membuat kunci di Panduan AWS Key Management Service Pengembang.

> **Important**
>
> Anda hanya dapat menggunakan tombol KMS yang diaktifkan Wilayah AWS sama dengan bucket. Saat memilih Pilih dari kunci KMS Anda, konsol S3 hanya mencantumkan 100 kunci KMS per Wilayah. Jika Anda memiliki lebih dari 100 tombol KMS di Wilayah yang sama, Anda hanya dapat melihat 100 kunci KMS pertama di konsol S3. Untuk menggunakan kunci KMS yang tidak terdaftar di konsol, pilih Masukkan ARN AWS KMS key , dan masukkan ARN kunci KMS. Saat Anda menggunakan enkripsi sisi server AWS KMS key untuk Amazon S3, Anda harus memilih kunci KMS enkripsi simetris. Amazon S3 hanya mendukung kunci KMS enkripsi simetris. Untuk informasi selengkapnya terkait kunci ini, lihat

Membuat kunci enkripsi simetris KMS dalam Panduan Pengembang AWS Key Management Service .

Untuk informasi selengkapnya tentang penggunaan SSE-KMS dengan Amazon S3, lihat Menggunakan enkripsi sisi server dengan AWS KMS kunci (SSE-KMS). Untuk informasi selengkapnya tentang DSSE-KMS, lihat the section called “Enkripsi di sisi server dua lapis (DSSE-KMS)”.

b. Saat Anda mengonfigurasikan bucket untuk menggunakan enkripsi default dengan SSE-KMS, Anda juga dapat mengaktifkan Kunci Bucket S3. S3 Bucket Keys menurunkan biaya enkripsi dengan mengurangi lalu lintas permintaan dari Amazon S3 ke. AWS KMS Untuk informasi selengkapnya, lihat Mengurangi biaya SSE-KMS dengan Kunci Bucket Amazon S3.

Untuk menggunakan Kunci Bucket S3, di bagian bawah Kunci Bucket, pilih Aktifkan.

> **Note**
>
> Kunci Bucket S3 tidak didukung untuk DSSE-KMS.

8. Pilih Simpan perubahan.

Menggunakan AWS CLI

Contoh ini menunjukkan cara mengonfigurasi enkripsi default dengan menggunakan SSE-S3 atau dengan menggunakan SSE-KMS dengan Kunci Bucket S3.

Untuk informasi selengkapnya tentang enkripsi default, lihat Mengatur perilaku enkripsi sisi server default untuk bucket Amazon S3. Untuk informasi selengkapnya tentang menggunakan enkripsi default AWS CLI untuk mengonfigurasi, lihat put-bucket-encryption.

Example – Enkripsi default dengan SSE-S3

Contoh ini mengonfigurasi enkripsi bucket default dengan kunci terkelola Amazon S3.

```
aws s3api put-bucket-encryption --bucket amzn-s3-demo-bucket --server-side-encryption-
configuration '{
    "Rules": [
        {
```

```
            "ApplyServerSideEncryptionByDefault": {
                "SSEAlgorithm": "AES256"
            }
        }
    ]
}'
```

Example – Enkripsi default dengan SSE-KMS menggunakan Kunci Bucket S3

Contoh ini mengonfigurasi enkripsi bucket default dengan SSE-KMS menggunakan Kunci Bucket S3.

```
aws s3api put-bucket-encryption --bucket amzn-s3-demo-bucket --server-side-encryption-
configuration '{
    "Rules": [
            {
                "ApplyServerSideEncryptionByDefault": {
                    "SSEAlgorithm": "aws:kms",
                    "KMSMasterKeyID": "KMS-Key-ARN"
                },
                "BucketKeyEnabled": true
            }
        ]
    }'
```

Penggunaan API REST

Gunakan operasi `PutBucketEncryption` API REST untuk mengaktifkan enkripsi default, dan untuk menyetel jenis enkripsi di sisi server yang akan menggunakan—SSE-S3, SSE-KMS, atau DSSE-KMS.

Untuk informasi selengkapnya, lihat PutBucketEncryption dalam Referensi API Amazon Simple Storage Service.

Memantau enkripsi default dengan AWS CloudTrail dan Amazon EventBridge

> **Important**
>
> Amazon S3 sudah menerapkan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3) sebagai tingkat dasar enkripsi bagi setiap bucket di Amazon S3. Mulai 5 Januari 2023, semua unggahan objek baru ke Amazon S3 secara otomatis akan dienkripsi tanpa biaya tambahan dan tidak akan berdampak pada kinerja. Status enkripsi otomatis untuk konfigurasi enkripsi default bucket S3 dan untuk unggahan objek baru tersedia di AWS

CloudTrail log, S3 Inventory, S3 Storage Lens, konsol Amazon S3, dan sebagai header respons API Amazon S3 tambahan di dan. AWS Command Line Interface AWS SDKs Untuk informasi selengkapnya, lihat FAQ enkripsi default.

Anda dapat melacak permintaan konfigurasi enkripsi default untuk bucket Amazon S3 menggunakan peristiwa AWS CloudTrail . Nama peristiwa API berikut digunakan dalam CloudTrail log:

- `PutBucketEncryption`
- `GetBucketEncryption`
- `DeleteBucketEncryption`

Anda juga dapat membuat EventBridge aturan untuk mencocokkan CloudTrail peristiwa untuk panggilan API ini. Untuk informasi selengkapnya tentang CloudTrail acara, lihatMengaktifkan pencatatan untuk objek dalam bucket menggunakan konsol tersebut. Untuk informasi selengkapnya tentang EventBridge acara, lihat Acara dari Layanan AWS.

Anda dapat menggunakan CloudTrail log untuk tindakan Amazon S3 tingkat objek untuk PUT melacak POST dan meminta ke Amazon S3. Anda dapat menggunakan tindakan ini untuk memverifikasi apakah enkripsi default digunakan untuk mengenkripsi objek saat permintaan PUT yang masuk tidak memiliki header enkripsi.

Saat Amazon S3 mengenkripsi objek menggunakan pengaturan enkripsi default, log mencakup salah satu bidang berikut sebagai pasangan nama-nilai: `"SSEApplied":"Default_SSE_S3"`, `"SSEApplied":"Default_SSE_KMS"`, atau `"SSEApplied":"Default_DSSE_KMS"`.

Saat Amazon S3 mengenkripsi objek menggunakan header enkripsi PUT, log mencakup salah satu bidang berikut sebagai pasangan nama-nilai: `"SSEApplied":"SSE_S3"`, `"SSEApplied":"SSE_KMS"`, `"SSEApplied":"DSSE_KMS"`, atau `"SSEApplied":"SSE_C"`.

Untuk pengunggahan multibagian, informasi ini disertakan dalam permintaan operasi `InitiateMultipartUpload` API Anda. Untuk informasi lebih lanjut tentang menggunakan CloudTrail dan CloudWatch, lihatPencatatan log dan pemantauan di Amazon S3.

FAQ enkripsi default

Amazon S3 sudah menerapkan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3) sebagai tingkat dasar enkripsi bagi setiap bucket di Amazon S3. Mulai 5 Januari 2023, semua

unggahan objek baru ke Amazon S3 secara otomatis akan dienkripsi tanpa biaya tambahan dan tidak akan berdampak pada kinerja. SSE-S3, yang menggunakan Standar Enkripsi Lanjutan 256-bit (AES-256), secara otomatis diterapkan ke semua bucket baru dan ke bucket S3 yang ada yang belum memiliki enkripsi default yang dikonfigurasi. Status enkripsi otomatis untuk konfigurasi enkripsi default bucket S3 dan untuk unggahan objek baru tersedia di AWS CloudTrail log, S3 Inventory, S3 Storage Lens, konsol Amazon S3, dan sebagai header respons Amazon S3 API tambahan di () dan file. AWS Command Line Interface AWS CLI AWS SDKs

Bagian berikut menjawab pertanyaan tentang pembaruan ini.

Apakah Amazon S3 mengubah pengaturan enkripsi default untuk bucket saya yang sudah ada yang sudah memiliki enkripsi default yang dikonfigurasi?

Tidak. Tidak ada perubahan pada konfigurasi enkripsi default untuk bucket yang sudah ada yang sudah memiliki enkripsi SSE-S3 atau sisi server dengan kunci AWS Key Management Service (AWS KMS) (SSE-KMS) yang dikonfigurasi. Untuk informasi lebih lanjut tentang cara mengatur perilaku enkripsi default untuk bucket, lihat Mengatur perilaku enkripsi sisi server default untuk bucket Amazon S3. Untuk informasi selengkapnya tentang pengaturan enkripsi SSE-S3 dan SSE-KMS, lihat Melindungi data dengan enkripsi di sisi klien.

Apakah enkripsi default diaktifkan pada bucket saya yang sudah ada yang enkripsi defaultnya tidak dikonfigurasi?

Ya. Amazon S3 sekarang mengonfigurasi enkripsi default pada semua bucket tidak terenkripsi yang sudah ada untuk menerapkan enkripsi di sisi server dengan kunci terkelola S3 (SSE-S3) sebagai tingkat dasar enkripsi untuk objek baru yang diunggah ke bucket ini. Objek yang sudah ada dalam bucket tidak terenkripsi tidak akan dienkripsi secara otomatis.

Bagaimana saya bisa melihat status enkripsi default dari unggahan objek baru?

Saat ini, Anda dapat melihat status enkripsi default dari unggahan objek baru di AWS CloudTrail log, Inventaris S3, dan Lensa Penyimpanan S3, konsol Amazon S3, dan sebagai header respons API Amazon S3 tambahan di () dan. AWS Command Line Interface AWS CLI AWS SDKs

- Untuk melihat CloudTrail acara Anda, lihat Melihat CloudTrail peristiwa di CloudTrail konsol di Panduan AWS CloudTrail Pengguna. CloudTrail log menyediakan pelacakan API untuk PUT dan POST permintaan ke Amazon S3. Saat enkripsi default digunakan untuk mengenkripsi objek di bucket Anda, CloudTrail log untuk PUT dan permintaan POST API akan menyertakan bidang berikut sebagai pasangan nama-nilai:. `"SSEApplied":"Default_SSE_S3"`

- Untuk melihat status enkripsi otomatis dari unggahan objek baru di Inventaris S3, konfigurasikan laporan Inventaris S3 untuk menyertakan bidang metadata Enkripsi, lalu lihat status enkripsi untuk setiap objek baru dalam laporan. Untuk informasi selengkapnya, lihat Mengatur Inventaris Amazon S3.
- Untuk melihat status enkripsi otomatis untuk unggahan objek baru di Lensa Penyimpanan S3, konfigurasikan dasbor Lensa Penyimpanan S3 dan lihat metrik Byte terenkripsi dan Jumlah objek terenkripsi dalam kategori Perlindungan data pada dasbor. Untuk informasi selengkapnya, lihat Menggunakan konsol S3 dan Melihat metrik Lensa Penyimpanan S3 di dasbor.
- Untuk melihat status enkripsi tingkat bucket otomatis di konsol Amazon S3, periksa Enkripsi default bucket Amazon S3 Anda di konsol Amazon S3. Untuk informasi selengkapnya, lihat Mengonfigurasi enkripsi default.
- Untuk melihat status enkripsi otomatis sebagai header respons Amazon S3 API tambahan di AWS Command Line Interface (AWS CLI) dan AWS SDKs, periksa header respons `x-amz-server-side-encryption` saat Anda menggunakan tindakan objek APIs, seperti PutObjectdan. GetObject

Apa yang harus saya lakukan untuk memanfaatkan perubahan ini?

Anda tidak perlu melakukan perubahan apa pun pada aplikasi Anda yang ada. Karena enkripsi default diaktifkan untuk semua bucket Anda, semua objek baru yang diunggah ke Amazon S3 secara otomatis dienkripsi.

Bisakah saya menonaktifkan enkripsi untuk objek baru yang ditulis ke bucket saya?

Tidak. SSE-S3 adalah enkripsi tingkat dasar baru yang diterapkan pada semua objek baru yang diunggah ke bucket Anda. Anda tidak dapat lagi menonaktifkan enkripsi untuk unggahan objek baru.

Apakah biaya saya akan terpengaruh?

Tidak. Enkripsi default dengan SSE-S3 tersedia tanpa biaya tambahan. Anda akan ditagih untuk penyimpanan, permintaan, dan fitur S3 lainnya, seperti biasa. Untuk informasi harga, lihat Harga Amazon S3.

Akankah Amazon S3 mengenkripsi objek saya yang tidak terenkripsi?

Tidak. Mulai 5 Januari 2023, Amazon S3 hanya secara otomatis mengenkripsi unggahan objek baru. Untuk mengenkripsi objek yang ada, Anda dapat menggunakan operasi batch S3 untuk membuat salinan terenkripsi objek Anda. Salinan terenkripsi ini akan mempertahankan data dan nama objek yang ada dan akan dienkripsi dengan menggunakan kunci enkripsi yang Anda tentukan. Untuk

detail selengkapnya, lihat Mengenkripsi objek dengan Operasi Batch Amazon S3 dalam AWS Blog Penyimpanan.

Saya tidak mengaktifkan enkripsi untuk bucket saya sebelum rilis ini. Apakah saya perlu mengubah cara saya mengakses objek?

Tidak. Enkripsi default dengan SSE-S3 secara otomatis mengenkripsi data Anda seperti yang tertulis ke Amazon S3 dan mendekripsinya untuk Anda saat Anda mengaksesnya. Tidak ada perubahan dalam cara Anda mengakses objek yang dienkripsi secara otomatis.

Apakah saya perlu mengubah cara saya mengakses objek terenkripsi di sisi klien?

Tidak. Semua objek terenkripsi di sisi klien yang dienkripsi sebelum diunggah ke Amazon S3 tiba sebagai objek teks sandi terenkripsi dalam Amazon S3. Objek ini sekarang akan memiliki lapisan enkripsi SSE-S3. Beban kerja Anda yang menggunakan objek yang terenkripsi di sisi klien tidak akan memerlukan perubahan apa pun pada layanan klien atau pengaturan otorisasi Anda.

> **Note**
>
> HashiCorp Pengguna Terraform yang tidak menggunakan versi AWS Penyedia yang diperbarui mungkin melihat penyimpangan yang tidak terduga setelah membuat bucket S3 baru tanpa konfigurasi enkripsi yang ditentukan pelanggan. Untuk menghindari penyimpangan ini, perbarui versi AWS Penyedia Terraform Anda ke salah satu versi berikut: 4.x rilis apa pun,, 3.76.1 atau. 2.70.4

Menggunakan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3)

> **Important**
>
> Amazon S3 sudah menerapkan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3) sebagai tingkat dasar enkripsi bagi setiap bucket di Amazon S3. Mulai 5 Januari 2023, semua unggahan objek baru ke Amazon S3 secara otomatis akan dienkripsi tanpa biaya tambahan dan tidak akan berdampak pada kinerja. Status enkripsi otomatis untuk konfigurasi enkripsi default bucket S3 dan untuk unggahan objek baru tersedia di AWS CloudTrail log, S3 Inventory, S3 Storage Lens, konsol Amazon S3, dan sebagai header respons API Amazon S3 tambahan di dan. AWS Command Line Interface AWS SDKs Untuk informasi selengkapnya, lihat FAQ enkripsi default.

Semua unggahan objek baru ke bucket Amazon S3 dienkripsi secara default dengan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3).

Enkripsi sisi-server melindungi data diam. Amazon S3 mengenkripsi setiap objek dengan kunci unik. Sebagai perlindungan tambahan, ia mengenkripsi kunci itu sendiri dengan kunci yang diputar secara berkala. Enkripsi sisi server Amazon S3 menggunakan Galois/Counter Mode Standar Enkripsi Lanjutan (AES-GCM) 256-bit untuk mengenkripsi semua objek yang diunggah.

Tidak ada biaya tambahan untuk menggunakan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3). Namun, permintaan untuk mengonfigurasi fitur enkripsi default dikenakan biaya permintaan Amazon S3 standar. Untuk informasi tentang harga, lihat Harga Amazon S3.

Jika Anda mengharuskan unggahan data dienkripsi hanya menggunakan kunci terkelola Amazon S3, Anda dapat menggunakan kebijakan bucket berikut. Misalnya, kebijakan bucket berikut menolak izin untuk mengunggah objek kecuali jika permintaan mencakup header `x-amz-server-side-encryption` untuk meminta enkripsi di sisi server:

```
{
  "Version": "2012-10-17",
  "Id": "PutObjectPolicy",
  "Statement": [
    {
      "Sid": "DenyObjectsThatAreNotSSES3",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
      "Condition": {
        "StringNotEquals": {
          "s3:x-amz-server-side-encryption": "AES256"
        }
      }
    }
   ]
}
```

> **Note**
>
> Enkripsi di sisi server hanya mengenkripsi data objek, bukan metadata objek.

Dukungan API untuk enkripsi di sisi server

Semua bucket Amazon S3 memiliki enkripsi yang dikonfigurasi secara default, dan semua objek baru yang diunggah ke bucket S3 secara otomatis dienkripsi saat sedang tidak aktif. Enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3) adalah konfigurasi enkripsi default untuk setiap bucket di Amazon S3. Untuk menggunakan jenis enkripsi yang berbeda, Anda dapat menentukan jenis enkripsi sisi server yang akan digunakan dalam PUT permintaan S3, atau memperbarui konfigurasi enkripsi default di bucket tujuan.

Jika Anda ingin menentukan jenis enkripsi yang berbeda dalam PUT permintaan Anda, Anda dapat menggunakan enkripsi sisi server dengan AWS Key Management Service () kunci (SSE-KMS AWS KMS), enkripsi sisi server dua lapis dengan kunci (DSSE-KMS), atau enkripsi sisi server dengan AWS KMS kunci yang disediakan pelanggan (SSE-C). Jika Anda ingin mengatur konfigurasi enkripsi default yang berbeda di dalam bucket tujuan, Anda dapat menggunakan SSE-KMS atau DSSE-KMS.

Untuk informasi selengkapnya tentang mengubah konfigurasi enkripsi default untuk bucket tujuan umum Anda, lihatMengonfigurasi enkripsi default.

Saat Anda mengubah konfigurasi enkripsi default bucket ke SSE-KMS, jenis enkripsi objek Amazon S3 yang ada di bucket tidak akan diubah. Untuk mengubah jenis enkripsi objek yang sudah ada sebelumnya setelah memperbarui konfigurasi enkripsi default ke SSE-KMS, Anda dapat menggunakan Operasi Batch Amazon S3. Anda menyediakan Operasi Batch S3 dengan daftar objek, dan Operasi Batch memanggil operasi API masing-masing. Anda dapat menggunakan Menyalin objek tindakan untuk menyalin objek yang ada, yang menuliskannya kembali ke bucket yang sama dengan objek terenkripsi SSE-KMS. Satu tugas Operasi Batch dapat melakukan operasi tertentu pada miliaran objek yang berisi data sebesar eksabita. Untuk informasi selengkapnya, lihat Melakukan operasi objek secara massal dengan Operasi Batch dan posting Blog AWS Penyimpanan Cara mengenkripsi objek yang ada secara surut di Amazon S3 menggunakan S3 Inventory, Amazon Athena, dan Operasi Batch S3.

Untuk mengkonfigurasi enkripsi sisi server dengan menggunakan REST pembuatan objek APIs, Anda harus memberikan header permintaan. `x-amz-server-side-encryption` Untuk informasi tentang REST APIs, lihatPenggunaan API REST.

Amazon S3 berikut APIs mendukung header ini:

- Operasi PUT–Tentukan header permintaan saat mengunggah data menggunakan API PUT. Untuk informasi lebih lanjut, lihat PUT Objek.

- Mulai Unggahan Multibagian–Menentukan header dalam permintaan awal saat mengunggah objek besar menggunakan operasi API multibagian. Untuk informasi lebih lanjut, lihat Mulai Unggahan Multibagian.
- Operasi COPY–Saat Anda menyalin objek, Anda memiliki objek sumber dan objek target. Untuk informasi lebih lanjut, lihat PUT Objek-Salin.

> **Note**
>
> Saat menggunakan operasi `POST` untuk mengunggah objek, alih-alih memberikan header permintaan, Anda memberikan informasi yang sama di kolom formulir. Untuk informasi lebih lanjut, lihat Objek POST.

AWS SDKs Juga menyediakan pembungkus APIs yang dapat Anda gunakan untuk meminta enkripsi sisi server. Anda juga dapat menggunakan AWS Management Console untuk mengunggah objek dan meminta enkripsi sisi server.

Untuk informasi lebih umum, lihat AWS KMS konsep dalam AWS Key Management Service Panduan Pengembang.

Topik

- Menentukan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3)

Menentukan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3)

Semua bucket Amazon S3 memiliki enkripsi yang dikonfigurasi secara default, dan semua objek baru yang diunggah ke bucket S3 secara otomatis dienkripsi saat sedang tidak aktif. Enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3) adalah konfigurasi enkripsi default untuk setiap bucket di Amazon S3. Untuk menggunakan jenis enkripsi yang berbeda, Anda dapat menentukan jenis enkripsi sisi server yang akan digunakan dalam `PUT` permintaan S3, atau memperbarui konfigurasi enkripsi default di bucket tujuan.

Jika Anda ingin menentukan jenis enkripsi yang berbeda dalam `PUT` permintaan Anda, Anda dapat menggunakan enkripsi sisi server dengan AWS Key Management Service () kunci (SSE-KMS AWS KMS), enkripsi sisi server dua lapis dengan kunci (DSSE-KMS), atau enkripsi sisi server dengan AWS KMS kunci yang disediakan pelanggan (SSE-C). Jika Anda ingin mengatur konfigurasi enkripsi default yang berbeda di dalam bucket tujuan, Anda dapat menggunakan SSE-KMS atau DSSE-KMS.

Untuk informasi selengkapnya tentang mengubah konfigurasi enkripsi default untuk bucket tujuan umum Anda, lihatMengonfigurasi enkripsi default.

Saat Anda mengubah konfigurasi enkripsi default bucket ke SSE-KMS, jenis enkripsi objek Amazon S3 yang ada di bucket tidak akan diubah. Untuk mengubah jenis enkripsi objek yang sudah ada sebelumnya setelah memperbarui konfigurasi enkripsi default ke SSE-KMS, Anda dapat menggunakan Operasi Batch Amazon S3. Anda menyediakan Operasi Batch S3 dengan daftar objek, dan Operasi Batch memanggil operasi API masing-masing. Anda dapat menggunakan Menyalin objek tindakan untuk menyalin objek yang ada, yang menuliskannya kembali ke bucket yang sama dengan objek terenkripsi SSE-KMS. Satu tugas Operasi Batch dapat melakukan operasi tertentu pada miliaran objek yang berisi data sebesar eksabita. Untuk informasi selengkapnya, lihat Melakukan operasi objek secara massal dengan Operasi Batch dan posting Blog AWS Penyimpanan Cara mengenkripsi objek yang ada secara surut di Amazon S3 menggunakan S3 Inventory, Amazon Athena, dan Operasi Batch S3.

Anda dapat menentukan SSE-S3 dengan menggunakan konsol S3, REST APIs,, dan (). AWS SDKs AWS Command Line Interface AWS CLI Untuk informasi selengkapnya, lihat Mengatur perilaku enkripsi sisi server default untuk bucket Amazon S3.

Menggunakan konsol S3

Topik ini menjelaskan cara mengatur atau mengubah jenis enkripsi objek dengan menggunakan AWS Management Console. Saat menyalin objek dengan menggunakan konsol, Amazon S3 menyalin objek seperti apa adanya. Itu berarti bahwa jika objek sumber dienkripsi, objek target juga dienkripsi. Anda dapat menggunakan konsol untuk menambahkan atau mengubah enkripsi untuk suatu objek.

> **Note**
>
> - Anda dapat mengubah enkripsi objek jika objek Anda kurang dari 5 GB. Jika objek Anda lebih besar dari 5 GB, Anda harus menggunakan AWS CLIatau AWS SDKsuntuk mengubah enkripsi objek.
> - Untuk daftar izin tambahan yang diperlukan untuk mengubah enkripsi objek, lihatthe section called “Izin yang diperlukan untuk operasi API S3”. Misalnya kebijakan yang memberikan izin ini, lihatthe section called “Contoh kebijakan berbasis identitas”.
> - Jika Anda mengubah enkripsi objek, sebuah objek baru akan dibuat untuk menggantikan objek yang lama. Jika Penentuan Versi S3 diaktifkan, versi baru objek akan dibuat, dan

objek yang sudah ada menjadi versi yang lebih lama. Peran yang mengubah properti juga menjadi pemilik objek baru (atau versi objek).

Untuk mengubah enkripsi untuk objek

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi, pilih Bucket, lalu pilih tab Bucket tujuan umum. Arahkan ke bucket Amazon S3 atau folder yang berisi objek yang ingin Anda ubah.
3. Pilih kotak centang untuk objek yang ingin Anda ubah.
4. Pada menu Tindakan, pilih Edit enkripsi sisi server dari daftar opsi yang muncul.
5. Gulir ke bagian enkripsi sisi server.
6. Di bawah Pengaturan enkripsi, pilih Gunakan pengaturan bucket untuk enkripsi default, atau Ganti pengaturan bucket untuk enkripsi default.
7. Jika Anda memilih Timpa pengaturan bucket untuk enkripsi default, konfigurasikan pengaturan enkripsi berikut.

   - Di bawah Jenis enkripsi, pilih Enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3). SSE-S3 menggunakan salah satu penyandian blok terkuat—Advanced Encryption Standard 256-bit (AES-256) untuk mengenkripsi setiap objek. Untuk informasi selengkapnya, lihat Menggunakan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3).
8. Di bawah Pengaturan salinan tambahan, pilih apakah Anda ingin Menyalin setelan sumber, Jangan tentukan pengaturan, atau Tentukan pengaturan. Pengaturan sumber salin adalah opsi default. Jika Anda hanya ingin menyalin objek tanpa atribut pengaturan sumber, pilih Jangan tentukan pengaturan. Pilih Tentukan pengaturan untuk menentukan pengaturan untuk kelas penyimpanan, tag objek ACLs, metadata, enkripsi sisi server, dan checksum tambahan.
9. Pilih Simpan perubahan.

> **Note**
>
> Tindakan ini menerapkan enkripsi untuk semua objek yang ditentukan. Saat mengenkripsi folder, tunggu hingga operasi penyimpanannya selesai sebelum menambahkan objek baru ke folder tersebut.

Penggunaan API REST

Saat pembuatan objek—yaitu, saat Anda mengunggah objek baru atau membuat salinan objek yang sudah ada—Anda dapat menentukan apakah Anda ingin Amazon S3 mengenkripsi data Anda dengan kunci yang dikelola Amazon S3 (SSE-S3) dengan menambahkan header ke permintaan `x-amz-server-side-encryption`. Mengatur nilai header ke algoritma enkripsi `AES256`, yang didukung Amazon S3. Amazon S3 mengonfirmasi bahwa objek Anda disimpan dengan SSE-S3 dengan mengembalikan header respons `x-amz-server-side-encryption`.

Operasi API unggahan REST berikut ini menerima header permintaan `x-amz-server-side-encryption`.

- PUT Objek
- PUT Objek-Salin
- Objek POST
- Mulai Unggahan Multibagian

Saat mengunggah objek besar menggunakan operasi API unggahan multibagian, Anda dapat menentukan enkripsi di sisi server dengan menambahkan `x-amz-server-side-encryption` header ke permintaan Mulai Pengunggahan Multibagian. Saat Anda menyalin objek yang ada, terlepas dari apakah objek sumbernya dienkripsi atau tidak, objek tujuan tidak dienkripsi kecuali jika Anda secara eksplisit meminta enkripsi di sisi server.

Header respons dari operasi API REST berikut mengembalikan `x-amz-server-side-encryption` header saat objek disimpan menggunakan SSE-S3.

- PUT Objek
- PUT Objek-Salin
- Objek POST
- Mulai Unggahan Multibagian
- Unggah Bagian
- Unggah Bagian-Salin
- Selesaikan Unggahan Multibagian
- Dapatkan Objek
- Head Objek

> **Note**
>
> Jangan mengirim header permintaan enkripsi untuk `GET` permintaan dan `HEAD` minta jika objek Anda menggunakan SSE-S3, atau Anda akan mendapatkan kesalahan kode status HTTP 400 (Permintaan Buruk).

Menggunakan AWS SDKs

Saat menggunakan AWS SDKs, Anda dapat meminta Amazon S3 untuk menggunakan enkripsi sisi server dengan kunci enkripsi terkelola Amazon S3 (SSE-S3). Bagian ini memberikan contoh penggunaan AWS SDKs dalam berbagai bahasa. Untuk informasi tentang lainnya SDKs, buka Kode Sampel dan Perpustakaan.

Java

Ketika Anda menggunakan AWS SDK untuk Java untuk meng-upload objek, Anda dapat menggunakan SSE-S3 untuk mengenkripsi itu. Untuk meminta enkripsi di sisi server, gunakan properti `ObjectMetadata` dari `PutObjectRequest` untuk menetapkan header permintaan `x-amz-server-side-encryption`. Saat Anda memanggil `putObject()` metode dari `AmazonS3Client`, Amazon S3 mengenkripsi dan menyimpan data.

Anda juga dapat meminta enkripsi SSE-S3 saat mengunggah objek menggunakan operasi API pengunggahan multibagian:

- Saat menggunakan operasi API pengunggahan multibagian tingkat tinggi, Anda menggunakan metode `TransferManager` untuk menerapkan enkripsi di sisi server ke objek saat Anda mengunggahnya. Anda dapat menggunakan metode pengunggahan apa saja yang menggunakan `ObjectMetadata` sebagai parameter. Untuk informasi selengkapnya, lihat Pengunggahan objek menggunakan unggahan multibagian.
- Saat menggunakan operasi API pengunggahan multibagian tingkat rendah, Anda menentukan enkripsi di sisi server saat Anda memulai pengunggahan multibagian. Anda menambahkan properti `ObjectMetadata` dengan memanggil metode `InitiateMultipartUploadRequest.setObjectMetadata()`. Untuk informasi selengkapnya, lihat Menggunakan AWS SDKs (API tingkat rendah).

Anda tidak dapat langsung mengubah status enkripsi suatu objek (mengenkripsi objek yang tidak dienkripsi atau mendekripsi objek yang dienkripsi). Untuk mengubah status enkripsi objek,

Anda membuat salinan objek, dengan menentukan status enkripsi yang diinginkan untuk salinan, lalu hapus objek asli. Amazon S3 mengenkripsi objek yang disalin hanya jika Anda secara eksplisit meminta enkripsi di sisi server. Untuk meminta enkripsi objek yang disalin melalui API Java, gunakan properti `ObjectMetadata` untuk menentukan enkripsi di sisi server dalam `CopyObjectRequest`.

Example Contoh

Contoh berikut menunjukkan cara menetapkan enkripsi di sisi server menggunakan AWS SDK untuk Java. Itu menunjukkan cara melakukan tugas berikut:

- Unggah objek baru menggunakan SSE-S3.
- Ubah status enkripsi objek (dalam contoh ini, mengenkripsi objek yang tidak dienkripsi sebelumnya) dengan membuat salinan objek.
- Periksa status enkripsi objek.

Untuk informasi lebih lanjut tentang enkripsi di sisi server, lihat Penggunaan API REST. Untuk petunjuk cara membuat dan menguji sampel yang berfungsi, lihat Memulai di Panduan AWS SDK untuk Java Pengembang.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.internal.SSEResultBase;
import com.amazonaws.services.s3.model.*;

import java.io.ByteArrayInputStream;

public class SpecifyServerSideEncryption {

    public static void main(String[] args) {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String keyNameToEncrypt = "*** Key name for an object to upload and encrypt
 ***";
        String keyNameToCopyAndEncrypt = "*** Key name for an unencrypted object to
 be encrypted by copying ***";
```

```
        String copiedObjectKeyName = "*** Key name for the encrypted copy of the
unencrypted object ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withRegion(clientRegion)
                    .withCredentials(new ProfileCredentialsProvider())
                    .build();

            // Upload an object and encrypt it with SSE.
            uploadObjectWithSSEEncryption(s3Client, bucketName, keyNameToEncrypt);

            // Upload a new unencrypted object, then change its encryption state
            // to encrypted by making a copy.
            changeSSEEncryptionStatusByCopying(s3Client,
                    bucketName,
                    keyNameToCopyAndEncrypt,
                    copiedObjectKeyName);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }

    private static void uploadObjectWithSSEEncryption(AmazonS3 s3Client, String
bucketName, String keyName) {
        String objectContent = "Test object encrypted with SSE";
        byte[] objectBytes = objectContent.getBytes();

        // Specify server-side encryption.
        ObjectMetadata objectMetadata = new ObjectMetadata();
        objectMetadata.setContentLength(objectBytes.length);

objectMetadata.setSSEAlgorithm(ObjectMetadata.AES_256_SERVER_SIDE_ENCRYPTION);
        PutObjectRequest putRequest = new PutObjectRequest(bucketName,
                keyName,
                new ByteArrayInputStream(objectBytes),
                objectMetadata);
```

```
        // Upload the object and check its encryption status.
        PutObjectResult putResult = s3Client.putObject(putRequest);
        System.out.println("Object \"" + keyName + "\" uploaded with SSE.");
        printEncryptionStatus(putResult);
    }

    private static void changeSSEEncryptionStatusByCopying(AmazonS3 s3Client,
            String bucketName,
            String sourceKey,
            String destKey) {
        // Upload a new, unencrypted object.
        PutObjectResult putResult = s3Client.putObject(bucketName, sourceKey,
"Object example to encrypt by copying");
        System.out.println("Unencrypted object \"" + sourceKey + "\" uploaded.");
        printEncryptionStatus(putResult);

        // Make a copy of the object and use server-side encryption when storing the
        // copy.
        CopyObjectRequest request = new CopyObjectRequest(bucketName,
                sourceKey,
                bucketName,
                destKey);
        ObjectMetadata objectMetadata = new ObjectMetadata();

objectMetadata.setSSEAlgorithm(ObjectMetadata.AES_256_SERVER_SIDE_ENCRYPTION);
        request.setNewObjectMetadata(objectMetadata);

        // Perform the copy operation and display the copy's encryption status.
        CopyObjectResult response = s3Client.copyObject(request);
        System.out.println("Object \"" + destKey + "\" uploaded with SSE.");
        printEncryptionStatus(response);

        // Delete the original, unencrypted object, leaving only the encrypted copy
in
        // Amazon S3.
        s3Client.deleteObject(bucketName, sourceKey);
        System.out.println("Unencrypted object \"" + sourceKey + "\" deleted.");
    }

    private static void printEncryptionStatus(SSEResultBase response) {
        String encryptionStatus = response.getSSEAlgorithm();
        if (encryptionStatus == null) {
            encryptionStatus = "Not encrypted with SSE";
        }
```

```
        System.out.println("Object encryption status is: " + encryptionStatus);
    }
}
```

.NET

Saat mengunggah sebuah objek, Anda dapat mengarahkan Amazon S3 untuk mengenkripsinya. Untuk mengubah status enkripsi objek yang sudah ada, Anda membuat salinan objek dan menghapus objek sumber. Secara default, operasi penyalinan mengenkripsi target hanya jika Anda secara eksplisit meminta enkripsi di sisi server objek target. Untuk menentukan SSE-S3 di `CopyObjectRequest`, tambahkan yang berikut ini:

```
ServerSideEncryptionMethod = ServerSideEncryptionMethod.AES256
```

Untuk sampel kerja tentang cara menyalin objek, lihat Menggunakan AWS SDKs.

Contoh berikut mengunggah objek. Dalam permintaan, contoh tersebut mengarahkan Amazon S3 untuk mengenkripsi objek. Contoh tersebut kemudian mengambil metadata objek dan memverifikasi metode enkripsi yang digunakan. Untuk informasi tentang menyiapkan dan menjalankan contoh kode, lihat Memulai SDK for .NET di AWSAWS SDK for .NET Developer Guide.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class SpecifyServerSideEncryptionTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string keyName = "*** key name for object created ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
```

```
            client = new AmazonS3Client(bucketRegion);
            WritingAnObjectAsync().Wait();
        }

        static async Task WritingAnObjectAsync()
        {
            try
            {
                var putRequest = new PutObjectRequest
                {
                    BucketName = bucketName,
                    Key = keyName,
                    ContentBody = "sample text",
                    ServerSideEncryptionMethod = ServerSideEncryptionMethod.AES256
                };

                var putResponse = await client.PutObjectAsync(putRequest);

                // Determine the encryption state of an object.
                GetObjectMetadataRequest metadataRequest = new
GetObjectMetadataRequest
                {
                    BucketName = bucketName,
                    Key = keyName
                };
                GetObjectMetadataResponse response = await
client.GetObjectMetadataAsync(metadataRequest);
                ServerSideEncryptionMethod objectEncryption =
response.ServerSideEncryptionMethod;

                Console.WriteLine("Encryption method used: {0}",
objectEncryption.ToString());
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered ***. Message:'{0}' when writing
an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
        }
```

```
    }
}
```

PHP

Topik ini menunjukkan cara menggunakan kelas dari versi 3 AWS SDK untuk PHP untuk menambahkan SSE-S3 ke objek yang Anda unggah ke Amazon S3. Untuk informasi selengkapnya tentang AWS SDK for Ruby API, AWS buka SDK for Ruby - Versi 2.

Untuk mengunggah objek ke Amazon S3, gunakan metode Aws\S3\S3Client::putObject(). Untuk menambahkan `x-amz-server-side-encryption` header permintaan ke permintaan pengunggahan, tentukan parameter `ServerSideEncryption` dengan nilai `AES256`, seperti yang ditunjukkan dalam contoh kode berikut. Untuk informasi lebih lanjut tentang permintaan enkripsi di sisi server, lihat Penggunaan API REST.

```
 require 'vendor/autoload.php';

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

// $filepath should be an absolute path to a file on disk.
$filepath = '*** Your File Path ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// Upload a file with server-side encryption.
$result = $s3->putObject([
    'Bucket'               => $bucket,
    'Key'                  => $keyname,
    'SourceFile'           => $filepath,
    'ServerSideEncryption' => 'AES256',
]);
```

Sebagai respons, Amazon S3 mengembalikan header `x-amz-server-side-encryption` dengan nilai algoritma enkripsi yang digunakan untuk mengenkripsi data objek Anda.

Saat mengunggah objek besar dengan menggunakan operasi API unggahan multibagian, Anda dapat menentukan SSE-S3 untuk objek yang Anda unggah, sebagai berikut ini:

- Saat Anda menggunakan operasi API unggahan multibagian tingkat rendah, tentukan enkripsi sisi server saat Anda memanggil metode Aws\ S3\ S3Client:: (). createMultipartUpload Untuk menambahkan header permintaan `x-amz-server-side-encryption` untuk permintaan Anda, tentukan parameter `array` kunci `ServerSideEncryption` dengan nilai `AES256`. Untuk informasi selengkapnya tentang operasi API unggahan multibagian tingkat rendah, lihat Menggunakan AWS SDKs (API tingkat rendah).
- Saat Anda menggunakan operasi API unggahan multibagian tingkat tinggi, tentukan enkripsi sisi server dengan menggunakan `ServerSideEncryption` parameter operasi API. CreateMultipartUpload Untuk contoh penggunaan metode `setOption()` dengan operasi API unggahan multibagian tingkat tinggi, lihat Pengunggahan objek menggunakan unggahan multibagian.

Untuk menentukan status enkripsi objek yang ada, ambil metadata objek dengan memanggil metode Aws\S3\S3Client::headObject() seperti yang ditunjukkan dalam contoh kode PHP berikut.

```
 require 'vendor/autoload.php';

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// Check which server-side encryption algorithm is used.
$result = $s3->headObject([
    'Bucket' => $bucket,
    'Key'    => $keyname,
]);
echo $result['ServerSideEncryption'];
```

Untuk mengubah status enkripsi objek yang sudah ada, buat salinan objek menggunakan metode Aws\S3\S3Client::copyObject() dan menghapus objek sumber. Secara default, `copyObject()` tidak mengenkripsi target kecuali jika Anda secara eksplisit meminta enkripsi di sisi server objek tujuan menggunakan parameter `ServerSideEncryption` dengan nilai `AES256`. Contoh kode PHP berikut membuat salinan objek dan menambahkan enkripsi server-side ke objek yang tersalin.

```
 require 'vendor/autoload.php';

use Aws\S3\S3Client;

$sourceBucket = '*** Your Source Bucket Name ***';
$sourceKeyname = '*** Your Source Object Key ***';

$targetBucket = '*** Your Target Bucket Name ***';
$targetKeyname = '*** Your Target Object Key ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// Copy an object and add server-side encryption.
$s3->copyObject([
    'Bucket'               => $targetBucket,
    'Key'                  => $targetKeyname,
    'CopySource'           => "$sourceBucket/$sourceKeyname",
    'ServerSideEncryption' => 'AES256',
]);
```

Untuk informasi selengkapnya, lihat topik berikut:

- AWS SDK untuk PHP untuk Amazon S3 Aws\ S3\ S3Kelas Klien
- Dokumentasi AWS SDK untuk PHP

Ruby

Saat menggunakan AWS SDK untuk Ruby untuk mengunggah objek, Anda dapat menentukan bahwa objek disimpan terenkripsi saat istirahat dengan SSE-S3. Saat Anda membaca kembali, objek tersebut akan didekripsi secara otomatis.

Contoh AWS SDK untuk Ruby Versi 3 berikut menunjukkan cara menentukan bahwa file yang diunggah ke Amazon S3 dienkripsi saat istirahat.

```
require 'aws-sdk-s3'

# Wraps Amazon S3 object actions.
class ObjectPutSseWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  def put_object_encrypted(object_content, encryption)
    @object.put(body: object_content, server_side_encryption: encryption)
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't put your content to #{object.key}. Here's why: #{e.message}"
    false
  end
end

# Example usage:
def run_demo
  bucket_name = "amzn-s3-demo-bucket"
  object_key = "my-encrypted-content"
  object_content = "This is my super-secret content."
  encryption = "AES256"

  wrapper = ObjectPutSseWrapper.new(Aws::S3::Object.new(bucket_name,
 object_content))
  return unless wrapper.put_object_encrypted(object_content, encryption)

  puts "Put your content into #{bucket_name}:#{object_key} and encrypted it with
 #{encryption}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Contoh kode berikut menunjukkan cara menentukan status enkripsi objek yang ada.

```
require 'aws-sdk-s3'
```

```
# Wraps Amazon S3 object actions.
class ObjectGetEncryptionWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  # Gets the object into memory.
  #
  # @return [Aws::S3::Types::GetObjectOutput, nil] The retrieved object data if
 successful; otherwise nil.
  def object
    @object.get
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't get object #{@object.key}. Here's why: #{e.message}"
  end
end

# Example usage:
def run_demo
  bucket_name = "amzn-s3-demo-bucket"
  object_key = "my-object.txt"

  wrapper = ObjectGetEncryptionWrapper.new(Aws::S3::Object.new(bucket_name,
 object_key))
  obj_data = wrapper.get_object
  return unless obj_data

  encryption = obj_data.server_side_encryption.nil? ? 'no' :
 obj_data.server_side_encryption
  puts "Object #{object_key} uses #{encryption} encryption."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Jika enkripsi di sisi server tidak digunakan untuk objek yang disimpan di Amazon S3, metode tersebut akan menghasilkan `null`.

Untuk mengubah status enkripsi objek yang ada, buat salinan objek tersebut dan hapus objek sumber. Secara default, metode penyalinan tidak mengenkripsi target kecuali Anda secara

eksplisit meminta enkripsi di sisi server. Anda dapat meminta enkripsi objek target dengan menentukan nilai `server_side_encryption` dalam argumen hash opsi, seperti yang ditunjukkan dalam contoh kode Ruby berikut. Contoh kode mendemonstrasikan cara menyalin sebuah objek dan mengenkripsi salinan dengan SSE-S3.

```
require 'aws-sdk-s3'

# Wraps Amazon S3 object actions.
class ObjectCopyEncryptWrapper
  attr_reader :source_object

  # @param source_object [Aws::S3::Object] An existing Amazon S3 object. This is
 used as the source object for
  #                                       copy actions.
  def initialize(source_object)
    @source_object = source_object
  end

  # Copy the source object to the specified target bucket, rename it with the target
 key, and encrypt it.
  #
  # @param target_bucket [Aws::S3::Bucket] An existing Amazon S3 bucket where the
 object is copied.
  # @param target_object_key [String] The key to give the copy of the object.
  # @return [Aws::S3::Object, nil] The copied object when successful; otherwise,
 nil.
  def copy_object(target_bucket, target_object_key, encryption)
    @source_object.copy_to(bucket: target_bucket.name, key: target_object_key,
 server_side_encryption: encryption)
    target_bucket.object(target_object_key)
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't copy #{@source_object.key} to #{target_object_key}. Here's why:
 #{e.message}"
  end
end

# Example usage:
def run_demo
  source_bucket_name = "amzn-s3-demo-bucket1"
  source_key = "my-source-file.txt"
  target_bucket_name = "amzn-s3-demo-bucket2"
  target_key = "my-target-file.txt"
  target_encryption = "AES256"
```

```
  source_bucket = Aws::S3::Bucket.new(source_bucket_name)
  wrapper = ObjectCopyEncryptWrapper.new(source_bucket.object(source_key))
  target_bucket = Aws::S3::Bucket.new(target_bucket_name)
  target_object = wrapper.copy_object(target_bucket, target_key, target_encryption)
  return unless target_object

  puts "Copied #{source_key} from #{source_bucket_name} to
 #{target_object.bucket_name}:#{target_object.key} and "\
       "encrypted the target with #{target_object.server_side_encryption}
 encryption."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Menggunakan AWS CLI

Untuk menentukan SSE-S3 saat Anda mengunggah objek dengan menggunakan AWS CLI, gunakan contoh berikut.

```
aws s3api put-object --bucket amzn-s3-demo-bucket1 --key object-key-name --server-side-
encryption AES256  --body file path
```

Untuk informasi selengkapnya, lihat put-object dalam Referensi AWS CLI . Untuk menentukan SSE-S3 saat Anda menyalin objek dengan menggunakan AWS CLI, lihat copy-object.

Menggunakan AWS CloudFormation

Untuk contoh pengaturan enkripsi menggunakan AWS CloudFormation, lihat Membuat bucket dengan enkripsi default dan contoh Membuat bucket dengan menggunakan enkripsi AWS KMS sisi server dengan S3 Bucket Key dalam *`AWS::S3::Bucket ServerSideEncryptionRule`* topik di Panduan Pengguna.AWS CloudFormation

Menggunakan enkripsi sisi server dengan AWS KMS kunci (SSE-KMS)

> **Important**
>
> Amazon S3 sudah menerapkan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3) sebagai tingkat dasar enkripsi bagi setiap bucket di Amazon S3. Mulai 5 Januari 2023, semua unggahan objek baru ke Amazon S3 secara otomatis akan dienkripsi tanpa

biaya tambahan dan tidak akan berdampak pada kinerja. Status enkripsi otomatis untuk konfigurasi enkripsi default bucket S3 dan untuk unggahan objek baru tersedia di AWS CloudTrail log, S3 Inventory, S3 Storage Lens, konsol Amazon S3, dan sebagai header respons API Amazon S3 tambahan di dan. AWS Command Line Interface AWS SDKs Untuk informasi selengkapnya, lihat FAQ enkripsi default.

Enkripsi di sisi server adalah enkripsi data di tempat tujuan oleh aplikasi atau layanan yang menerimanya.

Amazon S3 secara otomatis mengaktifkan enkripsi di sisi server dengan kunci yang dikelola Amazon S3 (SSE-S3) untuk unggah objek baru.

Kecuali Anda menentukan sebaliknya, bucket menggunakan SSE-S3 secara default untuk mengenkripsi objek. Namun, Anda dapat memilih untuk mengonfigurasi bucket untuk menggunakan enkripsi sisi server dengan kunci AWS Key Management Service (AWS KMS) (SSE-KMS) sebagai gantinya. Untuk informasi selengkapnya, lihat Menentukan enkripsi di sisi server dengan AWS KMS (SSE-KMS).

AWS KMS adalah layanan yang menggabungkan perangkat keras dan perangkat lunak yang aman dan sangat tersedia untuk menyediakan sistem manajemen kunci yang diskalakan untuk cloud. Amazon S3 menggunakan enkripsi sisi server dengan AWS KMS (SSE-KMS) untuk mengenkripsi data objek S3 Anda. Juga, ketika SSE-KMS diminta untuk objek, checksum S3 (sebagai bagian dari metadata objek) disimpan dalam bentuk terenkripsi. Untuk informasi selengkapnya tentang checksum, silakan lihat Memeriksa integritas objek di Amazon S3.

Jika Anda menggunakan kunci KMS, Anda dapat menggunakan AWS KMS melalui AWS Management Consoleatau AWS KMS API untuk melakukan hal berikut:

- Membuat, melihat, mengedit, memantau, mengaktifkan atau menonaktifkan, memutar, dan menjadwalkan penghapusan kunci KMS secara terpusat.
- Menentukan kebijakan yang mengendalikan bagaimana dan oleh siapa kunci KMS digunakan.
- Audit penggunaan kunci KMS untuk penggunaan yang benar. Audit didukung oleh AWS KMS API tetapi tidak oleh AWS KMS Konsol;.

Kontrol keamanan AWS KMS dapat membantu Anda memenuhi persyaratan kepatuhan terkait enkripsi. Anda dapat menggunakan kunci KMS ini untuk melindungi data Anda di bucket Amazon

S3. Saat Anda menggunakan enkripsi SSE-KMS dengan bucket S3, AWS KMS keys harus berada di Wilayah yang sama dengan bucket.

Ada biaya tambahan untuk penggunaan AWS KMS keys. Untuk informasi lebih umum, lihat konsep AWS KMS key dalam AWS Key Management Service Panduan Pengguna dan AWS KMS harga.

Untuk petunjuk tentang mengizinkan pengguna IAM mengakses bucket terenkripsi KMS, lihat Bucket Amazon S3 saya memiliki enkripsi default menggunakan kunci khusus. AWSAWS KMS Bagaimana cara mengizinkan pengguna mengunduh dan mengunggah ke bucket? di pusat AWS re:Post pengetahuan.

Izin

Agar berhasil membuat `PutObject` permintaan untuk mengenkripsi objek dengan AWS KMS kunci ke Amazon S3, Anda `kms:GenerateDataKey` memerlukan izin pada kunci tersebut. Untuk mengunduh objek yang dienkripsi dengan AWS KMS key, Anda memerlukan `kms:Decrypt` izin untuk kunci tersebut. Untuk melakukan unggahan multibagian untuk mengenkripsi objek dengan AWS KMS key, Anda harus memiliki `kms:GenerateDataKey` dan `kms:Decrypt` izin untuk kunci tersebut.

> **Important**
>
> Tinjau dengan cermat izin yang diberikan dalam kebijakan kunci KMS Anda. Selalu batasi izin kebijakan kunci KMS yang dikelola pelanggan hanya untuk kepala sekolah dan AWS layanan IAM yang harus mengakses tindakan kunci yang relevan. AWS KMS Untuk informasi selengkapnya, lihat Kebijakan utama di AWS KMS.

Topik

- AWS KMS keys
- Kunci Bucket Amazon S3
- Membutuhkan enkripsi di sisi server
- Konteks enkripsi
- Mengirim permintaan untuk objek AWS KMS terenkripsi
- Menentukan enkripsi di sisi server dengan AWS KMS (SSE-KMS)
- Mengurangi biaya SSE-KMS dengan Kunci Bucket Amazon S3

AWS KMS keys

Bila Anda menggunakan enkripsi sisi server dengan AWS KMS (SSE-KMS), Anda dapat menggunakan kunci AWS terkelola default, atau Anda dapat menentukan kunci terkelola pelanggan yang telah Anda buat. AWS KMS mendukung enkripsi amplop. S3 menggunakan AWS KMS fitur untuk enkripsi amplop untuk lebih melindungi data Anda. Enkripsi amplop adalah praktik mengenkripsi data teks biasa Anda dengan kunci data, dan kemudian mengenkripsi kunci data itu dengan kunci KMS. Untuk informasi selengkapnya tentang enkripsi amplop, lihat Enkripsi amplop di Panduan Developer AWS Key Management Service .

Jika Anda tidak menentukan kunci terkelola pelanggan, Amazon S3 secara otomatis membuat saat pertama kali Anda Akun AWS menambahkan objek yang dienkripsi dengan SSE-KMS ke bucket. Kunci yang dikelola AWS Secara default, Amazon S3 menggunakan kunci KMS ini untuk SSE-KMS.

> **Note**
>
> Objek yang dienkripsi menggunakan SSE-KMS dengan Kunci yang dikelola AWS tidak dapat dibagikan lintas akun. Jika Anda perlu membagikan data lintas akun SSE-KMS, Anda harus menggunakan kunci yang dikelola pelanggan dari. AWS KMS

Jika Anda ingin menggunakan CMK untuk SSE-KMS, Anda dapat membuat CMK simetris sebelum Anda mengonfigurasi SSE-KMS. Kemudian, ketika Anda mengonfigurasi SSE-KMS untuk bucket Anda, Anda dapat menentukan CMK saat ini. Untuk informasi selengkapnya tentang kunci enkripsi simetris, lihat Kunci KMS enkripsi simetris dalam Panduan Pengembang AWS Key Management Service .

Membuat CMK memberikan Anda lebih banyak fleksibilitas dan kendali atas pelanggan. Misalnya, Anda dapat membuat, memutar, dan menonaktifkan CMK. Anda juga dapat menentukan kontrol akses dan mengaudit CMK yang Anda gunakan untuk melindungi data Anda. Untuk informasi selengkapnya tentang kunci yang AWS dikelola dan dikelola pelanggan, lihat Kunci dan AWS kunci pelanggan di Panduan AWS Key Management Service Pengembang.

> **Note**
>
> Saat Anda menggunakan enkripsi di sisi server dengan CMK yang disimpan di penyimpanan kunci eksternal, tidak seperti kunci KMS standar, Anda bertanggung jawab untuk memastikan ketersediaan dan ketahanan material kunci Anda. Untuk informasi selengkapnya tentang

penyimpanan kunci eksternal dan cara mereka mengubah model tanggung jawab bersama, lihat Penyimpanan kunci eksternal di Panduan AWS Key Management Service Pengembang.

Menggunakan enkripsi SSE-KMS untuk operasi lintas akun

Saat menggunakan enkripsi untuk operasi lintas akun, perhatikan hal berikut:

- Jika Nama Sumber Daya AWS KMS key Amazon (ARN) atau alias tidak disediakan pada waktu permintaan atau melalui konfigurasi enkripsi default bucket, Kunci yang dikelola AWS (`aws/s3`) akan digunakan.
- Jika Anda mengunggah atau mengakses objek S3 dengan menggunakan prinsipal AWS Identity and Access Management (IAM) yang sama Akun AWS dengan kunci KMS Anda, Anda dapat menggunakan (). Kunci yang dikelola AWS `aws/s3`
- Jika Anda ingin memberikan akses lintas akun ke objek S3 Anda, gunakan CMK. Anda dapat mengonfigurasi kebijakan CMK untuk mengizinkan akses dari akun lain.
- Jika Anda menentukan kunci KMS yang dikelola pelanggan, kami sarankan untuk menggunakan ARN kunci KMS yang memenuhi syarat sepenuhnya. Jika Anda menggunakan alias kunci KMS sebagai gantinya, AWS KMS selesaikan kunci dalam akun pemohon. Perilaku ini dapat menghasilkan data yang dienkripsi dengan kunci KMS milik pemohon, dan bukan pemilik bucket.
- Anda harus menentukan kunci yang kepada Anda (pemohon) telah diberikan izin `Encrypt`. Untuk informasi selengkapnya, lihat Mengizinkan pengguna di akun lain untuk menggunakan kunci KMS untuk operasi kriptografi di Panduan Pengembang AWS Key Management Service .

Untuk informasi selengkapnya tentang kapan menggunakan kunci terkelola pelanggan dan kunci KMS AWS terkelola, lihat Haruskah saya menggunakan Kunci yang dikelola AWS atau kunci yang dikelola pelanggan untuk mengenkripsi objek saya di Amazon S3?

Alur kerja enkripsi SSE-KMS

Jika Anda memilih untuk mengenkripsi data menggunakan Kunci yang dikelola AWS atau kunci yang dikelola pelanggan, AWS KMS dan Amazon S3 melakukan tindakan enkripsi amplop berikut:

1. Amazon S3 meminta kunci data plaintext dan salinan kunci yang dienkripsi di bawah kunci KMS yang ditentukan.
2. AWS KMS menghasilkan kunci data, mengenkripsinya di bawah kunci KMS, dan mengirimkan kunci data teks biasa dan kunci data terenkripsi ke Amazon S3.

3. Amazon S3 mengenkripsi data menggunkaan kunci data dan menghapus kunci plaintext dari memori sesegera mungkin setelah digunakan.
4. Amazon S3 menyimpan kunci data terenkripsi sebagai metadata dengan data terenkripsi.

Saat Anda meminta agar data Anda didekripsi, Amazon S3 AWS KMS dan lakukan tindakan berikut:

1. Amazon S3 mengirimkan kunci data terenkripsi ke AWS KMS dalam permintaan. `Decrypt`
2. AWS KMS mendekripsi kunci data terenkripsi dengan menggunakan kunci KMS yang sama dan mengembalikan kunci data teks biasa ke Amazon S3.
3. Amazon S3 mendekripsi data terenkripsi, menggunakan kunci data teks biasa, dan menghapus kunci data teks biasa dari memori sesegera mungkin.

> **Important**
>
> Saat Anda menggunakan enkripsi sisi server AWS KMS key untuk Amazon S3, Anda harus memilih kunci KMS enkripsi simetris. Amazon S3 hanya mendukung kunci KMS enkripsi simetris. Untuk informasi selengkapnya terkait kunci ini, lihat Membuat kunci enkripsi simetris KMS dalam Panduan Pengembang AWS Key Management Service .

Mengaudit enkripsi SSE-KMS

Untuk mengidentifikasi permintaan yang menentukan SSE-KMS, Anda dapat menggunakan metrik Semua permintaan SSE-KMS dan % semua permintaan SSE-KMS dalam metrik Lensa Penyimpanan Amazon S3. Lensa Penyimpanan S3 adalah fitur analisis penyimpanan cloud yang dapat Anda gunakan untuk mendapatkan visibilitas seluruh organisasi ke dalam penggunaan dan aktivitas penyimpanan objek. Anda juga dapat menggunakan jumlah bucket berkemampuan SSE-KMS dan bucket berkemampuan% SSE-KMS untuk memahami jumlah bucket yang (SSE-KMS) untuk enkripsi bucket default. Untuk informasi selengkapnya, lihat Menilai aktivitas penyimpanan dan penggunaan Anda dengan Lensa Penyimpanan S3. Untuk daftar lengkap metrik, lihat Glosarium metrik Lensa Penyimpanan S3.

Untuk mengaudit penggunaan AWS KMS kunci Anda untuk data terenkripsi SSE-KMS Anda, Anda dapat menggunakan log. AWS CloudTrail Anda bisa mendapatkan wawasan tentang operasi kriptografi Anda, seperti GenerateDataKeydan Decrypt. CloudTrail mendukung banyak nilai atribut untuk memfilter pencarian Anda, termasuk nama acara, nama pengguna, dan sumber acara.

Kunci Bucket Amazon S3

Saat mengonfigurasi enkripsi sisi server menggunakan AWS KMS (SSE-KMS), Anda dapat mengonfigurasi bucket untuk menggunakan S3 Bucket Keys untuk SSE-KMS. Menggunakan kunci tingkat ember untuk SSE-KMS dapat mengurangi biaya AWS KMS permintaan Anda hingga 99 persen dengan mengurangi lalu lintas permintaan dari Amazon S3 ke. AWS KMS

Ketika Anda mengonfigurasi bucket Anda untuk menggunakan Kunci Bucket S3 untuk SSE-KMS pada objek baru, AWS KMS menghasilkan kunci tingkat bucket yang digunakan untuk membuat kunci data untuk objek dalam bucket. Kunci Bucket S3 ini digunakan untuk jangka waktu terbatas dalam Amazon S3, semakin mengurangi kebutuhan Amazon S3 untuk membuat permintaan AWS KMS untuk menyelesaikan operasi enkripsi. Untuk informasi lebih lanjut tentang menggunakan Kunci Bucket S3, lihat Mengurangi biaya SSE-KMS dengan Kunci Bucket Amazon S3.

Membutuhkan enkripsi di sisi server

Untuk mewajibkan enkripsi di sisi server terhadap semua objek dalam bucket Amazon S3 tertentu, Anda dapat menggunakan kebijakan bucket. Misalnya, kebijakan bucket berikut menolak izin objek unggah (`s3:PutObject`) untuk semua orang jika permintaan tidak menyertakan header `x-amz-server-side-encryption-aws-kms-key-id` yang meminta enkripsi di sisi server dengan SSE-KMS.

```
{
   "Version":"2012-10-17",
   "Id":"PutObjectPolicy",
   "Statement":[{
         "Sid":"DenyObjectsThatAreNotSSEKMS",
         "Effect":"Deny",
         "Principal":"*",
         "Action":"s3:PutObject",
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/*",
         "Condition":{
            "Null":{
               "s3:x-amz-server-side-encryption-aws-kms-key-id":"true"
            }
         }
      }
   ]
}
```

Untuk mengharuskan seseorang AWS KMS key digunakan untuk mengenkripsi objek dalam ember, Anda dapat menggunakan kunci `s3:x-amz-server-side-encryption-aws-kms-key-id` kondisi. Untuk menentukan kunci KMS, Anda harus menggunakan kunci Amazon Resource Name (ARN) yang ada dalam `arn:aws:kms:region:acct-id:key/key-id` format. AWS Identity and Access Management tidak memvalidasi jika string untuk `s3:x-amz-server-side-encryption-aws-kms-key-id` ada.

> **Note**
>
> Saat mengunggah objek, Anda dapat menentukan kunci KMS dengan menggunakan `x-amz-server-side-encryption-aws-kms-key-id` header atau mengandalkan konfigurasi enkripsi bucket default Anda. Jika PutObject permintaan Anda menentukan `aws:kms` di `x-amz-server-side-encryption` header, tetapi tidak menentukan `x-amz-server-side-encryption-aws-kms-key-id` header, maka Amazon S3 mengasumsikan bahwa Anda ingin menggunakan. Kunci yang dikelola AWS Terlepas dari itu, ID AWS KMS kunci yang digunakan Amazon S3 untuk enkripsi objek harus cocok dengan ID AWS KMS kunci dalam kebijakan, jika tidak Amazon S3 menolak permintaan tersebut.

Untuk daftar lengkap kunci kondisi khusus Amazon S3, lihat Kunci kondisi untuk Amazon S3 di Referensi Otorisasi Layanan.

Konteks enkripsi

Konteks enkripsi adalah seperangkat pasangan nilai kunci yang berisi informasi kontekstual tambahan terkait data. Konteks enkripsi tidak dienkripsi. Ketika konteks enkripsi ditentukan untuk operasi enkripsi, Amazon S3 harus menentukan konteks enkripsi yang sama untuk operasi dekripsi. Jika tidak, dekripsi gagal. AWS KMS menggunakan konteks enkripsi sebagai data otentikasi tambahan (AAD) untuk mendukung enkripsi yang diautentikasi. Untuk informasi umum tentang konteks enkripsi, lihat Konteks enkripsi dalam Panduan Pengembang AWS Key Management Service .

Secara default, Amazon S3 menggunakan objek atau bucket Amazon Resource Name (ARN) sebagai pasangan konteks enkripsi:

- Jika Anda menggunakan SSE-KMS tanpa mengaktifkan Kunci Bucket S3, objek ARN digunakan sebagai konteks enkripsi.

```
arn:aws:s3:::object_ARN
```

- Jika Anda menggunakan SSE-KMS dan mengaktifkan Kunci Bucket S3, ARN bucket akan digunakan sebagai konteks enkripsi. Untuk informasi selengkapnya tentang Kunci Bucket S3, lihat Mengurangi biaya SSE-KMS dengan Kunci Bucket Amazon S3.

```
arn:aws:s3:::bucket_ARN
```

Anda secara opsional dapat memberikan pasangan konteks enkripsi tambahan dengan menggunakan `x-amz-server-side-encryption-context` header dalam permintaan s3:PutObject. Namun, karena konteks enkripsi tidak dienkripsi, pastikan bahwa itu tidak menyertakan informasi yang sensitif. Amazon S3 menyimpan pasangan kunci tambahan ini bersama dengan konteks enkripsi default. Saat memproses permintaan PUT Anda, Amazon S3 menambahkan konteks enkripsi default ke `aws:s3:arn` yang Anda berikan.

Anda dapat menggunakan konteks enkripsi untuk mengidentifikasi dan mengategorikan operasi kriptografi Anda. Anda juga dapat menggunakan nilai ARN konteks enkripsi default untuk melacak permintaan yang relevan AWS CloudTrail dengan melihat Amazon S3 ARN mana yang digunakan dengan kunci enkripsi mana.

Di `requestParameters` bidang file CloudTrail log, konteks enkripsi terlihat mirip dengan yang berikut.

```
"encryptionContext": {
    "aws:s3:arn": "arn:aws:s3:::amzn-s3-demo-bucket1/file_name"
}
```

Saat Anda menggunakan SSE-KMS dengan fitur S3 Bucket Keys opsional, nilai konteks enkripsi adalah ARN bucket.

```
"encryptionContext": {
    "aws:s3:arn": "arn:aws:s3:::amzn-s3-demo-bucket1"
}
```

Mengirim permintaan untuk objek AWS KMS terenkripsi

> **Important**
>
> Semua GET dan PUT permintaan untuk objek AWS KMS terenkripsi harus dibuat menggunakan Secure Sockets Layer (SSL) atau Transport Layer Security (TLS). Permintaan juga harus ditandatangani menggunakan kredensil yang valid, seperti AWS Signature Version 4 (atau AWS Signature Version 2).

AWS Signature Version 4 adalah proses menambahkan informasi otentikasi ke AWS permintaan yang dikirim oleh HTTP. Untuk keamanan, sebagian besar permintaan AWS harus ditandatangani dengan kunci akses, yang terdiri dari ID kunci akses dan kunci akses rahasia. Kedua kunci ini umumnya disebut sebagai kredensial keamanan Anda. Untuk informasi selengkapnya, lihat Permintaan Autentikasi (Tanda Tangan AWS Versi 4) dan Proses penandatanganan Tanda Tangan Versi 4.

> **Important**
>
> Jika objek Anda menggunakan SSE-KMS, jangan mengirim header permintaan enkripsi untuk permintaan GET dan permintaan HEAD. Jika tidak, Anda akan mendapatkan kesalahan HTTP 400 Bad Request.

Topik

- Menentukan enkripsi di sisi server dengan AWS KMS (SSE-KMS)
- Mengurangi biaya SSE-KMS dengan Kunci Bucket Amazon S3

Menentukan enkripsi di sisi server dengan AWS KMS (SSE-KMS)

Semua bucket Amazon S3 memiliki enkripsi yang dikonfigurasi secara default, dan semua objek baru yang diunggah ke bucket S3 secara otomatis dienkripsi saat sedang tidak aktif. Enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3) adalah konfigurasi enkripsi default untuk setiap bucket di Amazon S3. Untuk menggunakan jenis enkripsi yang berbeda, Anda dapat menentukan jenis enkripsi sisi server yang akan digunakan dalam PUT permintaan S3, atau memperbarui konfigurasi enkripsi default di bucket tujuan.

Jika Anda ingin menentukan jenis enkripsi yang berbeda dalam PUT permintaan Anda, Anda dapat menggunakan enkripsi sisi server dengan AWS Key Management Service () kunci (SSE-KMS AWS KMS), enkripsi sisi server dua lapis dengan kunci (DSSE-KMS), atau enkripsi sisi server dengan AWS KMS kunci yang disediakan pelanggan (SSE-C). Jika Anda ingin mengatur konfigurasi enkripsi default yang berbeda di dalam bucket tujuan, Anda dapat menggunakan SSE-KMS atau DSSE-KMS.

Untuk informasi selengkapnya tentang mengubah konfigurasi enkripsi default untuk bucket tujuan umum Anda, lihatMengonfigurasi enkripsi default.

Saat Anda mengubah konfigurasi enkripsi default bucket ke SSE-KMS, jenis enkripsi objek Amazon S3 yang ada di bucket tidak akan diubah. Untuk mengubah jenis enkripsi objek yang sudah ada sebelumnya setelah memperbarui konfigurasi enkripsi default ke SSE-KMS, Anda dapat menggunakan Operasi Batch Amazon S3. Anda menyediakan Operasi Batch S3 dengan daftar objek, dan Operasi Batch memanggil operasi API masing-masing. Anda dapat menggunakan Menyalin objek tindakan untuk menyalin objek yang ada, yang menuliskannya kembali ke bucket yang sama dengan objek terenkripsi SSE-KMS. Satu tugas Operasi Batch dapat melakukan operasi tertentu pada miliaran objek yang berisi data sebesar eksabita. Untuk informasi selengkapnya, lihat Melakukan operasi objek secara massal dengan Operasi Batch dan posting Blog AWS Penyimpanan Cara mengenkripsi objek yang ada secara surut di Amazon S3 menggunakan S3 Inventory, Amazon Athena, dan Operasi Batch S3.

Anda dapat menentukan SSE-KMS dengan menggunakan konsol Amazon S3, operasi REST API, dan (). AWS SDKs AWS Command Line Interface AWS CLI Untuk informasi selengkapnya, lihat topik berikut.

> **Note**
>
> Anda dapat menggunakan Multi-wilayah AWS KMS keys di Amazon S3. Namun, Amazon S3 saat ini memperlakukan kunci multi-Wilayah selayaknya kunci satu Wilayah, dan tidak menggunakan fitur multi-Wilayah dari kunci tersebut. Untuk informasi selengkapnya, lihat Menggunakan kunci Multi-wilayah di Panduan AWS Key Management Service Pengembang.

> **Note**
>
> Jika Anda ingin menggunakan kunci KMS yang dimiliki oleh akun lain, Anda harus memiliki izin untuk menggunakan kunci tersebut. Untuk informasi selengkapnya tentang izin lintas

akun untuk kunci KMS, lihat Membuat kunci KMS yang dapat digunakan oleh akun lain di Panduan Pengembang AWS Key Management Service .

Menggunakan konsol S3

Topik ini menjelaskan cara mengatur atau mengubah jenis enkripsi objek untuk menggunakan enkripsi sisi server dengan AWS Key Management Service (AWS KMS) kunci (SSE-KMS) dengan menggunakan konsol Amazon S3.

Note

- Anda dapat mengubah enkripsi objek jika objek Anda kurang dari 5 GB. Jika objek Anda lebih besar dari 5 GB, Anda harus menggunakan AWS CLIatau AWS SDKsuntuk mengubah enkripsi objek.
- Untuk daftar izin tambahan yang diperlukan untuk mengubah enkripsi objek, lihatthe section called “Izin yang diperlukan untuk operasi API S3”. Misalnya kebijakan yang memberikan izin ini, lihatthe section called “Contoh kebijakan berbasis identitas”.
- Jika Anda mengubah enkripsi objek, sebuah objek baru akan dibuat untuk menggantikan objek yang lama. Jika Penentuan Versi S3 diaktifkan, versi baru objek akan dibuat, dan objek yang sudah ada menjadi versi yang lebih lama. Peran yang mengubah properti juga menjadi pemilik objek baru (atau versi objek).

Untuk menambahkan atau mengubah enkripsi untuk objek

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi, pilih Bucket, lalu pilih tab Bucket tujuan umum. Arahkan ke bucket Amazon S3 atau folder yang berisi objek yang ingin Anda ubah.
3. Pilih kotak centang untuk objek yang ingin Anda ubah.
4. Pada menu Tindakan, pilih Edit enkripsi sisi server dari daftar opsi yang muncul.
5. Gulir ke bagian enkripsi sisi server.
6. Di bawah Pengaturan enkripsi, pilih Gunakan pengaturan bucket untuk enkripsi default, atau Ganti pengaturan bucket untuk enkripsi default.

> **Important**
>
> Jika Anda menggunakan opsi SSE-KMS untuk konfigurasi enkripsi default, Anda tunduk pada kuota permintaan per detik (RPS) AWS KMS. Untuk informasi selengkapnya tentang kuota AWS KMS dan cara meminta kenaikan kuota, lihat Kuota di Panduan Pengembang AWS Key Management Service .

7. Jika Anda memilih Timpa pengaturan bucket untuk enkripsi default, konfigurasikan pengaturan enkripsi berikut.

   a. Di bawah Jenis enkripsi, pilih Enkripsi sisi server dengan AWS Key Management Service kunci (SSE-KMS).

   b. Di bagian bawah AWS KMS kunci, lakukan salah satu langkah berikut untuk memilih kunci KMS Anda:

      - Untuk memilih dari daftar kunci KMS yang tersedia, pilih Pilih dari AWS KMS keys Anda, dan pilih Kunci KMS Anda dari daftar kunci yang tersedia.

        Kunci Kunci yang dikelola AWS (`aws/s3`) dan kunci terkelola pelanggan Anda muncul dalam daftar ini. Untuk informasi selengkapnya tentang CMK, lihat Kunci pelanggan dan AWS kunci di AWS Key Management Service Panduan Pengembang.

      - Untuk memasukkan ARN kunci KMS, pilih Masukkan AWS KMS key ARN, lalu masukkan ARN kunci KMS Anda di bidang yang muncul.

      - Untuk membuat kunci terkelola pelanggan baru di AWS KMS konsol, pilih Buat kunci KMS.

        Untuk informasi selengkapnya tentang membuat AWS KMS key, lihat Membuat kunci di Panduan AWS Key Management Service Pengembang.

> **Important**
>
> Anda hanya dapat menggunakan tombol KMS yang tersedia Wilayah AWS sama dengan bucket. Konsol Amazon S3 hanya mencantumkan kunci 100 KMS pertama di Wilayah yang sama dengan bucket. Untuk menggunakan kunci KMS yang tidak terdaftar, Anda harus memasukkan ARN kunci KMS Anda. Jika Anda ingin menggunakan kunci KMS yang dimiliki oleh akun lain, Anda harus terlebih dahulu

memiliki izin untuk menggunakan kunci tersebut, dan kemudian Anda harus memasukkan ARN kunci KMS.
Amazon S3 hanya mendukung kunci KMS enkripsi simetris, dan tidak mendukung kunci KMS asimetris. Untuk informasi selengkapnya, lihat Mengidentifikasi tombol KMS simetris dan asimetris dalam Panduan Pengembang AWS Key Management Service .

8. Di bawah Pengaturan salinan tambahan, pilih apakah Anda ingin Menyalin setelan sumber, Jangan tentukan pengaturan, atau Tentukan pengaturan. Pengaturan sumber salin adalah opsi default. Jika Anda hanya ingin menyalin objek tanpa atribut pengaturan sumber, pilih Jangan tentukan pengaturan. Pilih Tentukan pengaturan untuk menentukan pengaturan untuk kelas penyimpanan, tag objek ACLs, metadata, enkripsi sisi server, dan checksum tambahan.
9. Pilih Simpan perubahan.

> **Note**
>
> Tindakan ini menerapkan enkripsi untuk semua objek yang ditentukan. Saat mengenkripsi folder, tunggu hingga operasi penyimpanannya selesai sebelum menambahkan objek baru ke folder tersebut.

Penggunaan API REST

Saat Anda membuat sebuah objek—yaitu, ketika Anda mengunggah objek baru atau menyalin objek yang sudah ada—Anda dapat menentukan penggunaan enkripsi di sisi server dengan AWS KMS keys (SSE-KMS) untuk mengenkripsi data Anda. Untuk melakukannya, tambahkan header `x-amz-server-side-encryption` ke permintaan. Atur nilai header ke algoritma enkripsi. `aws:kms`. Amazon S3 mengonfirmasi bahwa objek Anda disimpan menggunakan SSE-KMS dengan mengembalikan header respons `x-amz-server-side-encryption`.

Jika Anda menentukan header `x-amz-server-side-encryption` dengan nilai `aws:kms`, Anda juga dapat menggunakan header permintaan berikut ini:

- `x-amz-server-side-encryption-aws-kms-key-id`
- `x-amz-server-side-encryption-context`
- `x-amz-server-side-encryption-bucket-key-enabled`

Topik

- Operasi API REST Amazon S3 yang mendukung SSE-KMS
- Konteks enkripsi (x-amz-server-side-encryption-context)
- AWS KMS ID kunci (x-amz-server-side-encryption-aws-kms-key-id)
- Tombol Bucket S3 () x-amz-server-side-encryption-aws-bucket-key-enabled

Operasi API REST Amazon S3 yang mendukung SSE-KMS

Operasi API REST berikut ini menerima header permintaan `x-amz-server-side-encryption`, `x-amz-server-side-encryption-aws-kms-key-id`, dan `x-amz-server-side-encryption-context`.

- PutObject–Saat Anda mengunggah data dengan menggunakan operasi API `PUT`, Anda dapat menentukan header permintaan ini.
- CopyObject–Saat Anda menyalin objek, Anda memiliki objek sumber dan objek target. Ketika Anda melewati header SSE-KMS dengan `CopyObject` operasi, mereka hanya diterapkan ke objek target. Saat Anda menyalin objek yang ada, terlepas dari apakah objek sumber dienkripsi atau tidak, objek tujuan tidak dienkripsi kecuali Anda secara eksplisit meminta enkripsi sisi server.
- POST Object— Saat Anda menggunakan `POST` operasi untuk mengunggah objek, alih-alih header permintaan, Anda memberikan informasi yang sama di bidang formulir.
- CreateMultipartUpload— Saat Anda mengunggah objek besar dengan menggunakan operasi API unggahan multibagian, Anda dapat menentukan header ini. Anda menentukan header ini dalam `CreateMultipartUpload` permintaan.

Header respons dari operasi API REST berikut mengembalikan header `x-amz-server-side-encryption` saat objek disimpan menggunakan enkripsi di sisi server.

- PutObject
- CopyObject
- POST Object
- CreateMultipartUpload
- UploadPart
- UploadPartCopy
- CompleteMultipartUpload

- GetObject
- HeadObject

> **Important**
>
> - Semua `GET` dan `PUT` permintaan untuk objek yang dilindungi oleh AWS KMS gagal jika Anda tidak membuat permintaan ini dengan menggunakan Secure Sockets Layer (SSL), Transport Layer Security (TLS), atau Signature Version 4.
> - Jika objek Anda menggunakan SSE-KMS, jangan mengirim header permintaan enkripsi untuk `GET` permintaan dan `HEAD` permintaan, atau Anda akan mendapatkan kesalahan HTTP 400. BadRequest

Konteks enkripsi (**`x-amz-server-side-encryption-context`**)

Jika Anda menentukan `x-amz-server-side-encryption:aws:kms`, API Amazon S3 mendukung konteks enkripsi dengan header `x-amz-server-side-encryption-context`. Konteks enkripsi adalah seperangkat pasangan nilai kunci yang berisi informasi kontekstual tambahan terkait data.

Amazon S3 secara otomatis menggunakan objek atau bucket Amazon Resource Name (ARN) sebagai pasangan konteks enkripsi. Jika Anda menggunakan SSE-KMS tanpa mengaktifkan Kunci Bucket S3, Anda menggunakan objek ARN sebagai konteks enkripsi Anda; misalnya, `arn:aws:s3:::`*`object_ARN`*. Namun, jika Anda menggunakan SSE-KMS dan mengaktifkan S3 Bucket Key, Anda menggunakan objek ARN sebagai konteks enkripsi Anda; misalnya, `arn:aws:s3:::`*`bucket_ARN`*.

Anda dapat secara opsional memberikan pasangan konteks enkripsi tambahan dengan menggunakan header `x-amz-server-side-encryption-context`. Namun, karena konteks enkripsi tidak dienkripsi, pastikan itu tidak menyertakan informasi sensitif. Amazon S3 menyimpan pasangan kunci tambahan ini bersama dengan konteks enkripsi default.

Untuk informasi tentang konteks enkripsi di Amazon S3, lihat Konteks enkripsi. Untuk informasi umum tentang konteks enkripsi, lihat AWS Key Management Service Konsep-Konteks enkripsi dalam Panduan Pengembang AWS Key Management Service .

AWS KMS ID kunci (**`x-amz-server-side-encryption-aws-kms-key-id`**)

Anda dapat menggunakan header `x-amz-server-side-encryption-aws-kms-key-id` untuk menentukan ID CMK, yang digunakan untuk melindungi data. Jika Anda menentukan header `x-amz-server-side-encryption:aws:kms` tetapi tidak memberikan header `x-amz-server-side-encryption-aws-kms-key-id`, Amazon S3 menggunakan Kunci yang dikelola AWS (`aws/s3`) untuk melindungi data tersebut. Jika Anda ingin menggunakan CMK, Anda harus memberikan header `x-amz-server-side-encryption-aws-kms-key-id` kunci yang dikelola pelanggan.

> **Important**
>
> Saat Anda menggunakan enkripsi sisi server AWS KMS key untuk Amazon S3, Anda harus memilih kunci KMS enkripsi simetris. Amazon S3 hanya mendukung kunci KMS enkripsi simetris. Untuk informasi selengkapnya terkait kunci ini, lihat Membuat kunci enkripsi simetris KMS dalam Panduan Pengembang AWS Key Management Service .

Tombol Bucket S3 () **`x-amz-server-side-encryption-aws-bucket-key-enabled`**

Anda dapat menggunakan header `x-amz-server-side-encryption-aws-bucket-key-enabled` permintaan untuk mengaktifkan atau menonaktifkan Kunci Bucket S3 di tingkat objek. Kunci Bucket S3 mengurangi biaya AWS KMS permintaan Anda dengan mengurangi lalu lintas permintaan dari Amazon S3 ke. AWS KMS Untuk informasi selengkapnya, lihat Mengurangi biaya SSE-KMS dengan Kunci Bucket Amazon S3.

Jika Anda menentukan header `x-amz-server-side-encryption:aws:kms` tapi tidak memberikan header `x-amz-server-side-encryption-aws-bucket-key-enabled`, objek Anda menggunakan pengaturan Kunci Bucket S3 pada bucket tujuan untuk mengenkripsi objek Anda. Untuk informasi selengkapnya, lihat Mengonfigurasi Kunci Bucket S3 pada tingkat objek.

Menggunakan AWS CLI

Untuk menggunakan contoh AWS CLI perintah berikut, ganti *`user input placeholders`* dengan informasi Anda sendiri.

Saat Anda mengunggah objek baru atau menyalin objek yang ada, Anda dapat menentukan penggunaan enkripsi sisi server dengan AWS KMS kunci untuk mengenkripsi data Anda. Untuk melakukannya, tambahkan header `--server-side-encryption aws:kms` ke permintaan. Gunakan tombol `--ssekms-key-id` *`example-key-id`* untuk menambahkan AWS KMS kunci

terkelola pelanggan yang Anda buat. Jika Anda menentukan`--server-side-encryption aws:kms`, tetapi tidak memberikan ID AWS KMS kunci, Amazon S3 akan menggunakan kunci AWS terkelola.

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key example-object-key --server-
side-encryption aws:kms --ssekms-key-id example-key-id --body filepath
```

Anda juga dapat mengaktifkan atau menonaktifkan Tombol Bucket Amazon S3 pada operasi PUT atau COPY Anda dengan menambahkan `--bucket-key-enabled` atau `--no-bucket-key-enabled`. Amazon S3 Bucket Keys dapat mengurangi biaya AWS KMS permintaan Anda dengan mengurangi lalu lintas permintaan dari Amazon S3 ke. AWS KMS Untuk informasi selengkapnya, lihat Mengurangi biaya SSE-KMS menggunakan Kunci Bucket Amazon S3.

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key example-object-key --server-
side-encryption aws:kms --bucket-key-enabled --body filepath
```

Anda dapat mengenkripsi objek yang tidak terenkripsi untuk menggunakan SSE-KMS dengan menyalin objek kembali ke tempatnya.

```
aws s3api copy-object --bucket amzn-s3-demo-bucket --key example-object-key --
body filepath --bucket amzn-s3-demo-bucket --key example-object-key --sse aws:kms --
sse-kms-key-id example-key-id --body filepath
```

Menggunakan AWS SDKs

Saat menggunakan AWS SDKs, Anda dapat meminta Amazon S3 untuk digunakan AWS KMS keys untuk enkripsi sisi server. Contoh berikut menunjukkan bagaimana menggunakan SSE-KMS dengan AWS SDKs untuk Java dan .NET. Untuk informasi tentang lainnya SDKs, lihat Contoh kode dan pustaka di Pusat AWS Pengembang.

> **Important**
>
> Saat Anda menggunakan enkripsi sisi server AWS KMS key untuk Amazon S3, Anda harus memilih kunci KMS enkripsi simetris. Amazon S3 hanya mendukung kunci KMS enkripsi simetris. Untuk informasi selengkapnya terkait kunci ini, lihat Membuat kunci enkripsi simetris KMS dalam Panduan Pengembang AWS Key Management Service .

Operasi **`CopyObject`**

Saat menyalin objek, Anda menambahkan properti permintaan yang sama (`ServerSideEncryptionMethod` dan `ServerSideEncryptionKeyManagementServiceKeyId`) untuk meminta Amazon S3 menggunakan AWS KMS key. Untuk informasi selengkapnya tentang menyalin objek, lihat Menyalin, memindahkan, dan mengganti nama objek.

Operasi **`PUT`**

Java

Saat mengunggah objek menggunakan AWS SDK untuk Java, Anda dapat meminta Amazon S3 untuk menggunakan AWS KMS key dengan menambahkan properti seperti `SSEAwsKeyManagementParams` yang ditunjukkan dalam permintaan berikut:

```
PutObjectRequest putRequest = new PutObjectRequest(bucketName,
   keyName, file).withSSEAwsKeyManagementParams(new SSEAwsKeyManagementParams());
```

Dalam hal ini, Amazon S3 menggunakan Kunci yang dikelola AWS (`aws/s3`. Untuk informasi selengkapnya, lihat Menggunakan enkripsi sisi server dengan AWS KMS kunci (SSE-KMS). Anda dapat secara opsional membuat kunci KMS enkripsi simetris dan menentukannya dalam permintaan, seperti yang ditunjukkan pada contoh berikut:

```
PutObjectRequest putRequest = new PutObjectRequest(bucketName,
   keyName, file).withSSEAwsKeyManagementParams(new
 SSEAwsKeyManagementParams(keyID));
```

Untuk informasi selengkapnya tentang membuat kunci terkelola pelanggan, lihat Memprogram AWS KMS API di Panduan AWS Key Management Service Pengembang.

Untuk contoh kode kerja mengunggah sebuah objek, lihat topik berikut ini. Untuk menggunakan contoh ini, Anda harus memperbarui contoh kode dan memberikan informasi enkripsi seperti yang ditunjukkan dalam fragmen kode sebelumnya.

- Untuk mengunggah objek dalam operasi tunggal, lihat Mengunggah Objek.
- Untuk unggahan multibagian yang menggunakan operasi API unggahan multibagian tingkat tinggi atau tingkat rendah, lihat. Pengunggahan objek menggunakan unggahan multibagian

.NET

Saat mengunggah objek menggunakan AWS SDK untuk .NET, Anda dapat meminta Amazon S3 untuk menggunakan AWS KMS key dengan menambahkan properti seperti `ServerSideEncryptionMethod` yang ditunjukkan dalam permintaan berikut:

```
PutObjectRequest putRequest = new PutObjectRequest
 {
     BucketName = amzn-s3-demo-bucket,
     Key = keyName,
     // other properties
     ServerSideEncryptionMethod = ServerSideEncryptionMethod.AWSKMS
 };
```

Dalam hal ini, Amazon S3 menggunakan. Kunci yang dikelola AWS Untuk informasi selengkapnya, lihat Menggunakan enkripsi sisi server dengan AWS KMS kunci (SSE-KMS). Anda dapat secara opsional membuat kunci terkelola pelanggan enkripsi simetris Anda sendiri dan menentukannya dalam permintaan, seperti yang ditunjukkan pada contoh berikut:

```
PutObjectRequest putRequest1 = new PutObjectRequest
{
  BucketName = amzn-s3-demo-bucket,
  Key = keyName,
  // other properties
  ServerSideEncryptionMethod = ServerSideEncryptionMethod.AWSKMS,
  ServerSideEncryptionKeyManagementServiceKeyId = keyId
};
```

Untuk informasi selengkapnya tentang membuat kunci terkelola pelanggan, lihat Memprogram AWS KMS API di Panduan AWS Key Management Service Pengembang.

Untuk contoh kode kerja mengunggah sebuah objek, lihat topik berikut ini. Untuk menggunakan contoh ini, Anda harus memperbarui contoh kode dan memberikan informasi enkripsi seperti yang ditunjukkan dalam fragmen kode sebelumnya.

- Untuk mengunggah objek dalam operasi tunggal, lihat Mengunggah Objek.
- Untuk unggahan multibagian yang menggunakan operasi API unggahan multibagian tingkat tinggi atau tingkat rendah, lihat. Pengunggahan objek menggunakan unggahan multibagian

Ditandatangani URLs

Java

Saat membuat URL presigned untuk objek yang dienkripsi dengan AWS KMS key, Anda harus secara eksplisit menentukan Signature Versi 4, seperti yang ditunjukkan pada contoh berikut:

```
ClientConfiguration clientConfiguration = new ClientConfiguration();
clientConfiguration.setSignerOverride("AWSS3V4SignerType");
AmazonS3Client s3client = new AmazonS3Client(
        new ProfileCredentialsProvider(), clientConfiguration);
...
```

Untuk contoh kode, lihat Berbagi objek dengan presigned URLs.

.NET

Saat membuat URL presigned untuk objek yang dienkripsi dengan AWS KMS key, Anda harus secara eksplisit menentukan Signature Versi 4, seperti yang ditunjukkan pada contoh berikut:

```
AWSConfigs.S3Config.UseSignatureVersion4 = true;
```

Untuk contoh kode, lihat Berbagi objek dengan presigned URLs.

Mengurangi biaya SSE-KMS dengan Kunci Bucket Amazon S3

Amazon S3 Bucket Keys mengurangi biaya enkripsi sisi server Amazon S3 AWS Key Management Service dengan AWS KMS kunci () (SSE-KMS). Menggunakan kunci tingkat ember untuk SSE-KMS dapat mengurangi biaya AWS KMS permintaan hingga 99 persen dengan mengurangi lalu lintas permintaan dari Amazon S3 ke. AWS KMS Dengan beberapa klik di AWS Management Console, dan tanpa perubahan apa pun pada aplikasi klien Anda, Anda dapat mengonfigurasi bucket Anda untuk menggunakan Kunci Bucket S3 untuk enkripsi SSE-KMS pada objek baru.

> **Note**
>
> Kunci Bucket S3 tidak didukung untuk enkripsi sisi server dua lapis dengan kunci AWS Key Management Service () (AWS KMS DSSE-KMS).

Kunci Bucket S3 untuk SSE-KMS

Beban kerja yang mengakses jutaan atau miliaran objek yang dienkripsi dengan SSE-KMS dapat menghasilkan volume permintaan yang besar. AWS KMSSaat Anda menggunakan SSE-KMS untuk melindungi data Anda tanpa Kunci Bucket S3, Amazon S3 menggunakan kunci data individual untuk setiap AWS KMS objek. Dalam hal ini, Amazon S3 melakukan panggilan ke AWS KMS setiap kali permintaan dibuat terhadap objek yang dienkripsi KMS. Untuk informasi tentang cara kerja SSE-KMS, lihat Menggunakan enkripsi sisi server dengan AWS KMS kunci (SSE-KMS).

Saat Anda mengonfigurasi bucket untuk menggunakan Kunci Bucket S3 untuk SSE-KMS, buat kunci tingkat AWS ember berumur pendek dari, lalu simpan sementara di S3. AWS KMS Kunci tingkat bucket ini akan membuat kunci data untuk objek baru selama siklus hidupnya. Kunci Bucket S3 digunakan untuk jangka waktu terbatas dalam Amazon S3, mengurangi kebutuhan S3 untuk membuat permintaan AWS KMS untuk menyelesaikan operasi enkripsi. Ini mengurangi lalu lintas dari S3 menjadi AWS KMS, memungkinkan Anda mengakses objek yang AWS KMS dienkripsi di Amazon S3 dengan biaya yang lebih murah dari biaya sebelumnya.

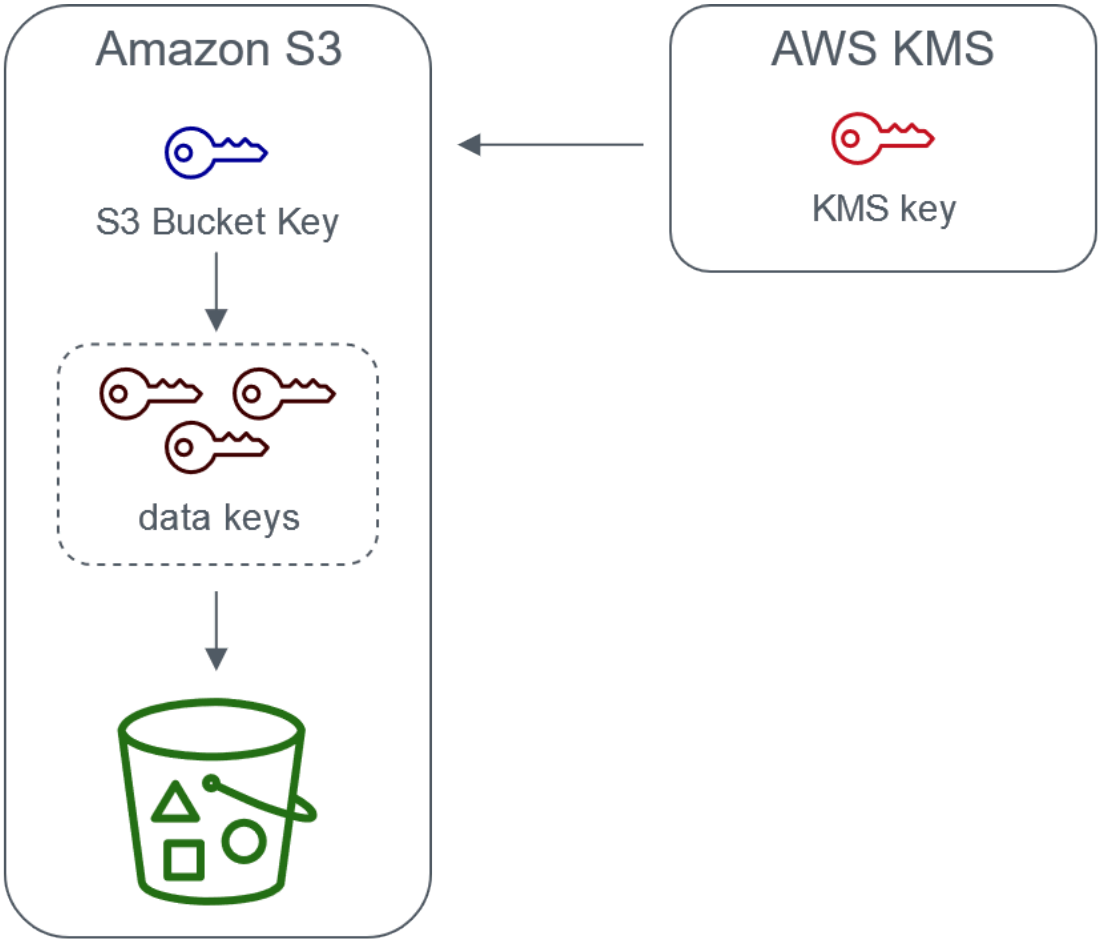
Kunci tingkat ember unik diambil setidaknya sekali per pemohon untuk memastikan bahwa akses pemohon ke kunci ditangkap dalam suatu peristiwa. AWS KMS CloudTrail Amazon S3 memperlakukan penelepon sebagai pemohon yang berbeda ketika mereka menggunakan peran atau akun yang berbeda, atau peran yang sama dengan kebijakan pelingkupan yang berbeda. AWS KMS penghematan permintaan mencerminkan jumlah pemohon, pola permintaan, dan usia relatif dari objek yang diminta. Misalnya, jumlah pemohon yang lebih sedikit, meminta beberapa objek dalam jendela waktu terbatas, dan dienkripsi dengan kunci tingkat ember yang sama, menghasilkan penghematan yang lebih besar.

> **Note**
>
> Menggunakan S3 Bucket Keys memungkinkan Anda menghemat biaya AWS KMS permintaan dengan mengurangi permintaan AWS KMS untuk`Encrypt`,`GenerateDataKey`, dan `Decrypt` operasi melalui penggunaan kunci tingkat ember. Secara desain, permintaan berikutnya yang memanfaatkan kunci tingkat ember ini tidak menghasilkan permintaan AWS KMS API atau memvalidasi akses terhadap kebijakan kunci. AWS KMS

Bila Anda mengonfigurasi Kunci Bucket S3, objek yang sudah ada di bucket tidak menggunakan kunci Bucket S3. Untuk mengonfigurasi Kunci Bucket S3 untuk objek yang sudah ada, Anda dapat

menggunakan operasi `CopyObject`. Untuk informasi selengkapnya, lihat Mengonfigurasi Kunci Bucket S3 pada tingkat objek.

Amazon S3 hanya akan berbagi Kunci Bucket S3 untuk objek dienkripsi oleh AWS KMS key yang sama. Kunci Bucket S3 kompatibel dengan kunci KMS yang dibuat oleh AWS KMS, bahan kunci impor, dan bahan kunci yang didukung oleh toko kunci khusus.



Server-side encryption with AWS Key Management service using an S3 Bucket Key

Mengonfigurasi Kunci Bucket S3

Anda dapat mengonfigurasi bucket untuk menggunakan Kunci Bucket S3 untuk SSE-KMS pada objek baru melalui konsol Amazon S3,, AWS SDKs atau REST API. AWS CLI Dengan Kunci Bucket S3 diaktifkan di bucket Anda, objek yang diunggah dengan kunci SSE-KMS tertentu yang berbeda akan menggunakan Kunci Bucket S3 miliknya sendiri. Terlepas dari pengaturan Kunci Bucket S3 Anda, Anda dapat menyertakan header `x-amz-server-side-encryption-bucket-key-enabled` dengan nilai `true` atau `false` atau dalam permintaan Anda, untuk mengganti pengaturan bucket.

Sebelum mengonfigurasi bucket untuk menggunakan Kunci Bucket S3, tinjau Perubahan yang perlu diperhatikan sebelum mengaktifkan Kunci Bucket S3.

Mengonfigurasi Kunci Bucket S3 menggunakan konsol Amazon S3

Ketika Anda membuat bucket baru, Anda dapat mengonfigurasi bucket Anda untuk menggunakan Kunci Bucket S3 untuk SSE-KMS pada objek baru. Anda juga dapat mengonfigurasi bucket yang ada untuk menggunakan Kunci Bucket S3 untuk SSE-KMS pada objek baru dengan memperbarui properti bucket Anda.

Untuk informasi selengkapnya, lihat Mengonfigurasi bucket Anda untuk menggunakan Kunci Bucket S3 dengan SSE-KMS untuk objek baru.

REST API, AWS CLI, dan dukungan AWS SDK untuk S3 Bucket Keys

Anda dapat menggunakan REST API, AWS CLI, atau AWS SDK untuk mengonfigurasi bucket agar menggunakan Kunci Bucket S3 untuk SSE-KMS pada objek baru. Anda juga dapat mengaktifkan Kunci Bucket S3 pada tingkat objek.

Untuk informasi selengkapnya, lihat berikut ini:

- Mengonfigurasi Kunci Bucket S3 pada tingkat objek
- Mengonfigurasi bucket Anda untuk menggunakan Kunci Bucket S3 dengan SSE-KMS untuk objek baru

Operasi API berikut mendukung Kunci Bucket S3 untuk SSE-KMS:

- PutBucketEncryption
  - `ServerSideEncryptionRule` menerima parameter `BucketKeyEnabled` untuk mengaktifkan dan menonaktifkan Kunci Bucket S3.
- GetBucketEncryption
  - `ServerSideEncryptionRule` mengembalikan pengaturan untuk `BucketKeyEnabled`.
- PutObject, CopyObject, CreateMultipartUpload, dan POST Object
  - Anda dapat menggunakan header permintaan `x-amz-server-side-encryption-bucket-key-enabled` untuk mengaktifkan atau menonaktifkan Kunci Bucket S3 pada tingkat objek.
- HeadObject, GetObject, UploadPartCopy, UploadPart, dan CompleteMultipartUpload
  - Respons header `x-amz-server-side-encryption-bucket-key-enabled` menunjukkan jika Kunci Bucket S3 diaktifkan atau dinonaktifkan untuk sebuah objek.

Bekerja dengan AWS CloudFormation

Di AWS CloudFormation, `AWS::S3::Bucket` sumber daya menyertakan properti enkripsi yang disebut `BucketKeyEnabled` yang dapat Anda gunakan untuk mengaktifkan atau menonaktifkan Kunci Bucket S3.

Untuk informasi selengkapnya, lihat Menggunakan AWS CloudFormation.

Perubahan yang perlu diperhatikan sebelum mengaktifkan Kunci Bucket S3

Sebelum mengaktifkan Kunci Bucket S3, perhatikan perubahan terkait berikut ini:

IAM atau kebijakan AWS KMS utama

Jika kebijakan atau kebijakan AWS KMS utama AWS Identity and Access Management (IAM) yang ada menggunakan objek Amazon Resource Name (ARN) sebagai konteks enkripsi untuk mempersempit atau membatasi akses ke kunci KMS, kebijakan ini tidak akan berfungsi dengan Kunci Bucket S3. Kunci Bucket S3 menggunakan bucket ARN sebagai konteks enkripsi. Sebelum mengaktifkan Kunci Bucket S3, perbarui kebijakan IAM atau kebijakan AWS KMS utama untuk menggunakan ARN bucket sebagai konteks enkripsi.

Untuk informasi selengkapnya tentang konteks enkripsi dan Kunci Bucket S3, lihat Konteks enkripsi.

CloudTrail acara untuk AWS KMS

Setelah Anda mengaktifkan Kunci Bucket S3, AWS KMS CloudTrail peristiwa Anda mencatat ARN bucket Anda alih-alih ARN objek Anda. Selain itu, Anda melihat lebih sedikit CloudTrail peristiwa KMS untuk objek SSE-KMS di log Anda. Karena materi utama dibatasi waktu di Amazon S3, lebih sedikit permintaan yang dibuat. AWS KMS

Menggunakan kunci Bucket S3 dengan replikasi

Anda dapat menggunakan Kunci Bucket S3 dengan Replikasi Wilayah yang Sama (SRR) dan Replikasi Lintas-Wilayah (CRR).

Ketika Amazon S3 mereplikasi objek terenkripsi, umumnya mempertahankan pengaturan enkripsi objek replika di bucket tujuan. Namun, jika objek sumber tidak dienkripsi dan bucket tujuan Anda menggunakan enkripsi default atau Kunci Bucket S3, Amazon S3 mengenkripsi objek dengan konfigurasi bucket tujuan.

Contoh berikut menggambarkan bagaimana Kunci Bucket S3 bekerja dengan replikasi. Untuk informasi selengkapnya, lihat Mereplikasi objek terenkripsi (SSE-S3, SSE-KMS, DSSE-KMS, SSE-C).

Example Contoh 1–Sumber objek menggunakan Kunci Bucket S3, bucket tujuan menggunakan enkripsi default

Jika objek sumber Anda menggunakan Kunci Bucket S3 namun bucket tujuan Anda menggunakan enkripsi default dengan SSE-KMS, objek replika akan mempertahankan pengaturan enkripsi Kunci Bucket S3 di bucket tujuan. Bucket tujuan masih menggunakan enkripsi default dengan SSE-KMS.

Example Contoh 2–Objek sumber tidak dienkripsi, bucket tujuan menggunakan Kunci Bucket S3 dengan SSE-KMS

Jika objek sumber Anda tidak dienkripsi dan bucket tujuan menggunakan Kunci Bucket S3 dengan SSE-KMS, objek replika dienkripsi menggunakan Kunci Bucket S3 dengan SSE-KMS di bucket tujuan. Ini menghasilkan objek sumber `ETag` yang berbeda dari objek replika `ETag`. Anda harus memperbarui aplikasi yang menggunakan `ETag` untuk mengakomodasi perbedaan ini.

Bekerja dengan Kunci Bucket S3

Untuk informasi lebih lanjut tentang mengaktifkan dan bekerja dengan Kunci Bucket S3, lihat bagian berikut:

- Mengonfigurasi bucket Anda untuk menggunakan Kunci Bucket S3 dengan SSE-KMS untuk objek baru
- Mengonfigurasi Kunci Bucket S3 pada tingkat objek
- Melihat pengaturan untuk Kunci Bucket S3

Mengonfigurasi bucket Anda untuk menggunakan Kunci Bucket S3 dengan SSE-KMS untuk objek baru

Saat mengonfigurasi enkripsi sisi server dengan kunci AWS Key Management Service (AWS KMS) (SSE-KMS), Anda dapat mengonfigurasi bucket untuk menggunakan Kunci Bucket S3 untuk SSE-KMS pada objek baru. S3 Bucket Keys mengurangi lalu lintas permintaan dari Amazon S3 AWS KMS ke dan mengurangi biaya SSE-KMS. Untuk informasi selengkapnya, lihat Mengurangi biaya SSE-KMS dengan Kunci Bucket Amazon S3.

Anda dapat mengonfigurasi bucket untuk menggunakan Kunci Bucket S3 untuk SSE-KMS pada objek baru menggunakan konsol Amazon S3, REST API, (), atau. AWS SDKs AWS Command Line Interface AWS CLI AWS CloudFormation Jika Anda ingin mengaktifkan atau nonaktifkan Kunci

Bucket S3 untuk objek yang ada, Anda dapat menggunakan sebuah `CopyObject` operasi. Untuk informasi selengkapnya, lihat Mengonfigurasi Kunci Bucket S3 pada tingkat objek dan Menggunakan Operasi Batch untuk mengaktifkan Kunci Bucket S3 untuk SSE-KMS.

Saat Kunci Bucket S3 diaktifkan untuk bucket sumber atau tujuan, konteks enkripsi akan menjadi Amazon Resource Name (ARN) bucket, bukan ARN objek (misalnya, `arn:aws:s3:::bucket_ARN`). Anda harus memperbarui kebijakan IAM Anda untuk menggunakan ARN bucket untuk konteks enkripsi. Untuk informasi selengkapnya, lihat Replikasi dan Kunci Bucket S3.

Contoh berikut menggambarkan bagaimana Kunci Bucket S3 bekerja dengan replikasi. Untuk informasi selengkapnya, lihat Mereplikasi objek terenkripsi (SSE-S3, SSE-KMS, DSSE-KMS, SSE-C).

Prasyarat

Sebelum mengonfigurasi bucket untuk menggunakan Kunci Bucket S3, tinjau Perubahan yang perlu diperhatikan sebelum mengaktifkan Kunci Bucket S3.

Menggunakan konsol S3

Di konsol S3, Anda dapat mengaktifkan atau nonaktifkan Kunci Bucket S3 untuk bucket baru atau yang sudah ada. Objek di konsol S3 mewarisi pengaturan Kunci Bucket S3 mereka dari konfigurasi bucket. Saat Anda mengaktifkan Kunci Bucket S3 untuk bucket Anda, objek baru yang Anda unggah ke bucket menggunakan Kunci Bucket S3 untuk SSE-KMS.

Mengunggah, menyalin, atau memodifikasi objek dalam bucket yang memiliki Kunci Bucket S3 diaktifkan

Jika Anda unggah, memodifikasi, atau menyalin objek dalam bucket yang mengaktifkan Kunci Bucket S3, pengaturan Kunci Bucket S3 untuk objek tersebut dapat diperbarui agar sesuai dengan konfigurasi bucket.

Jika sebuah objek telah mengaktifkan Kunci Bucket S3, pengaturan Kunci Bucket S3 untuk objek tersebut tidak berubah saat Anda menyalin atau memodifikasi objek. Namun, jika Anda mengubah atau menyalin objek yang tidak memiliki Kunci Bucket S3 yang diaktifkan, dan bucket tujuan memiliki konfigurasi Kunci Bucket S3, objek mewarisi pengaturan Kunci Bucket S3 bucket tujuan. Misalnya, jika objek sumber Anda tidak mengaktifkan Kunci Bucket S3 namun bucket tujuan mengaktifkan Kunci Bucket S3, Kunci Bucket S3 diaktifkan untuk objek tersebut.

Untuk mengaktifkan Kunci Bucket S3 saat Anda membuat bucket baru

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Pilih Buat bucket.
4. Masukkan nama bucket Anda, dan pilih Wilayah AWS.
5. Di bawah Enkripsi default, untuk Jenis kunci enkripsi, pilih kunci AWS Key Management Service (SSE-KMS).
6. Di bagian bawah AWS KMS kunci, lakukan salah satu langkah berikut untuk memilih kunci KMS Anda:

   - Untuk memilih dari daftar kunci KMS yang tersedia, pilih Pilih dari Anda AWS KMS keys, lalu pilih kunci KMS Anda dari daftar kunci yang tersedia.

     Kunci Kunci yang dikelola AWS (`aws/s3`) dan kunci terkelola pelanggan Anda muncul dalam daftar ini. Untuk informasi selengkapnya tentang kunci yang dikelola pelanggan, lihat Kunci dan AWS kunci pelanggan di Panduan AWS Key Management Service Pengembang.
   - Untuk memasukkan ARN kunci KMS, pilih Masukkan AWS KMS key ARN, dan masukkan ARN kunci KMS Anda di bidang yang muncul.
   - Untuk membuat kunci terkelola pelanggan baru di AWS KMS konsol, pilih Buat kunci KMS.

     Untuk informasi selengkapnya tentang membuat AWS KMS key, lihat Membuat Kunci di Panduan AWS Key Management Service Pengembang.
7. Pada Kunci Bucket, pilih Aktifkan.
8. Pilih Buat bucket.

   Amazon S3 menciptakan bucket Anda dengan Kunci Bucket S3 yang diaktifkan. Objek baru yang Anda unggah ke bucket akan menggunakan Kunci Bucket S3.

   Untuk menonaktifkan Kunci Bucket S3, ikuti langkah-langkah sebelumnya, dan pilih Nonaktifkan.

Cara mengaktifkan Kunci Bucket S3 untuk bucket yang ada

1. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.

3. Di dalam daftar Bucket, pilih bucket yang ingin Anda aktifkan Kunci Bucket S3-nya.
4. Pilih tab Properti.
5. Di bagian bawah Enkripsi default, pilih Edit.
6. Di bawah Enkripsi default, untuk Jenis kunci enkripsi, pilih kunci AWS Key Management Service (SSE-KMS).
7. Di bagian bawah AWS KMS kunci, lakukan salah satu langkah berikut untuk memilih kunci KMS Anda:

   - Untuk memilih dari daftar kunci KMS yang tersedia, pilih Pilih dari Anda AWS KMS keys, lalu pilih kunci KMS Anda dari daftar kunci yang tersedia.

     Kunci Kunci yang dikelola AWS (`aws/s3`) dan kunci terkelola pelanggan Anda muncul dalam daftar ini. Untuk informasi selengkapnya tentang kunci yang dikelola pelanggan, lihat Kunci dan AWS kunci pelanggan di Panduan AWS Key Management Service Pengembang.
   - Untuk memasukkan ARN kunci KMS, pilih Masukkan AWS KMS key ARN, dan masukkan ARN kunci KMS Anda di bidang yang muncul.
   - Untuk membuat kunci terkelola pelanggan baru di AWS KMS konsol, pilih Buat kunci KMS.

     Untuk informasi selengkapnya tentang membuat AWS KMS key, lihat Membuat Kunci di Panduan AWS Key Management Service Pengembang.
8. Pada Kunci Bucket, pilih Aktifkan.
9. Pilih Simpan perubahan.

   Amazon S3 mengaktifkan Kunci Bucket S3 untuk objek baru yang ditambahkan ke bucket Anda. Objek yang ada tidak menggunakan Kunci Bucket S3. Untuk mengonfigurasi Kunci Bucket S3 untuk objek yang sudah ada, Anda dapat menggunakan operasi `CopyObject`. Untuk informasi selengkapnya, lihat Mengonfigurasi Kunci Bucket S3 pada tingkat objek.

   Untuk menonaktifkan Kunci Bucket S3, ikuti langkah-langkah sebelumnya, dan pilih Nonaktifkan.

Penggunaan API REST

Anda dapat menggunakan PutBucketEncryptionuntuk mengaktifkan atau menonaktifkan Kunci Bucket S3 untuk bucket Anda. Untuk mengonfigurasi Kunci Bucket S3 dengan`PutBucketEncryption`, gunakan tipe ServerSideEncryptionRuledata, yang mencakup enkripsi default dengan SSE-KMS. Anda juga dapat menggunakan CMK dengan menentukan ID kunci KMS untuk kunci yang dikelola pelanggan.

Untuk informasi selengkapnya dan contoh sintaks, lihat PutBucketEncryption.

Menggunakan AWS SDK for Java

Contoh berikut ini mengaktifkan enkripsi bucket default dengan SSE-KMS dan Kunci Bucket S3 dengan menggunakan AWS SDK untuk Java.

Java

```
AmazonS3 s3client = AmazonS3ClientBuilder.standard()
    .withRegion(Regions.DEFAULT_REGION)
    .build();

ServerSideEncryptionByDefault serverSideEncryptionByDefault = new
 ServerSideEncryptionByDefault()
    .withSSEAlgorithm(SSEAlgorithm.KMS);
ServerSideEncryptionRule rule = new ServerSideEncryptionRule()
    .withApplyServerSideEncryptionByDefault(serverSideEncryptionByDefault)
    .withBucketKeyEnabled(true);
ServerSideEncryptionConfiguration serverSideEncryptionConfiguration =
    new ServerSideEncryptionConfiguration().withRules(Collections.singleton(rule));

SetBucketEncryptionRequest setBucketEncryptionRequest = new
 SetBucketEncryptionRequest()
    .withServerSideEncryptionConfiguration(serverSideEncryptionConfiguration)
    .withBucketName(bucketName);

s3client.setBucketEncryption(setBucketEncryptionRequest);
```

Menggunakan AWS CLI

Contoh berikut ini mengaktifkan enkripsi bucket default dengan SSE-KMS dan Kunci Bucket S3 dengan menggunakan AWS CLI. Ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3api put-bucket-encryption --bucket amzn-s3-demo-bucket --server-side-encryption-
configuration '{
        "Rules": [
            {
                "ApplyServerSideEncryptionByDefault": {
                    "SSEAlgorithm": "aws:kms",
                    "KMSMasterKeyID": "KMS-Key-ARN"
```

```
            },
            "BucketKeyEnabled": true
        }
    ]
}'
```

Menggunakan AWS CloudFormation

Untuk informasi selengkapnya tentang mengonfigurasi Kunci Bucket S3 dengan AWS CloudFormation, lihat AWS::S3::Bucket ServerSideEncryptionRuledi AWS CloudFormation Panduan Pengguna.

Mengonfigurasi Kunci Bucket S3 pada tingkat objek

Saat Anda melakukan operasi PUT atau COPY menggunakan REST API AWS SDKs, atau AWS CLI, Anda dapat mengaktifkan atau menonaktifkan Kunci Bucket S3 di tingkat objek dengan menambahkan header `x-amz-server-side-encryption-bucket-key-enabled` permintaan dengan `false` nilai `true` atau. S3 Bucket Keys mengurangi biaya enkripsi sisi server menggunakan AWS Key Management Service (AWS KMS) (SSE-KMS) dengan mengurangi lalu lintas permintaan dari Amazon S3 ke. AWS KMS Untuk informasi selengkapnya, lihat Mengurangi biaya SSE-KMS dengan Kunci Bucket Amazon S3.

Saat Anda mengonfigurasi Kunci Bucket S3 untuk suatu objek menggunakan operasi PUT atau COPY, Amazon S3 hanya memperbarui pengaturan untuk objek tersebut. Pengaturan Kunci Bucket S3 untuk bucket tujuan tidak berubah. Jika Anda mengirimkan permintaan PUT atau COPY untuk objek terenkripsi KMS ke dalam bucket dengan S3 Bucket Keys diaktifkan, operasi tingkat objek Anda akan secara otomatis menggunakan S3 Bucket Keys, kecuali jika Anda menonaktifkan kunci di header permintaan. Jika Anda tidak menentukan Kunci Bucket S3 untuk objek Anda, Amazon S3 menerapkan pengaturan Kunci Bucket S3 untuk bucket tujuan ke objek.

Prasyarat:

Sebelum mengonfigurasi bucket untuk menggunakan Kunci Bucket S3, tinjau Perubahan yang perlu diperhatikan sebelum mengaktifkan Kunci Bucket S3.

Topik

- Operasi Batch Amazon S3
- Penggunaan API REST
- Menggunakan AWS SDK for Java PutObject ()
- Menggunakan AWS CLI (PutObject)

Operasi Batch Amazon S3

Untuk mengenkripsi objek Amazon S3 yang tidak terenkripsi, Anda dapat menggunakan Operasi Batch Amazon S3. Anda harus menyediakan Operasi Batch S3 dengan daftar objek untuk dioperasikan, dan Operasi Batch akan memanggil masing-masing API untuk melakukan operasi tertentu.

Anda juga dapat menggunakan Operasi Salin Operasi Batch S3 untuk menyalin objek tidak terenkripsi yang ada, dan menuliskannya kembali ke bucket yang sama sebagai objek terenkripsi. Satu tugas Operasi Batch dapat melakukan operasi tertentu pada miliaran objek yang berisi data sebesar eksabita. Untuk informasi selengkapnya, lihat Melakukan operasi objek secara massal dengan Operasi Batch dan Mengenkripsi objek dengan Operasi Batch Amazon S3.

Penggunaan API REST

Saat Anda menggunakan SSE-KMS, Anda dapat mengaktifkan Kunci Bucket S3 untuk sebuah objek menggunakan operasi API berikut ini:

- PutObject— Saat Anda mengunggah objek, Anda dapat menentukan header `x-amz-server-side-encryption-bucket-key-enabled` permintaan untuk mengaktifkan atau menonaktifkan Kunci Bucket S3 di tingkat objek.
- CopyObject— Saat Anda menyalin objek dan mengkonfigurasi SSE-KMS, Anda dapat menentukan header `x-amz-server-side-encryption-bucket-key-enabled` permintaan untuk mengaktifkan atau menonaktifkan Kunci Bucket S3 untuk objek Anda.
- Objek POST – Saat Anda menggunakan `POST` operasi untuk mengunggah objek dan mengonfigurasi SSE-KMS, Anda dapat menggunakan bidang formulir `x-amz-server-side-encryption-bucket-key-enabled` untuk mengaktifkan atau nonaktifkan Kunci Bucket S3 untuk objek Anda.
- CreateMultipartUpload— Saat Anda mengunggah objek besar dengan menggunakan operasi `CreateMultipartUpload` API dan mengkonfigurasi SSE-KMS, Anda dapat menggunakan header `x-amz-server-side-encryption-bucket-key-enabled` permintaan untuk mengaktifkan atau menonaktifkan Kunci Bucket S3 untuk objek Anda.

Untuk mengaktifkan Kunci Bucket S3 pada tingkat objek, termasuk header permintaan `x-amz-server-side-encryption-bucket-key-enabled`. Untuk informasi selengkapnya tentang SSE-KMS dan API REST, lihat Penggunaan API REST.

Menggunakan AWS SDK for Java PutObject ()

Anda dapat menggunakan contoh berikut untuk mengonfigurasi Kunci Bucket S3 di tingkat objek menggunakan AWS SDK untuk Java.

Java

```
AmazonS3 s3client = AmazonS3ClientBuilder.standard()
    .withRegion(Regions.DEFAULT_REGION)
    .build();

String bucketName = "amzn-s3-demo-bucket1";
String keyName = "key name for object";
String contents = "file contents";

PutObjectRequest putObjectRequest = new PutObjectRequest(bucketName, keyName,
 contents)
    .withBucketKeyEnabled(true);

s3client.putObject(putObjectRequest);
```

Menggunakan AWS CLI (PutObject)

Anda dapat menggunakan AWS CLI contoh berikut untuk mengonfigurasi Kunci Bucket S3 di tingkat objek sebagai bagian dari `PutObject` permintaan.

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key object key name --server-side-
encryption aws:kms --bucket-key-enabled --body filepath
```

Melihat pengaturan untuk Kunci Bucket S3

Anda dapat melihat setelan untuk Kunci Bucket S3 di tingkat bucket atau objek dengan menggunakan konsol Amazon S3, REST API,AWS CLI() AWS Command Line Interface , atau. AWS SDKs

S3 Bucket Keys mengurangi lalu lintas permintaan dari Amazon S3 AWS KMS ke dan mengurangi biaya penggunaan enkripsi sisi server (SSE-KMS). AWS Key Management Service Untuk informasi selengkapnya, lihat Mengurangi biaya SSE-KMS dengan Kunci Bucket Amazon S3.

Untuk melihat pengaturan Kunci Bucket S3 untuk bucket atau objek yang mewarisi pengaturan Kunci Bucket S3 dari konfigurasi bucket, Anda memerlukan izin untuk melakukan tindakan `s3:GetEncryptionConfiguration`. Untuk informasi selengkapnya, lihat GetBucketEncryption dalam Referensi API Amazon Simple Storage Service.

Menggunakan konsol S3

Di konsol S3, Anda dapat melihat pengaturan Kunci Bucket S3 untuk bucket atau objek Anda. Pengaturan Kunci Bucket S3 diwarisi dari konfigurasi bucket kecuali objek sumber sudah memiliki Kunci Bucket S3 dikonfigurasi.

Objek dan folder dalam bucket yang sama dapat memiliki pengaturan Kunci Bucket S3 yang berbeda. Misalnya, jika Anda unggah objek menggunakan API REST dan mengaktifkan Kunci Bucket S3 untuk objek, objek mempertahankan pengaturan Kunci Bucket S3 di bucket tujuan, bahkan jika Kunci Bucket S3 dinonaktifkan di bucket tujuan. Sebagai contoh lain, jika Anda mengaktifkan kunci Bucket S3 untuk bucket yang ada, objek yang sudah ada di dalam bucket tidak menggunakan Kunci Bucket S3. Namun, objek baru memiliki kunci bucket S3 yang diaktifkan.

Untuk melihat pengaturan Kunci Bucket S3 untuk bucket Anda

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Di dalam daftar Bucket, pilih bucket yang ingin Anda aktifkan Kunci Bucket S3-nya.
4. Pilih Properti.
5. Di bagian Enkripsi default, di bawah Kunci Bucket, Anda melihat pengaturan Tombol Bucket S3 untuk bucket Anda.

   Jika Anda tidak dapat melihat pengaturan Kunci Bucket S3, Anda mungkin tidak memiliki izin untuk menjalankan tindakan `s3:GetEncryptionConfiguration`. Untuk informasi selengkapnya, lihat GetBucketEncryption dalam Referensi API Amazon Simple Storage Service.

Untuk melihat pengaturan Kunci Bucket S3 untuk bucket Anda

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di dalam daftar Bucket, pilih bucket yang ingin Anda aktifkan Kunci Bucket S3-nya.
3. Di daftar Objek, pilih nama objek Anda.

4. Di atas tab Detail, di bawah Pengaturan enkripsi di sisi server, memilih Edit.

   Pada Kunci Bucket, Anda melihat pengaturan Kunci Bucket S3 untuk objek Anda. Anda tidak bisa mengedit pengaturan ini.

Menggunakan AWS CLI

Untuk mengembalikan pengaturan Kunci Bucket S3 tingkat bucket

Untuk menggunakan contoh ini, ganti masing-masing *user input placeholder* dengan informasi Anda sendiri.

```
aws s3api get-bucket-encryption --bucket amzn-s3-demo-bucket1
```

Untuk informasi selengkapnya, lihat get-bucket-encryption dalam AWS CLI Referensi Perintah.

Untuk mengembalikan pengaturan Kunci Bucket S3 tingkat objek

Untuk menggunakan contoh ini, ganti masing-masing *user input placeholder* dengan informasi Anda sendiri.

```
aws s3api head-object --bucket amzn-s3-demo-bucket1 --key my_images.tar.bz2
```

Untuk informasi selengkapnya, lihat head-object dalam AWS CLI Referensi Perintah.

Penggunaan API REST

Untuk mengembalikan pengaturan Kunci Bucket S3 tingkat bucket

Untuk mengembalikan informasi enkripsi untuk bucket, termasuk pengaturan untuk Kunci Bucket S3, gunakan operasi `GetBucketEncryption`. Pengaturan Kunci bucket S3 dikembalikan di badan respons di elemen `ServerSideEncryptionConfiguration` dengan pengaturan `BucketKeyEnabled`. Untuk informasi selengkapnya, lihat GetBucketEncryption di Referensi API Amazon S3.

Untuk mengembalikan pengaturan tingkat objek untuk Kunci Bucket S3

Untuk mengembalikan status Kunci Bucket S3 untuk objek, gunakan operasi `HeadObject`. `HeadObject` mengembalikan header respons `x-amz-server-side-encryption-bucket-key-enabled` untuk menunjukkan apakah Kunci Bucket S3 diaktifkan atau dinonaktifkan untuk objek. Untuk informasi selengkapnya, lihat HeadObject di Referensi API Amazon S3.

Operasi API berikut juga mengembalikan header respons `x-amz-server-side-encryption-bucket-key-enabled` jika Kunci Bucket S3 dikonfigurasi untuk sebuah objek:

- PutObject
- PostObject
- CopyObject
- CreateMultipartUpload
- UploadPartCopy
- UploadPart
- CompleteMultipartUpload
- GetObject

Menggunakan enkripsi sisi server dual-layer dengan kunci (DSSE-KMS) AWS KMS

Menggunakan enkripsi sisi server dua lapis dengan kunci AWS Key Management Service (AWS KMS) (DSSE-KMS) menerapkan dua lapisan enkripsi ke objek saat diunggah ke Amazon S3. DSSE-KMS membantu Anda agar lebih mudah memenuhi standar kepatuhan yang mengharuskan Anda untuk menerapkan enkripsi multilayer ke data Anda dan memiliki kontrol penuh atas kunci enkripsi Anda.

Saat Anda menggunakan DSSE-KMS dengan bucket Amazon S3, AWS KMS kunci harus berada di Wilayah yang sama dengan bucket. Selain itu, ketika DSSE-KMS diminta untuk objek tersebut, checksum S3 yang merupakan bagian dari metadata objek disimpan dalam bentuk terenkripsi. Untuk informasi selengkapnya tentang checksum, lihat Memeriksa integritas objek di Amazon S3.

Ada biaya tambahan untuk menggunakan DSSE-KMS dan. AWS KMS keys Untuk informasi lebih lanjut tentang harga DSSE-KMS, lihat AWS KMS key konsep dalam AWS Key Management Service Panduan Pengguna dan AWS KMS harga.

> **Note**
>
> Kunci Bucket S3 tidak didukung untuk DSSE-KMS.

Memerlukan enkripsi sisi server dua lapis dengan (DSSE-KMS) AWS KMS keys

Untuk mewajibkan enkripsi di sisi server terhadap semua objek dalam bucket Amazon S3 tertentu, Anda dapat menggunakan kebijakan bucket. Misalnya, kebijakan bucket berikut menolak izin objek

unggah (`s3:PutObject`) untuk semua orang jika permintaan tidak menyertakan header `x-amz-server-side-encryption` yang meminta enkripsi di sisi server dengan SSE-KMS.

```
{
            "Version":"2012-10-17",
            "Id":"PutObjectPolicy",
            "Statement":[{
                  "Sid":"DenyUnEncryptedObjectUploads",
                  "Effect":"Deny",
                  "Principal":"*",
                  "Action":"s3:PutObject",
                  "Resource":"arn:aws:s3:::amzn-s3-demo-bucket/*",
                  "Condition":{
                     "StringNotEquals":{
                        "s3:x-amz-server-side-encryption":"aws:kms:dsse"
                     }
                  }
               }
            ]
         }
```

Topik

- Menentukan enkripsi di sisi server dua lapis dengan kunci AWS KMS (DSSE-KMS)

Menentukan enkripsi di sisi server dua lapis dengan kunci AWS KMS (DSSE-KMS)

Anda dapat menerapkan enkripsi saat mengunggah objek baru, atau menyalin objek yang sudah ada.

Anda dapat menentukan DSSE-KMS menggunakan konsol Amazon S3, Amazon S3 API REST, dan AWS Command Line Interface (AWS CLI). Untuk informasi selengkapnya, lihat topik berikut.

> **Note**
>
> Anda dapat menggunakan Multi-wilayah AWS KMS keys di Amazon S3. Namun, Amazon S3 saat ini memperlakukan kunci multi-Wilayah selayaknya kunci satu Wilayah, dan tidak menggunakan fitur multi-Wilayah dari kunci tersebut. Untuk informasi selengkapnya, lihat Menggunakan kunci Multi-wilayah di Panduan AWS Key Management Service Pengembang.

> **Note**
>
> Jika Anda ingin menggunakan kunci KMS yang dimiliki oleh akun lain, Anda harus memiliki izin untuk menggunakan kunci tersebut. Untuk informasi selengkapnya tentang izin lintas akun untuk kunci KMS, lihat Membuat kunci KMS yang dapat digunakan oleh akun lain di Panduan Pengembang AWS Key Management Service .

Menggunakan konsol S3

Bagian ini menjelaskan cara mengatur atau mengubah jenis enkripsi objek untuk menggunakan enkripsi sisi server dua lapis dengan AWS Key Management Service (AWS KMS) kunci (DSSE-KMS) dengan menggunakan konsol Amazon S3.

> **Note**
>
> - Anda dapat mengubah enkripsi objek jika objek Anda kurang dari 5 GB. Jika objek Anda lebih besar dari 5 GB, Anda harus menggunakan AWS CLIatau AWS SDKsuntuk mengubah enkripsi objek.
> - Untuk daftar izin tambahan yang diperlukan untuk mengubah enkripsi objek, lihatthe section called “Izin yang diperlukan untuk operasi API S3”. Misalnya kebijakan yang memberikan izin ini, lihatthe section called “Contoh kebijakan berbasis identitas”.
> - Jika Anda mengubah enkripsi objek, sebuah objek baru akan dibuat untuk menggantikan objek yang lama. Jika Penentuan Versi S3 diaktifkan, versi baru objek akan dibuat, dan objek yang sudah ada menjadi versi yang lebih lama. Peran yang mengubah properti juga menjadi pemilik objek baru (atau versi objek).

Untuk menambahkan atau mengubah enkripsi untuk objek

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi, pilih Bucket, lalu pilih tab Bucket tujuan umum. Arahkan ke bucket Amazon S3 atau folder yang berisi objek yang ingin Anda ubah.
3. Pilih kotak centang untuk objek yang ingin Anda ubah.
4. Pada menu Tindakan, pilih Edit enkripsi sisi server dari daftar opsi yang muncul.

5. Gulir ke bagian enkripsi sisi server.
6. Di bawah Pengaturan enkripsi, pilih Gunakan pengaturan bucket untuk enkripsi default, atau Ganti pengaturan bucket untuk enkripsi default.
7. Jika Anda memilih Timpa pengaturan bucket untuk enkripsi default, konfigurasikan pengaturan enkripsi berikut.

   a. Di bawah Jenis enkripsi, pilih Enkripsi sisi server dual-layer dengan AWS Key Management Service kunci (DSSE-KMS).

   b. Di bagian bawah AWS KMS kunci, lakukan salah satu langkah berikut untuk memilih kunci KMS Anda:

      - Untuk memilih dari daftar kunci KMS yang tersedia, pilih Pilih dari AWS KMS keys Anda, dan pilih Kunci KMS Anda dari daftar kunci yang tersedia.

        Kunci Kunci yang dikelola AWS (`aws/s3`) dan kunci terkelola pelanggan Anda muncul dalam daftar ini. Untuk informasi selengkapnya tentang CMK, lihat Kunci pelanggan dan AWS kunci di AWS Key Management Service Panduan Pengembang.
      - Untuk memasukkan ARN kunci KMS, pilih Masukkan AWS KMS key ARN, lalu masukkan ARN kunci KMS Anda di bidang yang muncul.
      - Untuk membuat kunci terkelola pelanggan baru di AWS KMS konsol, pilih Buat kunci KMS.

        Untuk informasi selengkapnya tentang membuat AWS KMS key, lihat Membuat kunci di Panduan AWS Key Management Service Pengembang.

> **Important**
>
> Anda hanya dapat menggunakan kunci KMS yang tersedia di Wilayah AWS yang sama dengan bucket. Konsol Amazon S3 hanya mencantumkan kunci 100 KMS pertama di Wilayah yang sama dengan bucket. Untuk menggunakan kunci KMS yang tidak terdaftar, Anda harus memasukkan ARN kunci KMS Anda. Jika Anda ingin menggunakan kunci KMS yang dimiliki oleh akun yang berbeda, Anda harus terlebih dahulu memiliki izin untuk menggunakan kunci tersebut, dan kemudian Anda harus memasukkan kunci KMS ARN.

Amazon S3 hanya mendukung kunci KMS enkripsi simetris, dan tidak mendukung kunci KMS asimetris. Untuk informasi selengkapnya, lihat Mengidentifikasi kunci KMS asimetris dalam Panduan Pengembang AWS Key Management Service .

8. Untuk Kunci Bucket, pilih Nonaktifkan. Kunci Bucket S3 tidak didukung untuk DSSE-KMS.
9. Di bawah Pengaturan salinan tambahan, pilih apakah Anda ingin Menyalin setelan sumber, Jangan tentukan pengaturan, atau Tentukan pengaturan. Pengaturan sumber salin adalah opsi default. Jika Anda hanya ingin menyalin objek tanpa atribut pengaturan sumber, pilih Jangan tentukan pengaturan. Pilih Tentukan pengaturan untuk menentukan pengaturan untuk kelas penyimpanan, tag objek ACLs, metadata, enkripsi sisi server, dan checksum tambahan.
10. Pilih Simpan perubahan.

Note

Tindakan ini menerapkan enkripsi untuk semua objek yang ditentukan. Saat mengenkripsi folder, tunggu hingga operasi penyimpanannya selesai sebelum menambahkan objek baru ke folder tersebut.

Penggunaan API REST

Saat Anda membuat objek — yaitu, ketika Anda mengunggah objek baru atau menyalin objek yang ada — Anda dapat menentukan penggunaan enkripsi sisi server dua lapis dengan (DSSE-KMS) untuk mengenkripsi data Anda. AWS KMS keys Untuk melakukannya, tambahkan header `x-amz-server-side-encryption` ke permintaan. Atur nilai header ke algoritma enkripsi. `aws:kms:dsse`. Amazon S3 mengonfirmasi bahwa objek Anda disimpan dengan enkripsi DSSE-KMS dengan mengembalikan header respons `x-amz-server-side-encryption`.

Jika Anda menentukan header `x-amz-server-side-encryption` dengan nilai `aws:kms:dsse`, Anda juga dapat menggunakan header permintaan berikut ini:

- `x-amz-server-side-encryption-aws-kms-key-id: SSEKMSKeyId`
- `x-amz-server-side-encryption-context: SSEKMSEncryptionContext`

Topik

- Operasi API REST Amazon S3 yang mendukung DSSE-KMS

- Konteks enkripsi (x-amz-server-side-encryption-context)
- AWS KMS ID kunci (x-amz-server-side-encryption-aws-kms-key-id)

Operasi API REST Amazon S3 yang mendukung DSSE-KMS

Operasi API REST berikut ini menerima header permintaan `x-amz-server-side-encryption`, `x-amz-server-side-encryption-aws-kms-key-id`, dan `x-amz-server-side-encryption-context`.

- PutObject–Saat Anda mengunggah data dengan menggunakan operasi API PUT, Anda dapat menentukan header permintaan ini.
- CopyObject–Saat Anda menyalin objek, Anda memiliki objek sumber dan objek target. Saat Anda melewati header DSSE-KMS dengan operasi `CopyObject`, itu diterapkan hanya untuk objek target. Saat Anda menyalin objek yang ada, terlepas dari apakah objek sumbernya dienkripsi atau tidak, objek tujuan tidak dienkripsi kecuali jika Anda secara eksplisit meminta enkripsi di sisi server.
- POST Object–Saat Anda menggunakan operasi POST untuk mengunggah objek, alih-alih header permintaan, Anda memberikan informasi yang sama di kolom formulir.
- CreateMultipartUpload–Saat Anda mengunggah objek besar dengan menggunakan unggahan multibagian, Anda dapat menentukan header ini dalam permintaan `CreateMultipartUpload`.

Header respons dari operasi API REST berikut mengembalikan header `x-amz-server-side-encryption` saat objek disimpan dengan enkripsi di sisi server.

- PutObject
- CopyObject
- POST Objek
- CreateMultipartUpload
- UploadPart
- UploadPartCopy
- CompleteMultipartUpload
- GetObject
- HeadObject

> **Important**
>
> - Semua `GET` dan `PUT` permintaan untuk objek yang dilindungi oleh AWS KMS gagal jika Anda tidak membuatnya dengan menggunakan Secure Sockets Layer (SSL), Transport Layer Security (TLS), atau Signature Version 4.
> - Jika objek Anda menggunakan DSSE-KMS, jangan mengirim header permintaan enkripsi untuk permintaan `GET` dan permintaan `HEAD`, atau Anda akan mendapatkan kesalahan HTTP 400 (Bad Request).

Konteks enkripsi (**`x-amz-server-side-encryption-context`**)

Jika Anda menentukan `x-amz-server-side-encryption:aws:kms:dsse`, API Amazon S3 mendukung konteks enkripsi dengan header `x-amz-server-side-encryption-context`. Konteks enkripsi adalah seperangkat pasangan nilai kunci yang berisi informasi kontekstual tambahan terkait data.

Amazon S3 secara otomatis menggunakan Amazon Resource Name (ARN) objek sebagai pasangan konteks enkripsi; misalnya, `arn:aws:s3:::`*`object_ARN`*.

Anda dapat secara opsional memberikan pasangan konteks enkripsi tambahan dengan menggunakan header `x-amz-server-side-encryption-context`. Namun, karena konteks enkripsi tidak dienkripsi, pastikan bahwa itu tidak menyertakan informasi yang sensitif. Amazon S3 menyimpan pasangan kunci tambahan ini bersama dengan konteks enkripsi default.

Untuk informasi tentang konteks enkripsi di Amazon S3, lihat Konteks enkripsi. Untuk informasi umum tentang konteks enkripsi, lihat AWS Key Management Service Konsep-Konteks enkripsi dalam Panduan Pengembang AWS Key Management Service .

AWS KMS ID kunci (**`x-amz-server-side-encryption-aws-kms-key-id`**)

Anda dapat menggunakan header `x-amz-server-side-encryption-aws-kms-key-id` untuk menentukan ID CMK, yang digunakan untuk melindungi data. Jika Anda menentukan `x-amz-server-side-encryption:aws:kms:dsse` header tetapi tidak memberikan `x-amz-server-side-encryption-aws-kms-key-id` header, Amazon S3 menggunakan Kunci yang dikelola AWS (`aws/s3`) untuk melindungi data. Jika Anda ingin menggunakan CMK, Anda harus memberikan header `x-amz-server-side-encryption-aws-kms-key-id` kunci yang dikelola pelanggan.

> **Important**
>
> Saat Anda menggunakan enkripsi sisi server AWS KMS key untuk Amazon S3, Anda harus memilih kunci KMS enkripsi simetris. Amazon S3 hanya mendukung kunci KMS enkripsi simetris. Untuk informasi selengkapnya terkait kunci ini, lihat Membuat kunci enkripsi simetris KMS dalam Panduan Pengembang AWS Key Management Service .

Menggunakan AWS CLI

Saat Anda mengunggah objek baru atau menyalin objek yang ada, Anda dapat menentukan penggunaan DSSE-KMS untuk mengenkripsi data Anda. Untuk melakukannya, tambahkan parameter `--server-side-encryption aws:kms:dsse` ke permintaan. Gunakan parameter `--ssekms-key-id` *`example-key-id`* untuk menambahkan kunci AWS KMS yang dikelola pelanggan yang telah Anda buat. Jika Anda menentukan`--server-side-encryption aws:kms:dsse`, tetapi tidak memberikan ID AWS KMS kunci, maka Amazon S3 akan menggunakan kunci AWS terkelola ()`aws/s3`.

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key example-object-key --server-
side-encryption aws:kms:dsse --ssekms-key-id example-key-id --body filepath
```

Anda dapat mengenkripsi objek yang tidak terenkripsi untuk menggunakan DSSE-KMS dengan menyalin objek kembali ke tempatnya.

```
aws s3api copy-object --bucket amzn-s3-demo-bucket --key example-object-key --copy-
source amzn-s3-demo-bucket/example-object-key --server-side-encryption aws:kms:dsse --
ssekms-key-id example-key-id
```

Menggunakan enkripsi di sisi server dengan kunci yang disediakan pelanggan (SSE-C)

Enkripsi sisi-server adalah tentang melindungi data diam. Enkripsi di sisi server hanya mengenkripsi data objek, bukan metadata objek. Dengan menggunakan enkripsi sisi server dengan kunci yang disediakan pelanggan (SSE-C), Anda dapat menyimpan data yang dienkripsi dengan kunci enkripsi Anda sendiri. Dengan kunci enkripsi yang Anda sediakan sebagai bagian dari permintaan Anda, Amazon S3 mengelola enkripsi data saat menulis ke disk dan dekripsi data saat Anda mengakses objek Anda. Oleh karena itu, Anda tidak perlu menyimpan kode apa pun untuk melakukan enkripsi dan dekripsi data. Satu-satunya hal yang perlu Anda lakukan adalah mengelola kunci enkripsi yang Anda berikan.

Saat Anda mengunggah objek, Amazon S3 menggunakan kunci enkripsi yang Anda berikan untuk menerapkan enkripsi AES-256 ke data Anda. Amazon S3 kemudian menghapus kunci enkripsi dari memori. Saat mengambil sebuah objek, Anda harus memberikan kunci enkripsi yang sama sebagai bagian dari permintaan Anda. Amazon S3 pertama-tama memverifikasi bahwa kunci enkripsi yang Anda berikan cocok, lalu mendekripsi objek sebelum mengembalikan data objek kepada Anda.

Tidak ada biaya tambahan untuk penggunaan SSE-C. Namun, permintaan untuk mengonfigurasi dan menggunakan SSE-C dikenakan biaya permintaan Amazon S3 standar. Untuk informasi tentang harga, lihat Harga Amazon S3.

> **Note**
>
> Amazon S3 tidak menyimpan kunci enkripsi yang Anda sediakan. Sebaliknya, ia menyimpan nilai Kode Autentikasi Pesan Berbasis Hash (HMAC) salted secara acak dari kunci enkripsi untuk memvalidasi permintaan di masa mendatang. Nilai HMAC salted tidak dapat digunakan untuk mendapatkan nilai kunci enkripsi atau untuk mendekripsi konten objek terenkripsi. Artinya, jika Anda kehilangan kunci enkripsi, Anda akan kehilangan objek.

S3 Replication mendukung objek yang dienkripsi dengan SSE-C. Untuk informasi selengkapnya tentang mereplikasi objek terenkripsi, lihat the section called “Mereplikasi objek terenkripsi”.

Untuk informasi selengkapnya tentang SSE-C, lihat topik berikut.

Topik

- Gambaran umum SSE-C
- Membutuhkan dan membatasi SSE-C
- Presigned URLs dan SSE-C
- Menentukan enkripsi di sisi server dengan kunci yang disediakan pelanggan (SSE-C)

Gambaran umum SSE-C

Bagian ini memberikan gambaran umum SSE-C. Saat menggunakan SSE-C, perhatikan pertimbangan berikut ini.

- Anda harus menggunakan HTTPS.

> **Important**
>
> Amazon S3 menolak permintaan apa pun yang dibuat melalui HTTP saat menggunakan SSE-C. Untuk pertimbangan keamanan, kami menyarankan Anda mempertimbangkan kunci apa pun yang salah Anda kirimkan melalui HTTP telah disusupi. Buang kuncinya dan putar seperlunya.

- Entity tag (ETag) dalam respon bukanlah MD5 hash dari data objek.
- Anda mengelola pemetaan kunci enkripsi mana yang digunakan untuk mengenkripsi objek yang ingin dituju. Amazon S3 tidak menyimpan kunci enkripsi. Anda bertanggung jawab untuk melacak kunci enkripsi yang Anda berikan untuk objek yang ingin dituju.
  - Jika versi bucket Anda diaktifkan dengan Penentuan Versi, setiap versi objek yang Anda unggah menggunakan fitur ini dapat memiliki kunci enkripsinya sendiri. Anda bertanggung jawab untuk melacak kunci enkripsi yang Anda berikan untuk objek yang ingin dituju.
  - Karena Anda mengelola kunci enkripsi di sisi klien, Anda mengelola perlindungan tambahan, seperti rotasi utama, di sisi klien.

> **Warning**
>
> Jika Anda kehilangan kunci enkripsi, setiap permintaan `GET` untuk objek tanpa kunci enkripsi akan gagal, dan Anda kehilangan objek tersebut.

Membutuhkan dan membatasi SSE-C

Untuk mewajibkan SSE-C bagi semua objek dalam bucket Amazon S3 tertentu, Anda dapat menggunakan kebijakan bucket.

Misalnya, kebijakan bucket berikut menolak izin unggah objek (`s3:PutObject`) untuk semua permintaan yang tidak menyertakan header `x-amz-server-side-encryption-customer-algorithm` yang meminta SSE-C.

```
{
    "Version": "2012-10-17",
    "Id": "PutObjectPolicy",
    "Statement": [
        {
            "Sid": "RequireSSECObjectUploads",
```

```
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Condition": {
                "Null": {
                    "s3:x-amz-server-side-encryption-customer-algorithm": "true"
                }
            }
        }
    ]
}
```

Anda juga dapat menggunakan kebijakan untuk membatasi enkripsi di sisi server pada semua objek di bucket Amazon S3 tertentu. Misalnya, kebijakan bucket berikut menolak izin unggah objek (`s3:PutObject`) untuk semua permintaan yang tidak menyertakan header `x-amz-server-side-encryption-customer-algorithm` yang meminta SSE-C.

```
{
    "Version": "2012-10-17",
    "Id": "PutObjectPolicy",
    "Statement": [
        {
            "Sid": "RestrictSSECObjectUploads",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Condition": {
                "Null": {
                    "s3:x-amz-server-side-encryption-customer-algorithm": "false"
                }
            }
        }
    ]
}
```

Important

Jika Anda menggunakan kebijakan bucket untuk mengharuskan SSE-C aktif`s3:PutObject`, Anda harus menyertakan `x-amz-server-side-encryption-customer-algorithm`

header di semua permintaan unggahan multibagian (CreateMultipartUpload, UploadPart, dan). CompleteMultipartUpload

Presigned URLs dan SSE-C

Anda dapat membuat URL yang telah ditandatangani sebelumnya yang dapat digunakan untuk operasi seperti mengunggah objek baru, mengambil objek yang sudah ada, atau mengambil metadata objek. URLsDukungan presigned SSE-C sebagai berikut:

- Saat membuat URL yang telah ditandatangani, Anda harus menentukan algoritma dengan menggunakan header `x-amz-server-side-encryption-customer-algorithm` dalam perhitungan tanda tangan.
- Saat menggunakan URL yang ditandatangani sebelumnya untuk mengunggah objek baru, mengambil objek yang sudah ada, atau mengambil metadata objek saja, Anda harus memberikan semua header enkripsi dalam permintaan aplikasi klien.

> **Note**
>
> Untuk non-SSE-C objek, Anda dapat membuat URL yang telah ditetapkan sebelumnya dan langsung menempelkan URL tersebut ke browser untuk mengakses data.
> Namun, Anda tidak dapat melakukan ini untuk objek SSE-C, karena selain URL yang ditandatangani sebelumnya, Anda juga harus menyertakan header HTTP yang kustom untuk objek SSE-C. Oleh karena itu, Anda dapat menggunakan presigned URLs untuk objek SSE-C hanya secara terprogram.

Untuk informasi lebih lanjut tentang presigned URLs, lihatthe section called “Menggunakan presigned URLs untuk mengunduh dan mengunggah objek”.

Menentukan enkripsi di sisi server dengan kunci yang disediakan pelanggan (SSE-C)

Pada saat pembuatan objek dengan API REST, Anda dapat menentukan enkripsi di sisi server dengan kunci yang disediakan pelanggan (SSE-C). Saat Anda menggunakan SSE-C, Anda harus memberikan informasi kunci enkripsi menggunakan header permintaan berikut.

Nama	Deskripsi
`x-amz-server-side-encryption-customer-algorithm`	Gunakan header ini untuk menentukan algoritma enkripsi. Nilai header harus berupa `AES256`.
`x-amz-server-side-encryption-customer-key`	Gunakan header ini untuk menyediakan kunci enkripsi 256 bit, berkode base64 agar Amazon S3 dapat digunakan untuk mengenkripsi atau mendekripsi data Anda.
`x-amz-server-side-encryption-customer-key-MD5`	Gunakan header ini untuk memberikan MD5 intisari 128-bit yang dikodekan base64 dari kunci enkripsi menurut RFC 1321. Amazon S3 menggunakan header ini untuk pemeriksaan integritas pesan guna memastikan bahwa kunci enkripsi dikirimkan tanpa kesalahan.

Anda dapat menggunakan pustaka pembungkus AWS SDK untuk menambahkan header ini ke permintaan Anda. Jika perlu, Anda dapat melakukan panggilan API REST Amazon S3 secara langsung di aplikasi Anda.

> **Note**
>
> Anda tidak dapat menggunakan konsol Amazon S3 untuk mengunggah objek dan meminta SSE-C. Anda juga tidak dapat menggunakan konsol untuk memperbarui (misalnya, mengubah kelas penyimpanan atau menambahkan metadata) objek yang sudah ada yang disimpan menggunakan SSE-C.

Penggunaan API REST

Amazon S3 sisanya yang mendukung APIs SSE-C

Amazon S3 berikut APIs mendukung enkripsi sisi server dengan kunci enkripsi yang disediakan pelanggan (SSE-C).

- GET operation–Saat mengambil objek menggunakan API GET (lihat GET Object), Anda dapat menentukan header permintaan.

- HEAD operation–Untuk mengambil metadata objek menggunakan API HEAD (lihat HEAD Object), Anda dapat menentukan header permintaan ini.
- PUT operation–Saat Anda mengunggah data dengan menggunakan API PUT Objek (lihat PUT Objek), Anda dapat menentukan header permintaan ini.
- Unggahan Multibagian–Saat mengunggah objek yang berukuran besar menggunakan API unggahan multibagian, Anda dapat menentukan header ini. Anda menentukan header ini dalam permintaan inisiasi (lihat Mulai Pengunggahan Multibagian) dan setiap permintaan pengunggahan bagian berikutnya (lihat Unggah Bagian atau Unggah Bagian-Salin). Untuk setiap permintaan unggahan bagian, informasi enkripsi harus sama dengan yang Anda berikan dalam permintaan unggahan multibagian.
- POST operation–Saat menggunakan POST operation untuk mengunggah objek (lihat Objek POST), alih-alih dengan header permintaan, Anda memberikan informasi yang sama di dalam bidang formulir.
- Copy operation–Saat Anda menyalin objek (lihat PUT Objek-Salin), Anda memiliki objek sumber dan objek target:
  - Jika Anda ingin objek target dienkripsi menggunakan enkripsi sisi server dengan kunci AWS terkelola, Anda harus memberikan header permintaan. `x-amz-server-side-encryption`
  - Jika Anda ingin objek target dienkripsi menggunakan SSE-C, Anda harus memberikan informasi enkripsi menggunakan tiga header yang dijelaskan dalam tabel sebelumnya.
  - Jika objek sumber dienkripsi menggunakan SSE-C, Anda harus memberikan informasi kunci enkripsi menggunakan header berikut sehingga Amazon S3 dapat mendekripsi objek untuk disalin.

Nama	Deskripsi
`x-amz-copy-source-server-side-encryption-customer-algorithm`	Sertakan header ini untuk menentukan algoritma yang harus digunakan Amazon S3 untuk mendekripsi objek sumber. Nilai ini harus berupa `AES256`.
`x-amz-copy-source-server-side-encryption-customer-key`	Sertakan header ini untuk menyediakan kunci enkripsi berenkode base64 untuk Amazon S3 untuk digunakan untuk mendekripsi objek sumber. Kunci enkripsi ini harus berupa kunci yang Anda

Nama	Deskripsi
	berikan kepada Amazon S3 saat membuat objek sumber. Jika tidak, Amazon S3 tidak dapat mendekripsi objeknya.
`x-amz-copy-source-server-side-encryption-customer-key-MD5`	Sertakan header ini untuk memberikan MD5 intisari 128-bit yang dikodekan base64 dari kunci enkripsi menurut RFC 1321.

Menggunakan AWS SDKs untuk menentukan SSE-C untuk operasi PUT, GET, Head, dan Copy

Contoh berikut ini menunjukkan cara untuk meminta enkripsi di sisi server dengan kunci yang disediakan pelanggan (SSE-C) untuk objek. Contoh melakukan operasi berikut. Setiap operasi menunjukkan cara menentukan SSE-C-related header dalam permintaan:

- Put objek–Mengunggah objek dan meminta enkripsi di sisi server menggunakan kunci enkripsi yang disediakan pelanggan.
- Get object–Mengunduh objek yang diunggah dalam langkah sebelumnya. Dalam permintaan tersebut, Anda memberikan informasi enkripsi yang sama dengan yang Anda berikan saat unggah objek tersebut. Amazon S3 memerlukan informasi ini untuk mendekripsi objek sehingga dapat mengembalikannya kepada Anda.
- Get object metadata–Mengambil metadata objek. Anda memberikan informasi enkripsi yang sama, yang digunakan saat objek tersebut dibuat.
- Copy object–Membuat salinan dari objek yang diunggah sebelumnya. Karena objek sumber disimpan menggunakan SSE-C, Anda harus memberikan informasi enkripsinya dalam permintaan salinan Anda. Secara default, Amazon S3 mengenkripsi salinan objek hanya jika Anda secara eksplisit memintanya. Contoh ini mengarahkan Amazon S3 untuk menyimpan salinan objek yang dienkripsi.

Java

> **Note**
>
> Contoh ini menunjukkan cara untuk mengunggah objek dalam satu operasi. Saat menggunakan API Unggahan Multibagian untuk mengunggah objek besar, Anda

memberikan informasi enkripsi dengan cara yang sama seperti yang ditunjukkan dalam contoh ini. Untuk contoh unggahan multipart yang menggunakan file AWS SDK untuk Java, lihat. Pengunggahan objek menggunakan unggahan multibagian

Untuk menambahkan informasi enkripsi yang diperlukan, Anda menyertakan `SSECustomerKey` dalam permintaan Anda. Untuk informasi selengkapnya tentang kelas `SSECustomerKey`, lihat bagian API REST.

Untuk informasi tentang SSE-C, lihat Menggunakan enkripsi di sisi server dengan kunci yang disediakan pelanggan (SSE-C). Untuk petunjuk cara membuat dan menguji sampel yang berfungsi, lihat Memulai di Panduan AWS SDK untuk Java Pengembang.

Example

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import javax.crypto.KeyGenerator;
import java.io.BufferedReader;
import java.io.File;
import java.io.IOException;
import java.io.InputStreamReader;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;

public class ServerSideEncryptionUsingClientSideEncryptionKey {
    private static SSECustomerKey SSE_KEY;
    private static AmazonS3 S3_CLIENT;
    private static KeyGenerator KEY_GENERATOR;

    public static void main(String[] args) throws IOException,
 NoSuchAlgorithmException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String keyName = "*** Key name ***";
```

```
        String uploadFileName = "*** File path ***";
        String targetKeyName = "*** Target key name ***";

        // Create an encryption key.
        KEY_GENERATOR = KeyGenerator.getInstance("AES");
        KEY_GENERATOR.init(256, new SecureRandom());
        SSE_KEY = new SSECustomerKey(KEY_GENERATOR.generateKey());

        try {
            S3_CLIENT = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            // Upload an object.
            uploadObject(bucketName, keyName, new File(uploadFileName));

            // Download the object.
            downloadObject(bucketName, keyName);

            // Verify that the object is properly encrypted by attempting to
retrieve it
            // using the encryption key.
            retrieveObjectMetadata(bucketName, keyName);

            // Copy the object into a new object that also uses SSE-C.
            copyObject(bucketName, keyName, targetKeyName);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }

    private static void uploadObject(String bucketName, String keyName, File file) {
        PutObjectRequest putRequest = new PutObjectRequest(bucketName, keyName,
file).withSSECustomerKey(SSE_KEY);
        S3_CLIENT.putObject(putRequest);
        System.out.println("Object uploaded");
    }
```

```
   private static void downloadObject(String bucketName, String keyName) throws
IOException {
       GetObjectRequest getObjectRequest = new GetObjectRequest(bucketName,
keyName).withSSECustomerKey(SSE_KEY);
       S3Object object = S3_CLIENT.getObject(getObjectRequest);

       System.out.println("Object content: ");
       displayTextInputStream(object.getObjectContent());
   }

   private static void retrieveObjectMetadata(String bucketName, String keyName) {
       GetObjectMetadataRequest getMetadataRequest = new
GetObjectMetadataRequest(bucketName, keyName)
               .withSSECustomerKey(SSE_KEY);
       ObjectMetadata objectMetadata =
S3_CLIENT.getObjectMetadata(getMetadataRequest);
       System.out.println("Metadata retrieved. Object size: " +
objectMetadata.getContentLength());
   }

   private static void copyObject(String bucketName, String keyName, String
targetKeyName)
           throws NoSuchAlgorithmException {
       // Create a new encryption key for target so that the target is saved using
       // SSE-C.
       SSECustomerKey newSSEKey = new SSECustomerKey(KEY_GENERATOR.generateKey());

       CopyObjectRequest copyRequest = new CopyObjectRequest(bucketName, keyName,
bucketName, targetKeyName)
               .withSourceSSECustomerKey(SSE_KEY)
               .withDestinationSSECustomerKey(newSSEKey);

       S3_CLIENT.copyObject(copyRequest);
       System.out.println("Object copied");
   }

   private static void displayTextInputStream(S3ObjectInputStream input) throws
IOException {
       // Read one line at a time from the input stream and display each line.
       BufferedReader reader = new BufferedReader(new InputStreamReader(input));
       String line;
       while ((line = reader.readLine()) != null) {
           System.out.println(line);
```

```
        }
        System.out.println();
    }
}
```

.NET

> **Note**
>
> Untuk contoh pengunggahan objek besar menggunakan API unggahan multibagian, lihat Pengunggahan objek menggunakan unggahan multibagian dan Menggunakan AWS SDKs (API tingkat rendah).

Untuk informasi tentang SSE-C, lihat Menggunakan enkripsi di sisi server dengan kunci yang disediakan pelanggan (SSE-C). Untuk informasi tentang menyiapkan dan menjalankan contoh kode, lihat Memulai SDK for .NET di AWSAWS SDK for .NET Developer Guide.

Example

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.IO;
using System.Security.Cryptography;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class SSEClientEncryptionKeyObjectOperationsTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string keyName = "*** key name for new object created ***";
        private const string copyTargetKeyName = "*** key name for object copy ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
```

```
        {
            client = new AmazonS3Client(bucketRegion);
            ObjectOpsUsingClientEncryptionKeyAsync().Wait();
        }
        private static async Task ObjectOpsUsingClientEncryptionKeyAsync()
        {
            try
            {
                // Create an encryption key.
                Aes aesEncryption = Aes.Create();
                aesEncryption.KeySize = 256;
                aesEncryption.GenerateKey();
                string base64Key = Convert.ToBase64String(aesEncryption.Key);

                // 1. Upload the object.
                PutObjectRequest putObjectRequest = await
UploadObjectAsync(base64Key);
                // 2. Download the object and verify that its contents matches what
you uploaded.
                await DownloadObjectAsync(base64Key, putObjectRequest);
                // 3. Get object metadata and verify that the object uses AES-256
encryption.
                await GetObjectMetadataAsync(base64Key);
                // 4. Copy both the source and target objects using server-side
encryption with
                //    a customer-provided encryption key.
                await CopyObjectAsync(aesEncryption, base64Key);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered ***. Message:'{0}' when writing
an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
        }

        private static async Task<PutObjectRequest> UploadObjectAsync(string
base64Key)
        {
            PutObjectRequest putObjectRequest = new PutObjectRequest
```

```
            {
                BucketName = bucketName,
                Key = keyName,
                ContentBody = "sample text",
                ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                ServerSideEncryptionCustomerProvidedKey = base64Key
            };
            PutObjectResponse putObjectResponse = await
client.PutObjectAsync(putObjectRequest);
            return putObjectRequest;
        }
        private static async Task DownloadObjectAsync(string base64Key,
PutObjectRequest putObjectRequest)
        {
            GetObjectRequest getObjectRequest = new GetObjectRequest
            {
                BucketName = bucketName,
                Key = keyName,
                // Provide encryption information for the object stored in Amazon
S3.
                ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                ServerSideEncryptionCustomerProvidedKey = base64Key
            };

            using (GetObjectResponse getResponse = await
client.GetObjectAsync(getObjectRequest))
            using (StreamReader reader = new
StreamReader(getResponse.ResponseStream))
            {
                string content = reader.ReadToEnd();
                if (String.Compare(putObjectRequest.ContentBody, content) == 0)
                    Console.WriteLine("Object content is same as we uploaded");
                else
                    Console.WriteLine("Error...Object content is not same.");

                if (getResponse.ServerSideEncryptionCustomerMethod ==
ServerSideEncryptionCustomerMethod.AES256)
                    Console.WriteLine("Object encryption method is AES256, same as
we set");
                else
                    Console.WriteLine("Error...Object encryption method is not the
same as AES256 we set");
```

```
                // Assert.AreEqual(putObjectRequest.ContentBody, content);
                // Assert.AreEqual(ServerSideEncryptionCustomerMethod.AES256,
getResponse.ServerSideEncryptionCustomerMethod);
            }
        }
        private static async Task GetObjectMetadataAsync(string base64Key)
        {
            GetObjectMetadataRequest getObjectMetadataRequest = new
GetObjectMetadataRequest
            {
                BucketName = bucketName,
                Key = keyName,

                // The object stored in Amazon S3 is encrypted, so provide the
necessary encryption information.
                ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                ServerSideEncryptionCustomerProvidedKey = base64Key
            };

            GetObjectMetadataResponse getObjectMetadataResponse = await
client.GetObjectMetadataAsync(getObjectMetadataRequest);
            Console.WriteLine("The object metadata show encryption method used is:
{0}", getObjectMetadataResponse.ServerSideEncryptionCustomerMethod);
            // Assert.AreEqual(ServerSideEncryptionCustomerMethod.AES256,
getObjectMetadataResponse.ServerSideEncryptionCustomerMethod);
        }
        private static async Task CopyObjectAsync(Aes aesEncryption, string
base64Key)
        {
            aesEncryption.GenerateKey();
            string copyBase64Key = Convert.ToBase64String(aesEncryption.Key);

            CopyObjectRequest copyRequest = new CopyObjectRequest
            {
                SourceBucket = bucketName,
                SourceKey = keyName,
                DestinationBucket = bucketName,
                DestinationKey = copyTargetKeyName,
                // Information about the source object's encryption.
                CopySourceServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                CopySourceServerSideEncryptionCustomerProvidedKey = base64Key,
```

```
                // Information about the target object's encryption.
                ServerSideEncryptionCustomerMethod =
 ServerSideEncryptionCustomerMethod.AES256,
                ServerSideEncryptionCustomerProvidedKey = copyBase64Key
            };
            await client.CopyObjectAsync(copyRequest);
        }
    }
}
```

Menggunakan AWS SDKs untuk menentukan SSE-C untuk unggahan multipart

Contoh di bagian sebelumnya menunjukkan cara meminta enkripsi di sisi server dengan kunci yang disediakan pelanggan (SSE-C) dalam operasi PUT, GET, Head, dan Copy. Bagian ini menjelaskan Amazon S3 lain APIs yang mendukung SSE-C.

Java

Untuk mengunggah objek besar, Anda dapat menggunakan API unggahan multibagian (lihat Mengunggah dan menyalin objek menggunakan unggahan multipart di Amazon S3). Anda dapat menggunakan level tinggi atau level rendah APIs untuk mengunggah objek besar. Ini APIs mendukung header terkait enkripsi dalam permintaan.

- Saat menggunakan API `TransferManager` tingkat tinggi, Anda menyediakan header kustom enkripsi di `PutObjectRequest` (lihat Pengunggahan objek menggunakan unggahan multibagian).
- Saat menggunakan API tingkat rendah, Anda memberikan informasi terkait enkripsi pada `InitiateMultipartUploadRequest`, diikuti dengan informasi enkripsi identik pada setiap `UploadPartRequest`. Anda tidak perlu memberikan header kustom enkripsi apa pun dalam `CompleteMultipartUploadRequest` Anda. Sebagai contoh, lihat Menggunakan AWS SDKs (API tingkat rendah).

Contoh berikut menggunakan `TransferManager` untuk membuat objek dan menunjukkan cara memberikan informasi terkait SSE-C. Contoh ini melakukan hal berikut:

- Membuat objek menggunakan metode `TransferManager.upload()`. Di dalam instans `PutObjectRequest`, Anda memberikan informasi kunci enkripsi untuk meminta. Amazon S3 mengenkripsi objek menggunakan kunci yang disediakan pelanggan.

- Membuat salinan objek dengan memanggil metode `TransferManager.copy()`. Contoh tersebut mengarahkan Amazon S3 untuk mengenkripsi salinan objek menggunakan `SSECustomerKey` yang baru. Karena objek sumber dienkripsi menggunakan SSE-C, `CopyObjectRequest` juga menyediakan kunci enkripsi objek sumber sehingga Amazon S3 dapat mendekripsi objek tersebut sebelum menyalinnya.

Example

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.CopyObjectRequest;
import com.amazonaws.services.s3.model.PutObjectRequest;
import com.amazonaws.services.s3.model.SSECustomerKey;
import com.amazonaws.services.s3.transfer.Copy;
import com.amazonaws.services.s3.transfer.TransferManager;
import com.amazonaws.services.s3.transfer.TransferManagerBuilder;
import com.amazonaws.services.s3.transfer.Upload;

import javax.crypto.KeyGenerator;
import java.io.File;
import java.security.SecureRandom;

public class ServerSideEncryptionCopyObjectUsingHLwithSSEC {

    public static void main(String[] args) throws Exception {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String fileToUpload = "*** File path ***";
        String keyName = "*** New object key name ***";
        String targetKeyName = "*** Key name for object copy ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withRegion(clientRegion)
                    .withCredentials(new ProfileCredentialsProvider())
                    .build();
            TransferManager tm = TransferManagerBuilder.standard()
```

```
                    .withS3Client(s3Client)
                    .build();

            // Create an object from a file.
            PutObjectRequest putObjectRequest = new PutObjectRequest(bucketName,
keyName, new File(fileToUpload));

            // Create an encryption key.
            KeyGenerator keyGenerator = KeyGenerator.getInstance("AES");
            keyGenerator.init(256, new SecureRandom());
            SSECustomerKey sseCustomerEncryptionKey = new
SSECustomerKey(keyGenerator.generateKey());

            // Upload the object. TransferManager uploads asynchronously, so this
call
            // returns immediately.
            putObjectRequest.setSSECustomerKey(sseCustomerEncryptionKey);
            Upload upload = tm.upload(putObjectRequest);

            // Optionally, wait for the upload to finish before continuing.
            upload.waitForCompletion();
            System.out.println("Object created.");

            // Copy the object and store the copy using SSE-C with a new key.
            CopyObjectRequest copyObjectRequest = new CopyObjectRequest(bucketName,
keyName, bucketName, targetKeyName);
            SSECustomerKey sseTargetObjectEncryptionKey = new
SSECustomerKey(keyGenerator.generateKey());
            copyObjectRequest.setSourceSSECustomerKey(sseCustomerEncryptionKey);

copyObjectRequest.setDestinationSSECustomerKey(sseTargetObjectEncryptionKey);

            // Copy the object. TransferManager copies asynchronously, so this call
returns
            // immediately.
            Copy copy = tm.copy(copyObjectRequest);

            // Optionally, wait for the upload to finish before continuing.
            copy.waitForCompletion();
            System.out.println("Copy complete.");
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
```

```
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

Untuk mengunggah objek besar, Anda dapat menggunakan API unggahan multibagian (lihatMengunggah dan menyalin objek menggunakan unggahan multipart di Amazon S3). AWS SDK for .NET menyediakan level tinggi atau APIs level rendah untuk mengunggah objek besar. Ini APIs mendukung header terkait enkripsi dalam permintaan.

- Saat menggunakan API `Transfer-Utility` tingkat tinggi, Anda menyediakan header kustom enkripsi di `TransferUtilityUploadRequest` seperti yang ditunjukkan. Untuk contoh kode, lihat Pengunggahan objek menggunakan unggahan multibagian.

```
TransferUtilityUploadRequest request = new TransferUtilityUploadRequest()
{
    FilePath = filePath,
    BucketName = existingBucketName,
    Key = keyName,
    // Provide encryption information.
    ServerSideEncryptionCustomerMethod =
 ServerSideEncryptionCustomerMethod.AES256,
    ServerSideEncryptionCustomerProvidedKey = base64Key,
};
```

- Saat menggunakan API tingkat rendah, Anda memberikan informasi terkait enkripsi dalam memulai permintaan unggahan multibagian, diikuti dengan informasi enkripsi yang sama dalam permintaan bagian unggahan berikutnya. Anda tidak perlu memberikan header kustom enkripsi apa pun dalam permintaan pengunggahan multibagian lengkap Anda. Sebagai contoh, lihat Menggunakan AWS SDKs (API tingkat rendah).

  Berikut ini adalah contoh pengunggahan multibagian tingkat rendah yang membuat salinan objek besar yang sudah ada. Pada contoh, objek yang akan disalin disimpan di Amazon S3 menggunakan SSE-C, dan Anda ingin menyimpan objek target juga menggunakan SSE-C. Dalam contoh ini, Anda melakukan hal berikut:

- Mulai permintaan pengunggahan multibagian dengan memberikan kunci enkripsi dan informasi terkait.
- Menyediakan kunci enkripsi objek sumber dan sasaran serta informasi terkait dalam `CopyPartRequest`.
- Dapatkan ukuran objek sumber yang akan disalin dengan mengambil metadata objek.
- Unggah objek ke dalam bagian 5 MB.

Example

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.IO;
using System.Security.Cryptography;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class SSECLowLevelMPUcopyObjectTest
    {
        private const string existingBucketName = "*** bucket name ***";
        private const string sourceKeyName      = "*** source object key name
 ***";
        private const string targetKeyName      = "*** key name for the target
 object ***";
        private const string filePath           = @"*** file path ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
        static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            CopyObjClientEncryptionKeyAsync().Wait();
        }

        private static async Task CopyObjClientEncryptionKeyAsync()
        {
            Aes aesEncryption = Aes.Create();
```

```
            aesEncryption.KeySize = 256;
            aesEncryption.GenerateKey();
            string base64Key = Convert.ToBase64String(aesEncryption.Key);

            await CreateSampleObjUsingClientEncryptionKeyAsync(base64Key,
s3Client);

            await CopyObjectAsync(s3Client, base64Key);
        }
        private static async Task CopyObjectAsync(IAmazonS3 s3Client, string
base64Key)
        {
            List<CopyPartResponse> uploadResponses = new List<CopyPartResponse>();

            // 1. Initialize.
            InitiateMultipartUploadRequest initiateRequest = new
InitiateMultipartUploadRequest
            {
                BucketName = existingBucketName,
                Key = targetKeyName,
                ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                ServerSideEncryptionCustomerProvidedKey = base64Key,
            };

            InitiateMultipartUploadResponse initResponse =
                await s3Client.InitiateMultipartUploadAsync(initiateRequest);

            // 2. Upload Parts.
            long partSize = 5 * (long)Math.Pow(2, 20); // 5 MB
            long firstByte = 0;
            long lastByte = partSize;

            try
            {
                // First find source object size. Because object is stored
encrypted with
                // customer provided key you need to provide encryption
information in your request.
                GetObjectMetadataRequest getObjectMetadataRequest = new
GetObjectMetadataRequest()
                {
                    BucketName = existingBucketName,
                    Key = sourceKeyName,
```

```
                    ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                    ServerSideEncryptionCustomerProvidedKey = base64Key // " *
**source object encryption key ***"
                };

                GetObjectMetadataResponse getObjectMetadataResponse = await
s3Client.GetObjectMetadataAsync(getObjectMetadataRequest);

                long filePosition = 0;
                for (int i = 1; filePosition <
getObjectMetadataResponse.ContentLength; i++)
                {
                    CopyPartRequest copyPartRequest = new CopyPartRequest
                    {
                        UploadId = initResponse.UploadId,
                        // Source.
                        SourceBucket = existingBucketName,
                        SourceKey = sourceKeyName,
                        // Source object is stored using SSE-C. Provide encryption
information.
                        CopySourceServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                        CopySourceServerSideEncryptionCustomerProvidedKey =
base64Key, //"***source object encryption key ***",
                        FirstByte = firstByte,
                        // If the last part is smaller then our normal part size
then use the remaining size.
                        LastByte = lastByte >
getObjectMetadataResponse.ContentLength ?
                            getObjectMetadataResponse.ContentLength - 1 :
lastByte,

                        // Target.
                        DestinationBucket = existingBucketName,
                        DestinationKey = targetKeyName,
                        PartNumber = i,
                        // Encryption information for the target object.
                        ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                        ServerSideEncryptionCustomerProvidedKey = base64Key
                    };
                    uploadResponses.Add(await
s3Client.CopyPartAsync(copyPartRequest));
```

```
                filePosition += partSize;
                firstByte += partSize;
                lastByte += partSize;
            }

            // Step 3: complete.
            CompleteMultipartUploadRequest completeRequest = new
CompleteMultipartUploadRequest
            {
                BucketName = existingBucketName,
                Key = targetKeyName,
                UploadId = initResponse.UploadId,
            };
            completeRequest.AddPartETags(uploadResponses);

            CompleteMultipartUploadResponse completeUploadResponse =
                await s3Client.CompleteMultipartUploadAsync(completeRequest);
        }
        catch (Exception exception)
        {
            Console.WriteLine("Exception occurred: {0}", exception.Message);
            AbortMultipartUploadRequest abortMPURequest = new
AbortMultipartUploadRequest
            {
                BucketName = existingBucketName,
                Key = targetKeyName,
                UploadId = initResponse.UploadId
            };
            s3Client.AbortMultipartUpload(abortMPURequest);
        }
    }
    private static async Task
CreateSampleObjUsingClientEncryptionKeyAsync(string base64Key, IAmazonS3
s3Client)
    {
        // List to store upload part responses.
        List<UploadPartResponse> uploadResponses = new
List<UploadPartResponse>();

        // 1. Initialize.
        InitiateMultipartUploadRequest initiateRequest = new
InitiateMultipartUploadRequest
        {
            BucketName = existingBucketName,
```

```
                Key = sourceKeyName,
                ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                ServerSideEncryptionCustomerProvidedKey = base64Key
            };

            InitiateMultipartUploadResponse initResponse =
               await s3Client.InitiateMultipartUploadAsync(initiateRequest);

            // 2. Upload Parts.
            long contentLength = new FileInfo(filePath).Length;
            long partSize = 5 * (long)Math.Pow(2, 20); // 5 MB

            try
            {
                long filePosition = 0;
                for (int i = 1; filePosition < contentLength; i++)
                {
                    UploadPartRequest uploadRequest = new UploadPartRequest
                    {
                        BucketName = existingBucketName,
                        Key = sourceKeyName,
                        UploadId = initResponse.UploadId,
                        PartNumber = i,
                        PartSize = partSize,
                        FilePosition = filePosition,
                        FilePath = filePath,
                        ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                        ServerSideEncryptionCustomerProvidedKey = base64Key
                    };

                    // Upload part and add response to our list.
                    uploadResponses.Add(await
s3Client.UploadPartAsync(uploadRequest));

                    filePosition += partSize;
                }

                // Step 3: complete.
                CompleteMultipartUploadRequest completeRequest = new
CompleteMultipartUploadRequest
                {
                    BucketName = existingBucketName,
```

```
                    Key = sourceKeyName,
                    UploadId = initResponse.UploadId,
                    //PartETags = new List<PartETag>(uploadResponses)

                };
                completeRequest.AddPartETags(uploadResponses);

                CompleteMultipartUploadResponse completeUploadResponse =
                    await s3Client.CompleteMultipartUploadAsync(completeRequest);

            }
            catch (Exception exception)
            {
                Console.WriteLine("Exception occurred: {0}", exception.Message);
                AbortMultipartUploadRequest abortMPURequest = new
 AbortMultipartUploadRequest
                {
                    BucketName = existingBucketName,
                    Key = sourceKeyName,
                    UploadId = initResponse.UploadId
                };
                await s3Client.AbortMultipartUploadAsync(abortMPURequest);
            }
        }
    }
}
```

Melindungi data menggunakan enkripsi di sisi klien

Enkripsi di sisi klien adalah tindakan mengenkripsi data Anda secara lokal untuk membantu memastikan keamanannya saat sedang transit dan saat disimpan. Untuk mengenkripsi objek Anda sebelum mengirimnya ke Amazon S3, gunakan Klien Enkripsi Amazon S3. Ketika objek Anda dienkripsi dengan cara ini, objek Anda tidak terpapar ke pihak ketiga mana pun, termasuk. AWS Amazon S3 menerima objek Anda yang sudah dienkripsi; Amazon S3 tidak berperan dalam mengenkripsi atau mendekripasi objek Anda. Anda dapat menggunakan Klien Enkripsi Amazon S3 dan enkripsi di sisi server untuk mengenkripsi data Anda. Saat Anda mengirim objek terenkripsi ke Amazon S3, Amazon S3 tidak mengenali objek sebagai dienkripsi, hanya mendeteksi objek biasa.

Klien Enkripsi Amazon S3 berfungsi sebagai perantara antara Anda dan Amazon S3. Setelah Anda membuat instans Klien Enkripsi Amazon S3, objek Anda secara otomatis dienkripsi dan didekripsi sebagai bagian dari Amazon S3 dan permintaan `PutObject` dan `GetObject` Anda.

Semua objek Anda dienkripsi dengan kunci data unik. Klien Enkripsi Amazon S3 tidak menggunakan atau berinteraksi dengan kunci bucket, meskipun Anda menentukan kunci KMS sebagai kunci pembungkus.

Panduan Pengembang Klien Enkripsi Amazon S3 berfokus pada versi 3.0 dan yang lebih baru dari Klien Enkripsi Amazon S3. Untuk informasi selengkapnya, lihat Apa itu Klien Enkripsi Amazon S3? dalam Panduan Pengembang Klien Enkripsi Amazon S3.

Untuk informasi selengkapnya tentang versi klien Enkripsi Amazon S3 sebelumnya, lihat Panduan Pengembang AWS SDK untuk bahasa pemrograman Anda.

- AWS SDK untuk Java
- AWS SDK untuk .NET
- AWS SDK untuk Go
- AWS SDK untuk PHP
- AWS SDK untuk Ruby
- AWS SDK untuk C++

Privasi lalu lintas antarjaringan

Topik ini menjelaskan cara Amazon S3 mengamankan koneksi dari layanan ke lokasi lain.

Lalu lintas antara layanan dan aplikasi dan klien lokal

Koneksi berikut dapat digabungkan dengan AWS PrivateLink untuk menyediakan konektivitas antara jaringan pribadi Anda dan AWS:

- Koneksi AWS Site-to-Site VPN. Untuk informasi lebih lanjut, lihat Apa itu AWS Site-to-Site VPN?
- AWS Direct Connect Koneksi. Untuk informasi lebih lanjut, lihat Apa itu AWS Direct Connect?

Akses ke Amazon S3 melalui jaringan melalui AWS dipublikasikan. APIs Klien harus mendukung Keamanan Lapisan Pengangkutan (TLS) 1.2. Kami merekomendasikan TLS 1.3. Klien juga harus mendukung suite cipher dengan perfect forward secrecy (PFS) seperti Ephemeral Diffie-Hellman (DHE) atau Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Sebagian besar sistem modern seperti Java 7 dan versi yang lebih baru support mode ini. Selain itu, Anda harus menandatangani permintaan menggunakan kunci akses ID dan kunci akses rahasia yang terkait dengan IAM

pengguna utama, atau Anda dapat menggunakan AWS Security Token Service (STS) untuk membuat kredensial keamanan sementara guna menandatangani permintaan.

Lalu lintas antar AWS sumber daya di Wilayah yang sama

Titik akhir cloud privat virtual (VPC) untuk Amazon S3 adalah entitas logis dalam sebuah VPC yang mengizinkan konektivitas hanya ke Amazon S3. Amazon VPC merutekan permintaan ke Amazon S3 dan merutekan respons kembali ke VPC. Untuk informasi selengkapnya, lihat Titik Akhir VPC di Panduan Pengguna VPC. Untuk contoh kebijakan bucket yang dapat Anda gunakan untuk mengontrol akses bucket S3 dari titik akhir VPC, lihat Megendalikan akses dari titik akhir VPC dengan kebijakan bucket.

AWS PrivateLink untuk Amazon S3

Dengan AWS PrivateLink Amazon S3, Anda dapat menyediakan titik akhir VPC antarmuka (titik akhir antarmuka) di cloud pribadi virtual (VPC) Anda. Titik akhir ini dapat diakses secara langsung dari aplikasi yang ada di tempat melalui VPN dan AWS Direct Connect, atau dalam pengintipan VPC yang berbeda Wilayah AWS .

Endpoint antarmuka diwakili oleh satu atau lebih antarmuka jaringan elastis (ENIs) yang diberi alamat IP pribadi dari subnet di VPC Anda. Permintaan ke Amazon S3 melalui titik akhir antarmuka tetap berada di jaringan Amazon. Anda juga dapat mengakses titik akhir antarmuka di VPC Anda dari aplikasi lokal AWS Direct Connect melalui AWS Virtual Private Network atau ().AWS VPN Untuk informasi selengkapnya tentang cara menghubungkan VPC dengan jaringan on-premise Anda, lihat AWS Direct Connect Panduan Pengguna dan AWS Site-to-Site VPN Panduan Pengguna.

Untuk informasi umum tentang titik akhir antarmuka, lihat Antarmuka titik akhir VPC (AWS PrivateLink) dalam Panduan AWS PrivateLink .

Topik

- Jenis titik akhir VPC untuk Amazon S3
- Pembatasan dan batasan AWS PrivateLink untuk Amazon S3
- Membuat titik akhir VPC
- Mengakses titik akhir antarmuka Amazon S3
- DNS privat
- Mengakses bucket, titik akses, serta operasi API Amazon S3 Control dari titik akhir antarmuka S3
- Memperbarui konfigurasi DNS on-premise

- Membuat kebijakan titik akhir VPC untuk Amazon S3

Jenis titik akhir VPC untuk Amazon S3

Anda dapat menggunakan dua jenis titik akhir VPC untuk mengakses Amazon S3: titik akhir gateway dan titik akhir antarmuka (dengan menggunakan). AWS PrivateLinkTitik akhir gateway adalah gateway yang Anda tentukan dalam tabel rute untuk mengakses Amazon S3 dari VPC melalui jaringan. AWS Titik akhir antarmuka memperluas fungsionalitas titik akhir gateway dengan menggunakan alamat IP pribadi untuk merutekan permintaan ke Amazon S3 dari dalam VPC Anda, di tempat, atau dari VPC di tempat lain dengan menggunakan peering VPC atau. Wilayah AWS AWS Transit Gateway Untuk informasi selengkapnya, lihat Apa itu peering VPC? dan Transit Gateway vs Peering VPC.

Titik akhir antarmuka kompatibel dengan titik akhir gateway. Jika Anda memiliki titik akhir gateway yang ada di VPC, Anda dapat menggunakan kedua jenis titik akhir di VPC yang sama.

Titik akhir gateway untuk Amazon S3	Titik akhir antarmuka untuk Amazon S3
Dalam kedua kasus, lalu lintas jaringan Anda tetap berada di AWS jaringan.	
Gunakan alamat IP publik Amazon S3	Gunakan alamat IP privat dari VPC Anda untuk mengakses Amazon S3
Gunakan nama Amazon S3 DNS yang sama	Memerlukan nama DNS Amazon S3 kustom titik akhir
Jangan izinkan mengakses dari on-premise	Izinkan mengakses dari on-premise
Jangan izinkan akses dari yang lain Wilayah AWS	Izinkan akses dari VPC di VPC lain Wilayah AWS dengan menggunakan VPC peering atau AWS Transit Gateway
Tidak ditagih	Ditagih

Untuk informasi selengkapnya tentang titik akhir gateway, lihat Titik akhir VPC Gateway dalam Panduan AWS PrivateLink .

Pembatasan dan batasan AWS PrivateLink untuk Amazon S3

Batasan VPC berlaku AWS PrivateLink untuk Amazon S3. Untuk informasi selengkapnya, lihat Pertimbangan antarmuka titik akhir dan AWS PrivateLink kuota dalam AWS PrivateLink Panduan. Selain itu, larangan berikut juga berlaku.

AWS PrivateLink untuk Amazon S3 tidak mendukung yang berikut:

- Titik Akhir Standar Proses Informasi Federal (FIPS)
- Titik akhir situs web
- Titik akhir warisan global
- Titik akhir Wilayah dasbor S3
- Titik akhir tumpukan ganda
- Menggunakan CopyObjectatau UploadPartCopydi antara ember dalam berbagai Wilayah AWS
- Keamanan Lapisan Transportasi (TLS) 1.0
- Keamanan Lapisan Pengangkutan (TLS) 1.1
- Keamanan Lapisan Transportasi (TLS) 1.3

Membuat titik akhir VPC

Untuk membuat titik akhir antarmuka VPC, lihat Membuat titik akhir VPC dalam AWS PrivateLink Panduan.

Mengakses titik akhir antarmuka Amazon S3

Saat Anda membuat titik akhir antarmuka, Amazon S3 menghasilkan dua jenis nama DNS S3 khusus titik akhir: Regional dan zonal.

- Nama DNS Regional mencakup ID titik akhir VPC unik, pengenal layanan, Wilayah AWS dan namanya. `vpce.amazonaws.com` Sebagai contoh, untuk ID titik akhir VPC *`vpce-1a2b3c4d`*, nama DNS yang dihasilkan mungkin mirip dengan *`vpce-1a2b3c4d-5e6f`*`.s3.us-east-1.vpce.amazonaws.com`.
- Nama DNS Zonal mencakup Zona Ketersediaan—misalnya, *`vpce-1a2b3c4d-5e6f`***`-us-east-1a.`**`s3.us-east-1.vpce.amazonaws.com`. Anda dapat menggunakan opsi ini jika arsitektur Anda mengisolasi Zona Ketersediaan. Contoh, Anda bisa menggunakannya untuk kontainer kesalahan atau untuk mengurangi biaya transfer data Regional.

Nama DNS S3 spesifik titik akhir dapat diselesaikan dari domain DNS publik S3.

DNS privat

Opsi DNS privat untuk titik akhir antarmuka VPC menyederhanakan perutean lalu lintas S3 melalui titik akhir VPC dan membantu Anda memanfaatkan jalur jaringan dengan biaya terendah yang tersedia untuk aplikasi Anda. Anda dapat menggunakan opsi DNS privat untuk merutekan lalu lintas S3 Regional tanpa memperbarui klien S3 Anda untuk menggunakan nama DNS spesifik titik akhir dari titik akhir antarmuka Anda, atau mengelola infrastruktur DNS. Dengan mengaktifkan nama DNS pribadi, kueri DNS S3 Regional menyelesaikan ke alamat IP pribadi AWS PrivateLink untuk titik akhir berikut:

- Titik akhir bucket regional (misalnya, `s3.us-east-1.amazonaws.com`)
- Kontrol titik akhir (misalnya, `s3-control.us-east-1.amazonaws.com`)
- Titik akhir titik akses (misalnya, `s3-accesspoint.us-east-1.amazonaws.com`)

AWS PrivateLink untuk Amazon S3 tidak mendukung Menggunakan titik akhir tumpukan ganda Amazon S3. Jika Anda menggunakan nama DNS dual-stack S3 sebagai nama DNS pribadi, IPv6 lalu lintas Anda akan dihapus atau, jika virtual private cloud (VPC) Anda memiliki gateway internet, IPv6 lalu lintas Anda akan diarahkan melalui gateway internet di VPC Anda.

Jika Anda memiliki titik akhir gateway di VPC Anda, Anda dapat secara otomatis merutekan permintaan dalam VPC melalui titik akhir gateway S3 yang ada dan on-premise melalui titik akhir antarmuka Anda. Pendekatan ini memungkinkan Anda untuk mengoptimalkan biaya jaringan Anda dengan menggunakan titik akhir gateway, yang tidak ditagih, untuk lalu lintas in-VPC Anda. Aplikasi lokal Anda dapat digunakan AWS PrivateLink dengan bantuan titik akhir Resolver masuk. Amazon menyediakan server DNS, yang disebut Resolver Route 53, untuk VPC Anda. Titik akhir Resolver masuk meneruskan kueri DNS dari jaringan on-premise ke Resolver Route 53.

> **Important**
>
> Untuk memanfaatkan jalur jaringan dengan biaya terendah saat menggunakan Aktifkan DNS pribadi hanya untuk titik akhir masuk, titik akhir gateway harus ada di VPC Anda. Kehadiran titik akhir gateway membantu memastikan bahwa lalu lintas dalam VPC selalu merutekan melalui jaringan AWS pribadi saat opsi Aktifkan DNS pribadi hanya untuk titik akhir masuk dipilih. Anda harus mempertahankan titik akhir gateway ini sementara Anda memiliki opsi Aktifkan DNS pribadi hanya untuk titik akhir masuk yang dipilih. Jika Anda ingin menghapus

titik akhir gateway Anda, Anda harus terlebih dahulu menghapus Aktifkan DNS pribadi hanya untuk titik akhir masuk.
Jika Anda ingin memperbarui titik akhir antarmuka yang ada ke Aktifkan DNS pribadi hanya untuk titik akhir masuk, konfirmasikan terlebih dahulu bahwa VPC Anda memiliki titik akhir gateway S3. Untuk informasi selengkapnya tentang titik akhir gateway dan mengelola nama DNS privat, lihat Titik akhir VPC Gateway dan Kelola nama DNS masing-masing di Panduan AWS PrivateLink .

Opsi Aktifkan DNS pribadi hanya untuk titik akhir masuk hanya tersedia untuk layanan yang mendukung titik akhir gateway.

Untuk informasi selengkapnya tentang membuat titik akhir VPC yang menggunakan Aktifkan DNS privat hanya untuk titik akhir yang masuk, lihat Membuat titik akhir antarmuka di Panduan AWS PrivateLink .

Menggunakan konsol VPC

Di konsol Anda memiliki dua opsi: Aktifkan nama DNS dan Aktifkan DNS pribadi hanya untuk titik akhir masuk. Aktifkan nama DNS adalah opsi yang didukung oleh AWS PrivateLink. Dengan menggunakan opsi Aktifkan nama DNS, Anda dapat menggunakan konektivitas pribadi Amazon ke Amazon S3, sambil membuat permintaan ke nama DNS titik akhir publik default. Ketika opsi ini diaktifkan, pelanggan dapat memanfaatkan jalur jaringan biaya terendah yang tersedia untuk aplikasi mereka.

Saat Anda mengaktifkan nama DNS pribadi pada titik akhir antarmuka VPC yang ada atau baru untuk Amazon S3, opsi Aktifkan DNS pribadi hanya untuk titik akhir masuk dipilih secara default. Jika opsi ini dipilih, aplikasi Anda hanya menggunakan titik akhir antarmuka untuk lalu lintas on-premise Anda. Lalu lintas in-VPC ini secara otomatis menggunakan titik akhir gateway berbiaya lebih rendah. Atau, Anda dapat menghapus Aktifkan DNS pribadi hanya untuk titik akhir masuk untuk merutekan semua permintaan S3 melalui titik akhir antarmuka Anda.

Menggunakan AWS CLI

Jika Anda tidak menentukan nilai untuk `PrivateDnsOnlyForInboundResolverEndpoint`, itu akan secara default diatur ke `true`. Namun, sebelum VPC Anda menerapkan pengaturan Anda, VPC melakukan pemeriksaan untuk memastikan bahwa Anda memiliki titik akhir gateway yang ada di VPC. Jika titik akhir gateway ada di VPC, panggilan berhasil. Jika tidak, Anda akan melihat pesan kesalahan berikut ini:

Untuk disetel PrivateDnsOnlyForInboundResolverEndpoint ke true, VPC *vpce_id* harus memiliki titik akhir gateway untuk layanan.

Untuk titik akhir Antarmuka VPC baru

Gunakan atribut `private-dns-enabled` dan `dns-options` untuk mengaktifkan DNS pribadi melalui baris perintah. Opsi `PrivateDnsOnlyForInboundResolverEndpoint` dalam atribut `dns-options` harus diatur ke `true`. Ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws ec2 create-vpc-endpoint \
--region us-east-1 \
--service-name s3-service-name \
--vpc-id client-vpc-id \
--subnet-ids client-subnet-id \
--vpc-endpoint-type Interface  \
--private-dns-enabled  \
--ip-address-type ip-address-type \
--dns-options PrivateDnsOnlyForInboundResolverEndpoint=true \
--security-group-ids client-sg-id
```

Untuk titik akhir VPC yang ada

Jika Anda ingin menggunakan DNS pribadi untuk titik akhir VPC yang ada, gunakan perintah contoh berikut dan ganti dengan informasi Anda *user input placeholders* sendiri.

```
aws ec2 modify-vpc-endpoint \
--region us-east-1 \
--vpc-endpoint-id client-vpc-id \
--private-dns-enabled \
--dns-options PrivateDnsOnlyForInboundResolverEndpoint=false
```

Jika Anda ingin memperbarui titik akhir VPC yang ada untuk mengaktifkan DNS pribadi hanya untuk Inbound Resolver, gunakan contoh berikut dan ganti nilai sampel dengan milik Anda sendiri.

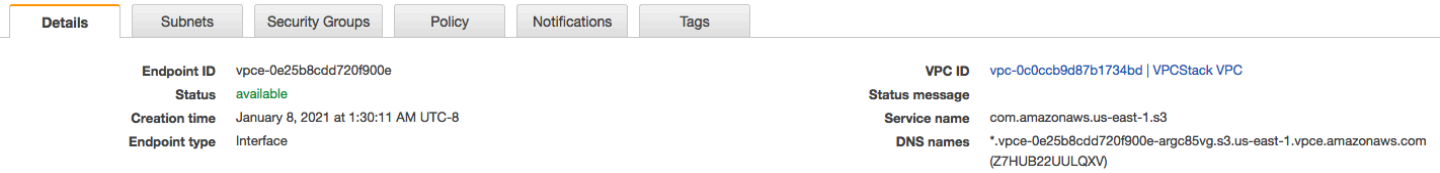
```
aws ec2 modify-vpc-endpoint \
```

```
--region us-east-1 \
--vpc-endpoint-id client-vpc-id \
--private-dns-enabled \
--dns-options PrivateDnsOnlyForInboundResolverEndpoint=true
```

Mengakses bucket, titik akses, serta operasi API Amazon S3 Control dari titik akhir antarmuka S3

Anda dapat menggunakan AWS CLI atau AWS SDKs untuk mengakses bucket, titik akses S3, dan operasi API Kontrol Amazon S3 melalui titik akhir antarmuka S3.

Citra berikut menunjukkan konsol VPC tab Detail, di mana Anda dapat menemukan nama DNS titik akhir VPC. Dalam contoh ini, ID titik akhir VPC (vpce-id) adalah `vpce-0e25b8cdd720f900e` dan Nama DNS adalah `*.vpce-0e25b8cdd720f900e-argc85vg.s3.us-east-1.vpce.amazonaws.com`.



Saat menggunakan nama DNS untuk mengakses sumber daya, ganti * dengan nilai yang sesuai. Nilai yang sesuai untuk digunakan sebagai pengganti * adalah sebagai berikut:

- `bucket`
- `accesspoint`
- `control`

Misalnya, untuk mengakses bucket, gunakan nama DNS seperti ini:

`bucket.vpce-0e25b8cdd720f900e-argc85vg.s3.us-east-1.vpce.amazonaws.com`

Untuk contoh cara menggunakan nama DNS untuk mengakses bucket, titik akses, dan operasi API Kontrol Amazon S3, lihat bagian berikut dari AWS CLI contoh dan AWS Contoh SDK.

Untuk informasi selengkapnya tentang cara melihat nama DNS kustom titik akhir Anda, lihat Melihat konfigurasi nama DNS privat layanan titik akhir dalam Panduan Pengguna VPC.

AWS CLI contoh

Untuk mengakses bucket S3, titik akses S3, atau operasi API Kontrol Amazon S3 melalui titik akhir antarmuka S3 dalam AWS CLI perintah, gunakan dan parameter. `--region` `--endpoint-url`

Contoh: Gunakan URL titik akhir untuk daftar objek dalam bucket Anda

Pada contoh berikut, ganti nama bucket *my-bucket*, Region *us-east-1*, dan nama DNS *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* ID titik akhir VPC dengan informasi Anda sendiri.

```
aws s3 ls s3://my-bucket/ --region us-east-1 --endpoint-url
 https://bucket.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com
```

Contoh: Gunakan URL titik akhir untuk membuat daftar objek dari titik akses

- Metode 1—Menggunakan Amazon Resource Name (ARN) dari titik akses dengan titik akhir titik akses

  Ganti ARN *us-east-1:123456789012:accesspoint/accesspointexamplename*, Wilayah *us-east-1*, dan ID *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* titik akhir VPC dengan informasi Anda sendiri.

  ```
  aws s3api list-objects-v2 --bucket arn:aws:s3:us-east-1:123456789012:accesspoint/
  accesspointexamplename --region us-east-1 --endpoint-url
   https://accesspoint.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com
  ```

  Jika Anda tidak berhasil menjalankan perintah, perbarui AWS CLI ke versi terbaru dan coba lagi. Untuk informasi selengkapnya tentang petunjuk pemutsufiks, lihat Menginstal atau memperbarui versi terbaru AWS CLI dalam AWS Command Line Interface Panduan Pengguna.

- Metode 2—Menggunakan alias titik akses dengan titik akhir bucket regional

  Dalam contoh berikut, ganti alias titik akses *accesspointexamplename-8tyekmigicmhun8n9kwpfur39dnw4use1a-s3alias*, Wilayah *us-east-1*, dan *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* ID titik akhir VPC dengan informasi Anda sendiri.

  ```
  aws s3api list-objects-v2 --
  bucket accesspointexamplename-8tyekmigicmhun8n9kwpfur39dnw4use1a-s3alias
  ```

```
 --region us-east-1 --endpoint-url https://bucket.vpce-1a2b3c4d-5e6f.s3.us-
east-1.vpce.amazonaws.com
```

- Metode 3—Menggunakan alias titik akses dengan titik akhir titik akses

  Pertama, untuk membuat titik akhir S3 dengan bucket yang disertakan sebagai bagian dari nama host, atur gaya pengalamatan `virtual aws s3api` untuk digunakan. Untuk informasi selengkapnya`AWS configure`, lihat Pengaturan konfigurasi dan file kredensial di Panduan AWS Command Line Interface Pengguna.

  ```
  aws configure set default.s3.addressing_style virtual
  ```

  Kemudian, dalam contoh berikut, ganti alias titik akses *accesspointexamplename-8tyekmigicmhun8n9kwpfur39dnw4use1a-s3alias*, Wilayah *us-east-1*, dan *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* ID titik akhir VPC dengan informasi Anda sendiri. Untuk informasi selengkapnya tentang alias titik akses, lihatAlias titik akses.

  ```
  aws s3api list-objects-v2 --
  bucket accesspointexamplename-8tyekmigicmhun8n9kwpfur39dnw4use1a-s3alias --
  region us-east-1 --endpoint-url https://accesspoint.vpce-1a2b3c4d-5e6f.s3.us-
  east-1.vpce.amazonaws.com
  ```

Contoh: Gunakan URL titik akhir untuk mendaftarkan tugas dengan operasi API kontrol S3

Dalam contoh berikut, ganti Wilayah *us-east-1*, ID titik akhir VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com*, dan ID akun *12345678* dengan informasi Anda sendiri.

```
aws s3control --region us-east-1 --endpoint-url
 https://control.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com list-jobs --
account-id 12345678
```

AWS Contoh SDK

Untuk mengakses bucket S3, titik akses S3, atau operasi API Kontrol Amazon S3 melalui titik akhir antarmuka S3 saat menggunakan AWS SDKs, perbarui ke versi terbaru. SDKs Kemudian konfigurasikan klien Anda untuk menggunakan URL titik akhir untuk mengakses bucket, titik akses, atau operasi API Kontrol Amazon S3 melalui titik akhir antarmuka S3.

SDK for Python (Boto3)

Contoh: Gunakan URL titik akhir untuk mengakses bucket S3

Dalam contoh berikut, ganti Wilayah *us-east-1* dan ID titik akhir VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* dengan informasi Anda sendiri.

```
s3_client = session.client(
service_name='s3',
region_name='us-east-1',
endpoint_url='https://bucket.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com'
)
```

Contoh: Gunakan URL titik akhir untuk mengakses titik akses bucket S3

Dalam contoh berikut, ganti Wilayah *us-east-1* dan ID titik akhir VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* dengan informasi Anda sendiri.

```
ap_client = session.client(
service_name='s3',
region_name='us-east-1',
endpoint_url='https://accesspoint.vpce-1a2b3c4d-5e6f.s3.us-
east-1.vpce.amazonaws.com'
)
```

Contoh: Gunakan URL titik akhir untuk mengakses API Kontrol Amazon S3

Dalam contoh berikut, ganti Wilayah *us-east-1* dan ID titik akhir VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* dengan informasi Anda sendiri.

```
control_client = session.client(
service_name='s3control',
region_name='us-east-1',
endpoint_url='https://control.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com'
)
```

SDK for Java 1.x

Contoh: Gunakan URL titik akhir untuk mengakses bucket S3

Dalam contoh berikut, ganti titik akhir VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* dengan informasi Anda sendiri.

```
// bucket client
final AmazonS3 s3 = AmazonS3ClientBuilder.standard().withEndpointConfiguration(
        new AwsClientBuilder.EndpointConfiguration(
                "https://bucket.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com",
                Regions.DEFAULT_REGION.getName()
        )
).build();
List<Bucket> buckets = s3.listBuckets();
```

Contoh: Gunakan URL titik akhir untuk mengakses titik akses bucket S3

Dalam contoh berikut, ganti ID titik akhir VPC dan *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* ARN *us-east-1:123456789012:accesspoint/prod* dengan informasi Anda sendiri.

```
// accesspoint client
final AmazonS3 s3accesspoint =
 AmazonS3ClientBuilder.standard().withEndpointConfiguration(
        new AwsClientBuilder.EndpointConfiguration(
                "https://accesspoint.vpce-1a2b3c4d-5e6f.s3.us-
east-1.vpce.amazonaws.com",
                Regions.DEFAULT_REGION.getName()
        )
).build();
ObjectListing objects = s3accesspoint.listObjects("arn:aws:s3:us-
east-1:123456789012:accesspoint/prod");
```

Contoh: Gunakan URL titik akhir untuk mengakses operasi API Kontrol Amazon S3

Dalam contoh berikut, ganti titik akhir VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* dengan informasi Anda sendiri.

```
// control client
final AWSS3Control s3control =
 AWSS3ControlClient.builder().withEndpointConfiguration(
        new AwsClientBuilder.EndpointConfiguration(
                "https://control.vpce-1a2b3c4d-5e6f.s3.us-
east-1.vpce.amazonaws.com",
```

```
                Regions.DEFAULT_REGION.getName()
        )
).build();
final ListJobsResult jobs = s3control.listJobs(new ListJobsRequest());
```

SDK for Java 2.x

Contoh: Gunakan URL titik akhir untuk mengakses bucket S3

Dalam contoh berikut, ganti ID titik akhir VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* dan Wilayah *Region.US_EAST_1* dengan informasi Anda sendiri.

```
// bucket client
Region region = Region.US_EAST_1;
s3Client = S3Client.builder().region(region)

 .endpointOverride(URI.create("https://bucket.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com"))
                   .build()
```

Contoh: Gunakan URL titik akhir untuk mengakses titik akses bucket S3

Dalam contoh berikut, ganti ID titik akhir VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* dan Wilayah *Region.US_EAST_1* dengan informasi Anda sendiri.

```
// accesspoint client
Region region = Region.US_EAST_1;
s3Client = S3Client.builder().region(region)

 .endpointOverride(URI.create("https://accesspoint.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com"))
                   .build()
```

Contoh: Gunakan URL titik akhir untuk mengakses API Kontrol Amazon S3

Dalam contoh berikut, ganti ID titik akhir VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* dan Wilayah *Region.US_EAST_1* dengan informasi Anda sendiri.

```
// control client
Region region = Region.US_EAST_1;
s3ControlClient = S3ControlClient.builder().region(region)

 .endpointOverride(URI.create("https://control.vpce-1a2b3c4d-5e6f.s3.us-
east-1.vpce.amazonaws.com"))
                                  .build()
```

Memperbarui konfigurasi DNS on-premise

Saat menggunakan nama DNS kustom titik akhir untuk mengakses titik akhir antarmuka untuk Amazon S3, Anda tidak perlu memperbarui penyelesai DNS on-premise Anda. Anda dapat menyelesaikan nama DNS titik akhir kustom dengan alamat IP privat titik akhir antarmuka dari domain Amazon S3 DNS publik.

Menggunakan titik akhir antarmuka untuk mengakses Amazon S3 tanpa titik akhir gateway atau gateway internet di VPC

Titik akhir antarmuka di VPC Anda dapat merutekan kedua aplikasi di VPC dan aplikasi on-premise ke Amazon S3 melalui jaringan Amazon, seperti yang digambarkan dalam diagram berikut.

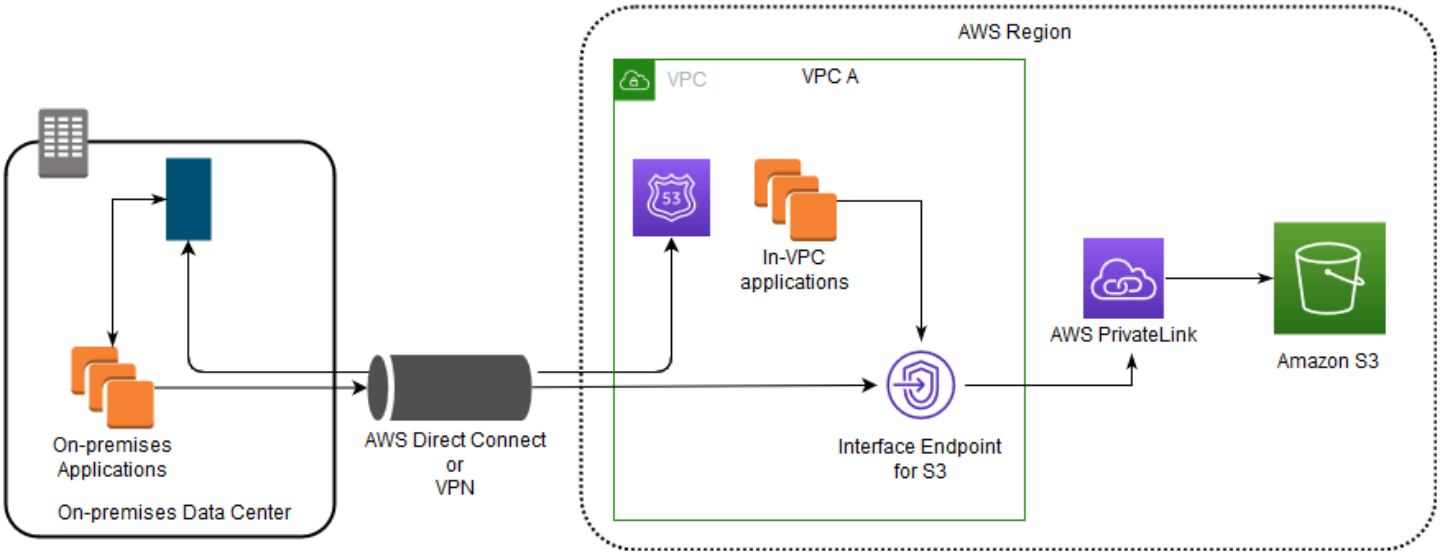


Diagram ini menggambarkan hal sebagai berikut:

- Jaringan lokal Anda menggunakan AWS Direct Connect atau menyambung AWS VPN ke VPC A.
- Aplikasi Anda on-premise dan di VPC A menggunakan nama DNS titik akhir kustom untuk mengakses Amazon S3 melalui titik akhir antarmuka S3.

- Aplikasi lokal mengirim data ke titik akhir antarmuka di VPC melalui AWS Direct Connect (atau). AWS VPN AWS PrivateLink memindahkan data dari titik akhir antarmuka ke Amazon S3 melalui AWS jaringan.
- Aplikasi in-vPC juga mengirim lalu lintas ke titik akhir antarmuka. AWS PrivateLink memindahkan data dari titik akhir antarmuka ke Amazon S3 melalui AWS jaringan.

Menggunakan titik akhir gateway dan antarmuka titik akhir bersama-sama dalam VPC yang sama untuk mengakses Amazon S3

Anda dapat membuat titik akhir antarmuka dan mempertahankan titik akhir gateway yang ada di VPC yang sama, seperti yang ditunjukkan diagram berikut. Dengan mengambil pendekatan ini, Anda mengizinkan aplikasi dalam VPC untuk terus mengakses Amazon S3 melalui titik akhir gateway, yang tidak ditagih. Kemudian, hanya aplikasi on-premise Anda yang akan menggunakan titik akhir antarmuka untuk mengakses Amazon S3. Untuk mengakses S3 dengan cara ini, Anda harus memperbarui aplikasi lokal Anda untuk menggunakan nama DNS kustom titik akhir untuk Amazon S3.

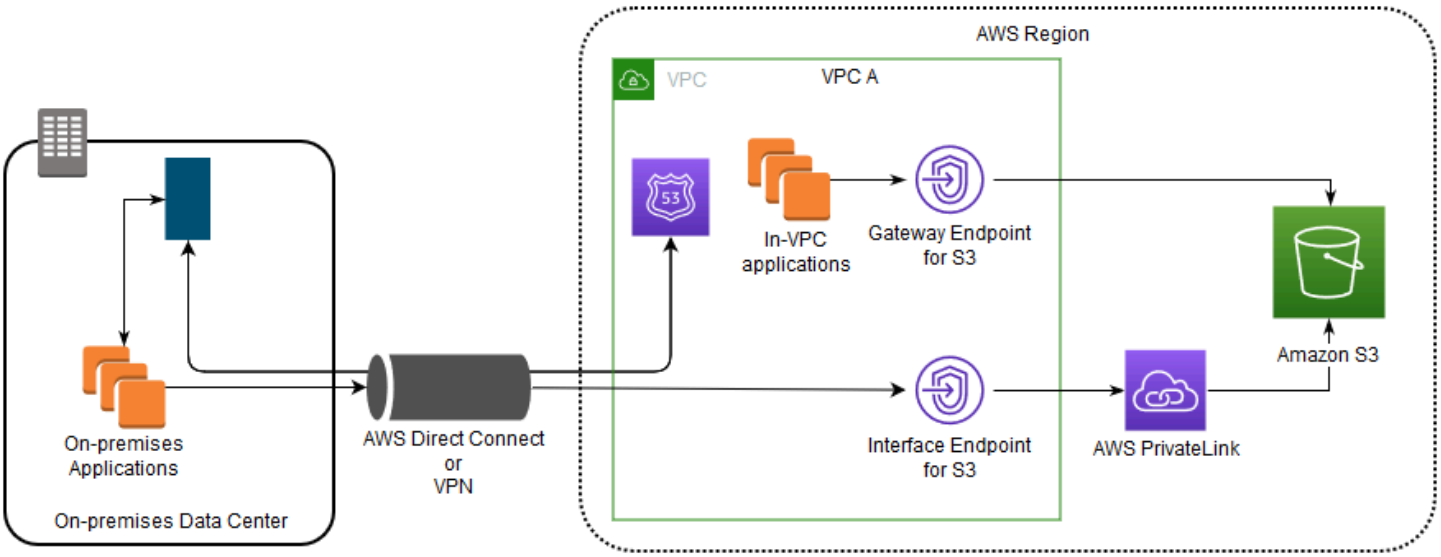


Diagram ini menggambarkan hal sebagai berikut:

- Aplikasi lokal menggunakan nama DNS khusus titik akhir untuk mengirim data ke titik akhir antarmuka dalam VPC melalui (atau). AWS Direct Connect AWS VPN AWS PrivateLink memindahkan data dari titik akhir antarmuka ke Amazon S3 melalui AWS jaringan.
- Menggunakan nama Amazon S3 Regional default, aplikasi in-VPC mengirim data ke titik akhir gateway yang terhubung ke Amazon S3 melalui jaringan. AWS

Untuk informasi selengkapnya tentang titik akhir gateway, lihat Gateway titik akhir VPC di Panduan Pengguna VPC.

Membuat kebijakan titik akhir VPC untuk Amazon S3

Anda dapat melampirkan kebijakan titik akhir ke titik akhir VPC yang mengendalikan akses ke Amazon S3. Kebijakan titik akhir mencantumkan informasi berikut:

- Prinsip AWS Identity and Access Management (IAM) yang dapat melakukan tindakan
- Tindakan-tindakan yang dapat dilakukan
- Sumber daya yang padanya tindakan dapat dilakukan

Anda juga dapat menggunakan kebijakan bucket Amazon S3 untuk membatasi akses ke bucket tertentu dari titik akhir VPC tertentu menggunakan syarat `aws:sourceVpce` dalam kebijakan bucket Anda. Contoh berikut menunjukkan kebijakan yang membatasi akses ke bucket atau titik akhir.

Topik

- Contoh: Membatasi Akses ke bucket kustom dari titik akhir VPC
- Contoh: Membatasi akses ke bucket kustom di akun kustom dari titik akhir VPC
- Contoh: Membatasi Akses ke titik akhir VPC kustom di kebijakan bucket S3

Contoh: Membatasi Akses ke bucket kustom dari titik akhir VPC

Anda dapat membuat kebijakan titik akhir yang membatasi akses ke bucket S3 Amazon tertentu saja. Jenis kebijakan ini berguna jika Anda memiliki kebijakan lain Layanan AWS di VPC Anda yang menggunakan bucket. Kebijakan bucket berikut hanya membatasi akses ke file *amzn-s3-demo-bucket1*. Untuk menggunakan kebijakan titik akhir ini, ganti *amzn-s3-demo-bucket1* dengan nama bucket Anda.

```
{
  "Version": "2012-10-17",
  "Id": "Policy1415115909151",
  "Statement": [
    { "Sid": "Access-to-specific-bucket-only",
      "Principal": "*",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
```

```
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket1",
                   "arn:aws:s3:::amzn-s3-demo-bucket1/*"]
    }
  ]
}
```

Contoh: Membatasi akses ke bucket kustom di akun kustom dari titik akhir VPC

Anda dapat membuat kebijakan titik akhir yang membatasi akses hanya ke bucket S3 di bucket tertentu. Akun AWS Untuk mencegah klien dalam VPC Anda mengakses bucket yang tidak Anda miliki, gunakan pernyataan berikut ini di kebijakan titik akhir Anda. Pernyataan contoh berikut membuat kebijakan yang membatasi akses ke sumber daya yang dimiliki oleh satu Akun AWS ID, *111122223333*.

```
{
  "Statement": [
    {
      "Sid": "Access-to-bucket-in-specific-account-only",
      "Principal": "*",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Effect": "Deny",
      "Resource": "arn:aws:s3:::*",
      "Condition": {
        "StringNotEquals": {
          "aws:ResourceAccount": "111122223333"
        }
      }
    }
  ]
}
```

> **Note**
>
> Untuk menentukan Akun AWS ID sumber daya yang diakses, Anda dapat menggunakan salah satu `aws:ResourceAccount` atau `s3:ResourceAccount` kunci dalam kebijakan IAM Anda. Namun, ketahuilah bahwa beberapa Layanan AWS mengandalkan akses ke bucket yang AWS dikelola. Oleh karena itu, menggunakan kunci `aws:ResourceAccount`

atau `s3:ResourceAccount` dalam kebijakan IAM Anda juga dapat memengaruhi akses ke sumber daya ini.

Contoh: Membatasi Akses ke titik akhir VPC kustom di kebijakan bucket S3

Contoh: Membatasi Akses ke titik akhir VPC kustom di kebijakan bucket S3

Kebijakan bucket Amazon S3 berikut memungkinkan akses ke bucket tertentu, *amzn-s3-demo-bucket2*, hanya dari titik akhir VPC *vpce-1a2b3c4d*. Kebijakan ini menolak semua akses ke bucket jika titik akhir yang ditentukan tidak digunakan. `aws:sourceVpce`Kondisi menentukan titik akhir dan tidak memerlukan Amazon Resource Name (ARN) untuk sumber daya titik akhir VPC, tetapi hanya ID titik akhir. Untuk menggunakan kebijakan bucket ini, ganti *amzn-s3-demo-bucket2* dan *vpce-1a2b3c4d* dengan nama bucket dan titik akhir Anda.

Important

- Saat menerapkan kebijakan bucket Amazon S3 berikut untuk membatasi akses hanya ke titik akhir VPC tertentu, Anda dapat memblokir akses Anda ke bucket tanpa bermaksud melakukannya. Kebijakan bucket yang dimaksudkan untuk secara kustom membatasi akses bucket ke koneksi yang berasal dari titik akhir VPC Anda dapat memblokir semua koneksi ke bucket. Untuk informasi tentang cara mengatasi masalah ini, lihat Kebijakan bucket saya memiliki ID titik akhir VPC atau VPC yang salah. Bagaimana saya dapat memperbaiki kebijakan tersebut sehingga saya dapat mengakses bucket? dalam Dukungan Pusat Pengetahuan.
- Sebelum menggunakan kebijakan contoh berikut ini, ganti ID titik akhir VPC dengan nilai yang sesuai untuk kasus penggunaan Anda. Jika tidak, Anda tidak akan dapat mengakses bucket Anda.
- Kebijakan ini nonaktifkan akses konsol ke bucket tertentu, karena permintaan konsol tidak berasal dari titik akhir VPC tertentu.

```
{
  "Version": "2012-10-17",
  "Id": "Policy1415115909152",
  "Statement": [
    { "Sid": "Access-to-specific-VPCE-only",
```

```
      "Principal": "*",
      "Action": "s3:*",
      "Effect": "Deny",
      "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket2",
                   "arn:aws:s3:::amzn-s3-demo-bucket2/*"],
      "Condition": {"StringNotEquals": {"aws:sourceVpce": "vpce-1a2b3c4d"}}
    }
  ]
}
```

Untuk contoh kebijakan lainnya, lihat Titik akhir untuk Amazon S3 di Panduan Pengguna VPC.

Untuk informasi selengkapnya tentang konektivitas VPC, lihat opsi konektivitas di AWS whitepaper Opsi Network-to-VPC Konektivitas Amazon Virtual Private Cloud.

Validasi kepatuhan untuk Amazon S3

Keamanan dan kepatuhan Amazon S3 dinilai oleh auditor pihak ketiga sebagai bagian dari beberapa program AWS kepatuhan, termasuk yang berikut:

- Kontrol Sistem dan Organisasi (SOC)
- Standar Keamanan Data Industri Kartu Pembayaran (PCI DSS)
- Program Manajemen Risiko dan Otorisasi Federal (FedRAMP)
- Undang-Undang Akuntabilitas dan Portabilitas Asuransi Kesehatan (HIPAA)

AWS menyediakan daftar AWS layanan yang sering diperbarui dalam lingkup program kepatuhan khusus di AWS Layanan dalam Lingkup oleh Program Kepatuhan.

Laporan audit pihak ketiga tersedia untuk Anda unduh AWS Artifact. Untuk informasi selengkapnya, lihat Mengunduh Laporan di AWS Artifak.

Untuk informasi selengkapnya tentang program AWS kepatuhan, lihat Program AWS Kepatuhan.

Tanggung jawab kepatuhan Anda saat menggunakan Amazon S3 ditentukan oleh sensitivitas data Anda, tujuan kepatuhan organisasi Anda, serta undang-undang dan peraturan yang berlaku. Jika penggunaan Amazon S3 Anda tunduk pada kepatuhan standar seperti HIPAA, PCI, atau FedRAMP, AWS menyediakan sumber daya untuk membantu:

- Panduan Memulai Cepat Keamanan dan Kepatuhan yang membahas pertimbangan arsitektur dan langkah-langkah untuk menerapkan lingkungan dasar yang berfokus pada keamanan dan kepatuhan. AWS
- Arsitektur untuk Keamanan dan Kepatuhan HIPAA menguraikan bagaimana perusahaan menggunakan AWS untuk membantu mereka memenuhi persyaratan HIPAA.
- AWS Sumber Daya Kepatuhan menyediakan beberapa buku kerja dan panduan berbeda yang mungkin berlaku untuk industri dan lokasi Anda.
- AWS Config dapat digunakan untuk menilai seberapa baik konfigurasi sumber daya Anda telah mematuhi praktik internal, pedoman industri, dan peraturan yang berlaku.
- AWS Security Hub memberi Anda pandangan komprehensif tentang status keamanan Anda di dalamnya AWS dan membantu Anda memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik.

- Mengunci objek dengan Object Lock dapat membantu Anda memenuhi persyaratan teknis dari regulator layanan keuangan (seperti SEC, FINRA, dan CFTC) yang memerlukan penyimpanan data “tulis sekali, baca banyak” (WORM) untuk jenis informasi pembukuan dan pencatatan tertentu.
- Membuat katalog dan menganalisis data Anda dengan S3 Inventory dapat membantu Anda mengaudit dan melaporkan replikasi dan status enkripsi objek Anda untuk kebutuhan bisnis, kepatuhan, dan regulasi.

Ketahanan dalam Amazon S3

Infrastruktur AWS global dibangun di sekitar Wilayah dan Zona Ketersediaan. Wilayah AWS menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Zona Ketersediaan ini menawarkan cara efektif untuk merancang dan mengoperasikan aplikasi dan basis data. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur pusat data tunggal atau multi tradisional. Jika Anda secara khusus perlu mereplikasi data Anda pada jarak geografis yang lebih besar, Anda dapat menggunakanMereplikasi objek di dalam dan di seluruh Wilayah, yang memungkinkan penyalinan objek secara otomatis dan asinkron di seluruh ember dalam berbagai jenis. Wilayah AWS

Masing-masing Wilayah AWS memiliki beberapa Availability Zone. Anda dapat menerapkan aplikasi Anda di beberapa Zona Ketersediaan di Wilayah yang sama untuk toleransi kesalahan dan latensi rendah. Zona Ketersediaan saling terhubung satu sama lain dengan jaringan serat optik pribadi yang cepat, memungkinkan Anda untuk merancang aplikasi secara mudah yang menerapkan pengalihan otomatis antara Zona Ketersediaan tanpa gangguan.

Untuk informasi selengkapnya tentang Wilayah AWS dan Availability Zone, lihat Infrastruktur AWS Global.

Selain infrastruktur AWS global, Amazon S3 menawarkan beberapa fitur untuk membantu mendukung ketahanan data dan kebutuhan cadangan Anda.

Konfigurasi siklus hidup

> Konfigurasi siklus aktif adalah serangkaian aturan yang menentukan tindakan yang diterapkan Amazon S3 pada sekelompok objek. Dengan aturan konfigurasi siklus aktif, Anda dapat memberi tahu Amazon S3 untuk melakukan transisi objek ke kelas penyimpanan yang lebih murah, mengarsipkan, atau menghapusnya. Untuk informasi selengkapnya, lihat Mengelola siklus hidup objek.

Penentuan Versi

> Penentuan Versi adalah cara menyimpan beberapa varian objek dalam bucket yang sama. Anda dapat menggunakan Penentuan Versi untuk menyimpan, mengambil, dan memulihkan setiap versi dari setiap objek yang disimpan dalam bucket Amazon S3. Dengan Penentuan Versi, Anda dapat dengan mudah memulihkan dari tindakan pengguna yang tidak diinginkan, serta kegagalan aplikasi. Untuk informasi selengkapnya, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.

Kunci Objek S3

Anda dapat menggunakan Kunci Objek S3 untuk menyimpan objek menggunakan model tulis-sekali-baca-banyak (WORM). Dengan Kunci Objek S3, Anda dapat mencegah objek agar tidak dihapus atau ditimpa selama jangka waktu tertentu atau tanpa batas waktu. Kunci Objek S3 memungkinkan Anda memenuhi persyaratan peraturan yang memerlukan penyimpanan WORM atau hanya untuk menambahkan lapisan perlindungan tambahan terhadap perubahan dan penghapusan objek. Untuk informasi selengkapnya, lihat Mengunci objek dengan Object Lock.

Kelas penyimpanan

Amazon S3 menawarkan serangkaian kelas penyimpanan untuk dipilih tergantung pada persyaratan beban kerja Anda. Kelas penyimpanan S3 Standard-IA dan S3 One Zone-IA dirancang untuk data yang Anda akses sekitar sebulan sekali dan membutuhkan akses milidetik. Kelas penyimpanan S3 Glacier Instant Retrieval dirancang untuk data arsip berumur panjang yang diakses dengan akses milidetik yang Anda akses sekitar seperempat sekali. Untuk data arsip yang tidak memerlukan akses langsung, seperti backup, Anda dapat menggunakan kelas penyimpanan S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive. Untuk informasi selengkapnya, lihat Memahami dan mengelola kelas penyimpanan Amazon S3.

Praktik keamanan terbaik berikut juga menangani ketahanan:

- Enable versioning
- Consider Amazon S3 cross-region replication
- Identify and audit all your Amazon S3 buckets

Enkripsi cadangan Amazon S3

Jika Anda menyimpan cadangan menggunakan Amazon S3, enkripsi cadangan Anda tergantung pada konfigurasi bucket tersebut. Amazon S3 menyediakan cara untuk menetapkan perilaku enkripsi default untuk bucket S3. Anda dapat menetapkan enkripsi default pada bucket sehingga semua objek dienkripsi saat disimpan dalam bucket. Enkripsi default mendukung kunci yang disimpan di AWS KMS (SSE-KMS). Untuk informasi selengkapnya, lihat Mengatur perilaku enkripsi sisi server default untuk bucket Amazon S3.

Untuk informasi lebih lanjut tentang Penentuan Versi dan Kunci Objek, lihat topik berikut: Mempertahankan beberapa versi objek dengan S3 Versioning Mengunci objek dengan Object Lock

Keamanan infrastruktur di Amazon S3

Sebagai layanan terkelola, Amazon S3 dilindungi oleh prosedur keamanan jaringan AWS global yang dijelaskan dalam pilar keamanan Kerangka Well-Architected AWS .

Akses ke Amazon S3 melalui jaringan melalui AWS dipublikasikan. APIs Klien harus mendukung Keamanan Lapisan Pengangkutan (TLS) 1.2. Kami merekomendasikan juga mendukung TLS 1.3. (Untuk informasi selengkapnya tentang rekomendasi ini, lihat Koneksi AWS cloud yang lebih cepat dengan TLS 1.3 di Blog AWS Keamanan.)

> **Note**
>
> TLS 1.3 didukung di semua titik akhir S3, kecuali untuk Amazon AWS PrivateLink S3 dan Titik Akses Multi-Wilayah.

Klien juga harus mendukung suite cipher dengan perfect forward secrecy (PFS) seperti Ephemeral Diffie-Hellman (DHE) atau Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Selain itu, permintaan harus ditandatangani menggunakan AWS Signature V4 atau AWS Signature V2, yang memerlukan kredensyal yang valid untuk diberikan.

Ini dapat APIs dipanggil dari lokasi jaringan mana pun. Namun, Amazon S3 mendukung kebijakan akses berbasis sumber daya, yang dapat mencakup pembatasan berdasarkan alamat IP sumber. Anda juga dapat menggunakan kebijakan bucket Amazon S3 untuk mengontrol akses ke bucket dari titik akhir virtual private cloud (VPC) tertentu, atau spesifik. VPCs Secara efektif, ini mengisolasi akses jaringan ke bucket Amazon S3 tertentu hanya dari VPC tertentu dalam jaringan. AWS Untuk informasi selengkapnya, lihat Mengendalikan akses dari titik akhir VPC dengan kebijakan bucket.

Praktik terbaik keamanan berikut juga membahas keamanan infrastruktur di Amazon S3:

- Consider VPC endpoints for Amazon S3 access
- Identify and audit all your Amazon S3 buckets

Analisis konfigurasi dan kerentanan di Amazon S3

AWS menangani tugas-tugas keamanan dasar seperti sistem operasi tamu (OS) dan patching database, konfigurasi firewall, dan pemulihan bencana. Prosedur ini telah ditinjau dan disertifikasi oleh pihak ketiga yang sesuai. Untuk detail selengkapnya, lihat sumber daya berikut:

- Validasi kepatuhan untuk Amazon S3
- Model Tanggung Jawab Bersama
- Amazon Web Services: Gambaran Umum Proses Keamanan

Praktik terbaik keamanan berikut juga membahas konfigurasi dan analisis kerentanan di Amazon S3:

- Identify and audit all your Amazon S3 buckets
- Aktifkan AWS Config

Manajemen akses

Amazon S3 menyediakan berbagai alat manajemen akses. Berikut ini adalah daftar fitur dan alat ini. Anda tidak memerlukan semua alat manajemen akses ini, tetapi Anda harus menggunakan satu atau lebih untuk memberikan akses ke bucket Amazon S3, objek, dan lainnya. Sumber daya S3 Aplikasi yang tepat dari alat-alat ini dapat membantu memastikan bahwa sumber daya Anda hanya dapat diakses oleh pengguna yang dituju.

Alat manajemen akses yang paling umum digunakan adalah kebijakan akses. Kebijakan akses dapat berupa kebijakan berbasis sumber daya yang dilampirkan ke AWS sumber daya, seperti kebijakan bucket untuk bucket. Kebijakan akses juga dapat berupa kebijakan berbasis identitas yang dilampirkan pada identitas AWS Identity and Access Management (IAM), seperti pengguna, grup, atau peran IAM. Kebijakan akses menjelaskan siapa yang memiliki akses ke hal-hal apa. Tulis kebijakan akses untuk memberikan Akun AWS izin kepada pengguna, grup, dan peran IAM untuk melakukan operasi pada sumber daya. Misalnya, Anda dapat memberikan `PUT Object` izin kepada yang lain Akun AWS sehingga akun lain dapat mengunggah objek ke bucket Anda.

Berikut ini adalah alat manajemen akses yang tersedia di Amazon S3. Untuk panduan yang lebih komprehensif tentang kontrol akses Amazon S3, lihat. Kontrol akses di Amazon S3

Kebijakan bucket

Kebijakan bucket Amazon S3 adalah kebijakan berbasis sumber daya berformat JSON AWS Identity and Access Management (IAM) yang dilampirkan ke bucket tertentu. Gunakan kebijakan bucket untuk memberikan izin identitas lain Akun AWS atau IAM untuk bucket dan objek di dalamnya. Banyak kasus penggunaan manajemen akses S3 dapat dipenuhi dengan menggunakan kebijakan bucket. Dengan kebijakan bucket, Anda dapat mempersonalisasi akses bucket untuk membantu memastikan bahwa hanya identitas yang telah Anda setujui yang dapat mengakses sumber daya dan melakukan tindakan di dalamnya. Untuk informasi selengkapnya, lihat Kebijakan bucket untuk Amazon S3.

Kebijakan berbasis identitas

Kebijakan pengguna berbasis identitas atau IAM adalah jenis kebijakan AWS Identity and Access Management (IAM). Kebijakan berbasis identitas adalah kebijakan berformat JSON yang dilampirkan ke pengguna, grup, atau peran IAM di akun Anda. AWS Anda dapat menggunakan kebijakan berbasis identitas untuk memberikan akses identitas IAM ke bucket atau objek Anda. Anda dapat membuat pengguna, grup, dan peran IAM di akun Anda dan melampirkan kebijakan akses kepada mereka. Anda kemudian dapat memberikan akses ke AWS sumber daya, termasuk sumber daya Amazon S3. Untuk informasi selengkapnya, lihat Kebijakan berbasis identitas untuk Amazon S3.

Pemberian Akses S3

Gunakan Hibah Akses S3 untuk membuat hibah akses ke data Amazon S3 Anda untuk kedua identitas di direktori identitas perusahaan, sepertiActive Directory, dan identitas (IAM). AWS Identity and Access Management S3 Access Grants membantu Anda mengelola izin data dalam skala besar. Selain itu, S3 Access Grants mencatat identitas pengguna akhir dan aplikasi yang digunakan untuk mengakses data S3 di. AWS CloudTrail Ini memberikan riwayat audit terperinci hingga identitas pengguna akhir untuk semua akses ke data di bucket S3 Anda. Untuk informasi selengkapnya, lihat Mengelola akses dengan S3 Access Grants.

Titik Akses

Titik Akses Amazon S3 menyederhanakan pengelolaan akses data dalam skala besar untuk aplikasi yang menggunakan kumpulan data bersama di S3. Access Points diberi nama endpoint jaringan yang dilampirkan ke bucket. Anda dapat menggunakan titik akses untuk melakukan operasi objek S3 dalam skala besar, seperti mengunggah dan mengambil objek. Bucket dapat memiliki hingga 10.000 titik akses yang terpasang, dan untuk setiap titik akses, Anda dapat menerapkan izin dan kontrol jaringan yang berbeda untuk memberi Anda kontrol terperinci atas akses ke objek S3 Anda. Poin Akses S3 dapat dikaitkan dengan bucket di akun yang sama atau di akun tepercaya lainnya. Kebijakan Access Points adalah kebijakan berbasis sumber daya yang dievaluasi bersama dengan

kebijakan bucket yang mendasarinya. Untuk informasi selengkapnya, lihat Mengelola akses ke kumpulan data bersama dengan titik akses.

Daftar kontrol akses (ACL)

ACL adalah daftar hibah yang mengidentifikasi penerima hibah dan izin yang diberikan. ACLs memberikan izin baca atau tulis dasar kepada yang lain Akun AWS. ACLs gunakan skema XML-khusus Amazon S3. ACL adalah jenis kebijakan AWS Identity and Access Management (IAM). Objek ACL digunakan untuk mengelola akses ke objek, dan bucket ACL digunakan untuk mengelola akses ke bucket. Dengan kebijakan bucket, ada satu kebijakan untuk seluruh bucket, tetapi objek ACLs ditentukan untuk setiap objek. Kami menyarankan agar Anda tetap ACLs dimatikan, kecuali dalam keadaan yang tidak biasa di mana Anda harus mengontrol akses secara individual untuk setiap objek. Untuk informasi selengkapnya tentang penggunaan ACLs, lihatMengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.

> **Warning**
>
> Mayoritas kasus penggunaan modern di Amazon S3 tidak memerlukan penggunaan. ACLs

Kepemilikan Objek

Untuk mengelola akses ke objek Anda, Anda harus menjadi pemilik objek. Anda dapat menggunakan setelan tingkat ember Kepemilikan Objek untuk mengontrol kepemilikan objek yang diunggah ke bucket. Juga, gunakan Object Ownership untuk mengaktifkan ACLs. Secara default, Kepemilikan Objek diatur ke setelan diberlakukan pemilik Bucket dan ACLs semuanya dimatikan. Saat ACLs dimatikan, pemilik bucket memiliki semua objek di bucket dan secara eksklusif mengelola akses ke data. Untuk mengelola akses, pemilik bucket menggunakan kebijakan atau alat manajemen akses lainnya, tidak termasuk. ACLs Untuk informasi selengkapnya, lihat Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.

Untuk panduan yang lebih komprehensif tentang kontrol akses Amazon S3 dan praktik terbaik tambahan, lihat. Kontrol akses di Amazon S3

Perlindungan data di Amazon S3

Selain ketahanan yang ditawarkan oleh infrastruktur AWS global, Amazon S3 menawarkan sejumlah fitur untuk membantu melindungi data Anda dari penghapusan yang tidak disengaja atau kegagalan Regional.

Replikasi S3

Anda dapat menggunakan replikasi langsung untuk mengaktifkan penyalinan objek secara otomatis dan asinkron di seluruh bucket Amazon S3. Bucket yang dikonfigurasi untuk replikasi objek dapat dimiliki oleh akun yang sama Akun AWS atau berbeda. Anda dapat mereplikasi objek ke satu bucket tujuan atau ke beberapa bucket tujuan. Bucket tujuan bisa berbeda Wilayah AWS atau dalam Wilayah yang sama dengan ember sumber. Untuk mengaktifkan kontrol failover, Anda dapat mengonfigurasi replikasi menjadi dua arah (dua arah) sehingga bucket sumber dan tujuan dapat tetap sinkron selama kegagalan Regional. Untuk informasi selengkapnya, lihat the section called “Mereplikasi objek di dalam dan di seluruh Wilayah”.

Titik Akses Multi-Wilayah dan kontrol failover

Titik Akses Multi-Wilayah Amazon S3 menyediakan titik akhir global yang dapat digunakan aplikasi untuk memenuhi permintaan dari bucket S3 yang terletak di beberapa Wilayah AWS. Anda dapat menggunakan Titik Akses Multi-Wilayah untuk membangun aplikasi multi-Wilayah dengan arsitektur yang sama dengan yang digunakan di satu Wilayah, dan kemudian menjalankan aplikasi tersebut di mana saja di seluruh dunia. Alih-alih mengirimkan permintaan melalui internet publik yang padat, Titik Akses Multi-Wilayah menyediakan ketahanan jaringan bawaan dengan percepatan permintaan berbasis internet ke Amazon S3. Permintaan aplikasi yang dibuat ke titik akhir global Titik Akses Multi-Wilayah menggunakan AWS Global Accelerator untuk secara otomatis merutekan jaringan AWS global ke bucket S3 terdekat dengan status perutean aktif. Untuk informasi selengkapnya tentang Titik Akses Multi-Wilayah, lihatthe section called “Mengelola lalu lintas multi-wilayah ”.

Dengan kontrol failover Titik Akses Multi-Wilayah Amazon S3, Anda dapat mempertahankan kelangsungan bisnis selama terjadinya gangguan lalu lintas Wilayah, sekaligus memberikan arsitektur multi-Wilayah pada aplikasi Anda untuk memenuhi kebutuhan kepatuhan dan redundansi. Jika lalu lintas Regional Anda terganggu, Anda dapat menggunakan kontrol failover Titik Akses Multi-Wilayah untuk memilih mana Wilayah AWS di belakang Titik Akses Multi-Wilayah Amazon S3 yang akan memproses permintaan akses data dan penyimpanan.

Untuk mendukung failover, Anda dapat mengatur Titik Akses Multi-Wilayah dalam konfigurasi aktif-pasif, dengan lalu lintas yang berjalan ke Wilayah aktif selama kondisi normal, dan Wilayah pasif dalam keadaan siaga untuk failover. Jika Anda mengaktifkan Replikasi Lintas Wilayah (CRR) S3 dengan aturan replikasi dua arah, Anda bisa menjaga agar bucket tetap tersinkronisasi selama failover. Untuk informasi selengkapnya tentang kontrol failover, lihatthe section called “Konfigurasi failover”.

Penentuan Versi S3

Penentuan Versi adalah cara menyimpan beberapa varian objek dalam bucket yang sama. Anda dapat menggunakan Penentuan Versi untuk menyimpan, mengambil, dan memulihkan setiap versi dari setiap objek yang disimpan dalam bucket Amazon S3. Dengan Penentuan Versi, Anda dapat dengan mudah memulihkan dari tindakan pengguna yang tidak diinginkan, serta kegagalan aplikasi. Untuk informasi selengkapnya, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.

Kunci Objek S3

Anda dapat menggunakan Kunci Objek S3 untuk menyimpan objek menggunakan model tulis-sekali-baca-banyak (WORM). Dengan Kunci Objek S3, Anda dapat mencegah objek agar tidak dihapus atau ditimpa selama jangka waktu tertentu atau tanpa batas waktu. Kunci Objek S3 memungkinkan Anda memenuhi persyaratan peraturan yang memerlukan penyimpanan WORM atau hanya untuk menambahkan lapisan perlindungan tambahan terhadap perubahan dan penghapusan objek. Untuk informasi selengkapnya, lihat Mengunci objek dengan Object Lock.

AWS Backup

Amazon S3 terintegrasi secara native dengan AWS Backup, layanan berbasis kebijakan yang dikelola sepenuhnya yang dapat Anda gunakan untuk menentukan kebijakan pencadangan secara terpusat guna melindungi data Anda di Amazon S3. Setelah Anda menentukan kebijakan pencadangan dan menetapkan sumber daya Amazon S3 ke kebijakan AWS Backup , mengotomatiskan pembuatan cadangan Amazon S3 dan menyimpan cadangan dengan aman di brankas cadangan terenkripsi yang Anda tetapkan dalam paket cadangan. Untuk informasi selengkapnya, lihat the section called “Mencadangkan data Anda”.

Untuk tutorial tentang penggunaan beberapa fitur ini bersama-sama untuk melindungi data Anda, lihat Tutorial: Melindungi data di Amazon S3 terhadap penghapusan yang tidak disengaja atau bug aplikasi menggunakan Pembuatan Versi S3, Kunci Objek S3, dan Replikasi S3.

> **Important**
>
> Selain menggunakan fitur sebelumnya untuk melindungi data Anda, kami sarankan untuk meninjau rekomendasi di. the section called “Praktik terbaik keamanan”

Topik

- Mereplikasi objek di dalam dan di seluruh Wilayah
- Mengelola lalu lintas Multi-wilayah dengan Titik Akses Multi-Wilayah
- Mempertahankan beberapa versi objek dengan S3 Versioning
- Mengunci objek dengan Object Lock
- Mencadangkan data Amazon S3 Anda

Mereplikasi objek di dalam dan di seluruh Wilayah

Anda dapat menggunakan replikasi untuk mengaktifkan penyalinan objek secara otomatis dan asinkron di seluruh bucket Amazon S3. Bucket yang dikonfigurasi untuk replikasi objek dapat dimiliki oleh Akun AWS yang sama atau oleh akun yang berbeda. Anda dapat mereplikasi objek ke satu bucket tujuan atau ke beberapa bucket tujuan. Bucket tujuan bisa berbeda Wilayah AWS atau dalam Wilayah yang sama dengan ember sumber.

Ada dua jenis replikasi: replikasi langsung dan replikasi sesuai permintaan.

- Replikasi langsung — Untuk secara otomatis mereplikasi objek baru dan yang diperbarui saat ditulis ke keranjang sumber, gunakan replikasi langsung. Replikasi langsung tidak mereplikasi objek apa pun yang ada di bucket sebelum Anda mengatur replikasi. Untuk mereplikasi objek yang ada sebelum Anda mengatur replikasi, gunakan replikasi sesuai permintaan.
- Replikasi sesuai permintaan — Untuk mereplikasi objek yang ada dari bucket sumber ke satu atau beberapa bucket tujuan sesuai permintaan, gunakan Replikasi Batch S3. Untuk informasi selengkapnya tentang mereplikasi objek yang ada, lihat Kapan menggunakan Replikasi Batch S3.

Ada dua bentuk replikasi hidup: Cross-Region Replication (CRR) dan Same-Region Replication (SRR).

- Replikasi Lintas Wilayah (CRR) - Anda dapat menggunakan CRR untuk mereplikasi objek di bucket Amazon S3 secara berbeda. Wilayah AWS Untuk informasi lebih lanjut tentang CRR, lihatthe section called “Kapan menggunakan Replikasi Lintas-Wilayah”.
- Same-Region Replication (SRR) - Anda dapat menggunakan SRR untuk menyalin objek di bucket Amazon S3 secara bersamaan. Wilayah AWS Untuk informasi lebih lanjut tentang SRR, lihatthe section called “Kapan menggunakan Replikasi Lintas-Wilayah”.

Topik

- Mengapa menggunakan replikasi?
- Kapan menggunakan Replikasi Lintas-Wilayah
- Kapan menggunakan Replikasi Lintas-Wilayah
- Kapan menggunakan replikasi dua arah (replikasi bidireksional)
- Kapan menggunakan Replikasi Batch S3
- Persyaratan beban kerja dan replikasi langsung
- Apa yang direplikasi Amazon S3?
- Persyaratan dan pertimbangan untuk replikasi
- Menyiapkan ikhtisar replikasi langsung
- Mengelola atau menjeda replikasi langsung
- Mereplikasi objek yang ada dengan Batch Replication
- Memecahkan masalah replikasi
- Memantau replikasi dengan metrik, pemberitahuan acara, dan status

Mengapa menggunakan replikasi?

Replikasi dapat membantu Anda melakukan hal berikut:

- Replikasi objek sambil mempertahankan metadata — Anda dapat menggunakan replikasi untuk membuat salinan objek Anda yang mempertahankan semua metadata, seperti waktu dan versi pembuatan objek asli. IDs Kemampuan ini penting jika Anda harus memastikan bahwa replika Anda identik dengan objek sumber.
- Replikasi objek ke dalam kelas penyimpanan yang berbeda–Anda dapat menggunakan replikasi untuk secara langsung memasukkan objek ke dalam S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive, atau kelas penyimpanan lainnya di bucket tujuan. Anda juga dapat mereplikasi

data Anda ke kelas penyimpanan yang sama dan menggunakan konfigurasi siklus hidup pada bucket tujuan untuk memindahkan objek Anda ke kelas penyimpanan yang lebih dingin seiring bertambahnya usia.

- Pertahankan salinan objek di bawah kepemilikan yang berbeda — Terlepas dari siapa yang memiliki objek sumber, Anda dapat memberi tahu Amazon S3 untuk mengubah kepemilikan replika ke yang memiliki Akun AWS bucket tujuan. Ini disebut sebagai opsi penggantian pemilik. Anda dapat menggunakan opsi ini untuk membatasi akses ke replika objek.
- Simpan objek yang disimpan di beberapa tempat Wilayah AWS— Untuk memastikan perbedaan geografis di mana data Anda disimpan, Anda dapat mengatur beberapa bucket tujuan di berbagai tempat. Wilayah AWS Fitur ini dapat membantu Anda memenuhi persyaratan kepatuhan tertentu.
- Replikasi objek dalam waktu 15 menit — Untuk mereplikasi data Anda di Wilayah yang sama Wilayah AWS atau di berbagai Wilayah dalam jangka waktu yang dapat diprediksi, Anda dapat menggunakan Kontrol Waktu Replikasi S3 (S3 RTC). S3 RTC mereplikasi 99,99 persen objek baru yang disimpan di Amazon S3 dalam 15 menit (didukung oleh perjanjian tingkat layanan). Untuk informasi selengkapnya, lihat the section called “Menggunakan Kontrol Waktu Replikasi S3”.

  > **Note**
  >
  > S3 RTC tidak berlaku untuk Replikasi Batch. Replikasi Batch adalah pekerjaan replikasi sesuai permintaan, dan dapat dilacak dengan Operasi Batch S3. Untuk informasi selengkapnya, lihat Melacak status tugas dan laporan penyelesaian.

- Menyinkronkan bucket, mereplikasi objek yang ada, dan mereplikasi objek yang sebelumnya gagal atau direplikasi–Untuk menyinkronkan bucket dan mereplikasi objek yang ada, gunakan Replikasi Batch sebagai tindakan replikasi sesuai permintaan. Untuk informasi selengkapnya tentang kapan menggunakan Replikasi Batch, lihat Kapan menggunakan Replikasi Batch S3.
- Replikasi objek dan failover ke bucket di Wilayah AWS lain–Untuk menjaga agar semua metadata dan objek tetap sinkron di seluruh bucket selama replikasi data, gunakan aturan replikasi dua arah (juga dikenal sebagai replikasi bidireksional) sebelum mengonfigurasi kontrol failover Titik Akses Multi-Wilayah Amazon S3. Aturan replikasi dua arah membantu memastikan bahwa ketika data ditulis ke bucket S3 yang lalu lintas failover, data tersebut kemudian direplikasi kembali ke bucket sumber.

Kapan menggunakan Replikasi Lintas-Wilayah

Replikasi Lintas Wilayah (CRR) S3 digunakan untuk menyalin objek di seluruh bucket Amazon S3 di Wilayah AWS yang berbeda. CRR dapat membantu Anda melakukan ini:

- Memenuhi persyaratan kepatuhan–Meskipun Amazon S3 menyimpan data Anda di beberapa Zona Ketersediaan yang secara geografis berjauhan secara default, persyaratan kepatuhan mungkin mengatur bahwa Anda menyimpan data dengan jarak yang lebih jauh. Untuk memenuhi persyaratan ini, gunakan Replikasi Lintas Wilayah untuk mereplikasi data antara Wilayah AWS yang jauh.
- Minimalkan latensi — Jika pelanggan Anda berada di dua lokasi geografis, Anda dapat meminimalkan latensi dalam mengakses objek dengan mempertahankan salinan objek Wilayah AWS yang secara geografis lebih dekat dengan pengguna Anda.
- Tingkatkan efisiensi operasional — Jika Anda memiliki cluster komputasi dalam dua kelompok berbeda Wilayah AWS yang menganalisis kumpulan objek yang sama, Anda dapat memilih untuk mempertahankan salinan objek di Wilayah tersebut.

Kapan menggunakan Replikasi Lintas-Wilayah

Replikasi Wilayah yang Sama (SRR) digunakan untuk menyalin objek di seluruh bucket Amazon S3 dalam Wilayah AWS yang sama. CRR dapat membantu Anda melakukan ini:

- Menggabungkan log ke dalam satu bucket–Jika Anda menyimpan log dalam beberapa bucket atau beberapa akun, Anda dapat dengan mudah mereplikasi log ke dalam satu bucket di Wilayah. Melakukan hal itu memungkinkan pemrosesan log yang lebih sederhana di satu lokasi.
- Mengonfigurasi replikasi langsung antara akun produksi dan pengujian–Jika Anda atau pelanggan Anda memiliki akun produksi dan pengujian yang menggunakan data yang sama, Anda dapat mereplikasi objek di antara beberapa akun tersebut, sambil mempertahankan metadata objek.
- Mematuhi undang-undang kedaulatan data — Anda mungkin diminta untuk menyimpan beberapa salinan data Anda secara terpisah Akun AWS dalam Wilayah tertentu. Replikasi di Wilayah yang sama dapat membantu Anda mereplikasi data penting secara otomatis ketika peraturan kepatuhan tidak mengizinkan data meninggalkan negara Anda.

Kapan menggunakan replikasi dua arah (replikasi bidireksional)

- Membangun kumpulan data bersama di beberapa Wilayah AWS — Dengan sinkronisasi modifikasi replika, Anda dapat dengan mudah mereplikasi perubahan metadata, seperti daftar kontrol akses objek (ACLs), tag objek, atau kunci objek, pada objek replikasi. Replikasi dua arah ini penting jika Anda ingin menjaga agar semua objek dan perubahan metadata objek tetap sinkron. Anda dapat mengaktifkan sinkronisasi modifikasi replika pada aturan replikasi baru atau yang sudah ada saat melakukan replikasi dua arah antara dua bucket atau lebih dalam Wilayah AWS yang sama atau berbeda.
- Menyinkronkan data di seluruh Wilayah selama failover — Anda dapat menyinkronkan data dalam bucket di antaranya Wilayah AWS dengan mengonfigurasi aturan replikasi dua arah dengan S3 Cross-Region Replication (CRR) langsung dari Titik Akses Multi-Wilayah. Untuk membuat keputusan berdasarkan informasi tentang kapan memulai failover, Anda juga dapat mengaktifkan metrik replikasi S3 sehingga Anda dapat memantau replikasi di Amazon CloudWatch, di Kontrol Waktu Replikasi S3 (S3 RTC), atau dari Titik Akses Multi-Wilayah.
- Menjadikan aplikasi Anda sangat tersedia–Bahkan jika terjadi gangguan lalu lintas Regional, Anda dapat menggunakan aturan replikasi dua arah untuk menjaga agar semua metadata dan objek tetap sinkron di seluruh bucket selama replikasi data.

Kapan menggunakan Replikasi Batch S3

Replikasi Batch mereplikasi objek yang ada ke bucket yang berbeda sebagai opsi sesuai permintaan. Tidak seperti replikasi langsung, pekerjaan ini dapat dijalankan sesuai kebutuhan. Replikasi Batch dapat membantu Anda melakukan hal berikut:

- Mereplikasi objek yang ada–Anda dapat menggunakan Replikasi Batch untuk mereplikasi objek yang ditambahkan ke bucket sebelum Replikasi Wilayah yang Sama atau Replikasi Lintas Wilayah dikonfigurasi.
- Mereplikasi objek yang sebelumnya gagal direplikasi–Anda dapat memfilter pekerjaan Replikasi Batch untuk mencoba mereplikasi objek dengan status replikasi GAGAL.
- Replikasi objek yang sudah direplikasi–Anda mungkin diminta untuk menyimpan beberapa salinan data Anda di Akun AWS atau Wilayah AWS terpisah. Replikasi Batch dapat mereplikasi objek yang ada ke tujuan yang baru ditambahkan.
- Replikasi replika objek yang dibuat dari aturan replikasi–Konfigurasi replikasi membuat replika objek dalam bucket tujuan. Replika objek hanya dapat direplikasi dengan Replikasi Batch.

Persyaratan beban kerja dan replikasi langsung

Bergantung pada persyaratan beban kerja Anda, beberapa jenis replikasi langsung akan lebih cocok untuk kasus penggunaan Anda daripada yang lain. Gunakan tabel berikut untuk menentukan jenis replikasi yang akan digunakan untuk situasi Anda, dan apakah akan menggunakan Kontrol Waktu Replikasi S3 (S3 RTC) untuk beban kerja Anda. S3 RTC mereplikasi 99,99 persen objek baru yang disimpan di Amazon S3 dalam waktu 15 menit (didukung oleh perjanjian tingkat layanan, atau SLA). Untuk informasi selengkapnya, lihat the section called “Menggunakan Kontrol Waktu Replikasi S3”.

Persyaratan beban kerja	S3 RTC (15 menit SLA)	Replikasi Lintas Wilayah (CRR)	Replikasi Wilayah yang Sama (SRR)
Replikasi objek antara yang berbeda Akun AWS	Ya	Ya	Ya
Replikasi objek dalam waktu yang sama Wilayah AWS dalam 24-48 jam (tidak didukung SLA)	Tidak	Tidak	Ya
Replikasi objek antara yang berbeda Wilayah AWS dalam 24-48 jam (tidak didukung SLA)	Tidak	Ya	Tidak
Waktu replikasi yang dapat diprediksi: Didukung oleh SLA untuk mereplikasi 99,9 persen objek dalam 15 menit	Ya	Tidak	Tidak

Apa yang direplikasi Amazon S3?

Amazon S3 hanya mereplikasi item tertentu dalam bucket yang dikonfigurasi untuk replikasi.

Topik

- Apa yang direplikasi dengan konfigurasi replikasi?
- Apa yang tidak direplikasi dengan konfigurasi replikasi?

Apa yang direplikasi dengan konfigurasi replikasi?

Secara default, Amazon S3 mereplikasi berikut ini:

- Objek yang dibuat setelah Anda menambahkan konfigurasi replikasi.
- Objek tidak terenkripsi.
- Objek yang dienkripsi menggunakan kunci yang disediakan pelanggan (SSE-C), objek yang dienkripsi saat istirahat di bawah kunci terkelola Amazon S3 (SSE-S3) atau kunci KMS yang disimpan di (SSE-KMS). AWS Key Management Service Untuk informasi selengkapnya, lihat the section called “Mereplikasi objek terenkripsi”.
- Metadata objek dari objek sumber ke replika. Untuk informasi tentang mereplikasi metadata dari replika ke objek sumber, lihat Mereplikasi perubahan metadata dengan sinkronisasi modifikasi replika.
- Hanya objek di bucket sumber yang pemilik bucket memiliki izin untuk membaca objek dan mengakses daftar kontrol (ACLs).

  Untuk informasi selengkapnya tentang kepemilikan sumber daya, lihat bucket Amazon S3 dan kepemilikan objek.
- Object ACL diperbarui, kecuali jika Anda mengarahkan Amazon S3 untuk mengubah kepemilikan replika ketika bucket sumber dan tujuan tidak dimiliki oleh akun yang sama.

  Untuk informasi selengkapnya, lihat Mengubah pemilik replika.

  Ini bisa memakan waktu beberapa saat hingga Amazon S3 dapat membawa keduanya ACLs sinkron. Perubahan kepemilikan ini hanya berlaku untuk objek yang dibuat setelah Anda menambahkan konfigurasi replikasi ke bucket.
- Tag objek, jika ada.
- Informasi retensi Kunci Objek S3, jika ada.

Saat Amazon S3 mereplikasi objek yang memiliki informasi retensi yang berlaku, itu akan menerapkan kontrol retensi yang sama ke replika Anda, mengabaikan periode retensi default yang dikonfigurasi pada bucket tujuan Anda. Jika Anda tidak memiliki kontrol retensi yang diterapkan pada objek di bucket sumber, dan Anda mereplikasi ke dalam bucket tujuan yang sudah ditetapkan dengan periode retensi default, periode retensi default bucket tujuan diterapkan ke replika objek Anda. Untuk informasi selengkapnya, lihat Mengunci objek dengan Object Lock.

Bagaimana cara menghapus operasi yang memengaruhi replikasi

Jika Anda menghapus objek dari bucket sumber, tindakan berikut terjadi secara default:

- Jika Anda membuat permintaan DELETE tanpa menentukan ID versi objek, Amazon S3 akan menambahkan penanda hapus. Amazon S3 menangani penanda hapus sebagai berikut:
  - Jika Anda menggunakan versi terbaru konfigurasi replikasi (yaitu, Anda menentukan elemen `Filter` pada aturan konfigurasi replikasi), Amazon S3 tidak mereplikasi penanda hapus secara default. Namun, Anda dapat menambahkan replikasi penanda hapus ke non-tag-based aturan. Untuk informasi selengkapnya, lihat Mereplikasi penanda hapus di antara bucket.
  - Jika Anda tidak menentukan elemen `Filter`, Amazon S3 berasumsi bahwa konfigurasi replikasi adalah versi V1 dan mereplikasi penanda hapus yang dihasilkan dari tindakan pengguna. Namun, jika Amazon S3 menghapus objek karena tindakan siklus hidup, penanda hapus tidak direplikasi ke bucket tujuan.
- Jika Anda menentukan ID versi objek yang akan dihapus dalam permintaan `DELETE`, Amazon S3 akan menghapus versi objek tersebut di bucket sumber. Tetapi ini tidak mereplikasi penghapusan di bucket tujuan. Dengan kata lain, itu tidak menghapus versi objek yang sama dari bucket tujuan. Ini melindungi data dari penghapusan berbahaya.

Apa yang tidak direplikasi dengan konfigurasi replikasi?

Secara default, Amazon S3 mereplikasi berikut ini:

- Objek dalam bucket sumber yang merupakan replika yang dibuat oleh aturan replikasi lain. Misalnya, Anda mengonfigurasi replikasi dengan bucket A sebagai sumber dan bucket B sebagai tujuan. Sekarang bayangkan bahwa Anda menambahkan konfigurasi replikasi lain dengan bucket B sebagai sumber dan bucket C sebagai tujuan. Dalam hal ini, objek dalam bucket B yang merupakan replika objek dalam bucket A tidak direplikasi ke bucket C.

Untuk mereplikasi objek yang merupakan replika, gunakan Replikasi Batch. Pelajari lebih lanjut mengonfigurasi Replikasi Batch di Mereplikasi objek yang ada.

- Objek dalam bucket sumber yang telah direplikasi ke tujuan yang berbeda. Misalnya, jika Anda mengubah bucket tujuan dalam konfigurasi replikasi yang ada, Amazon S3 tidak akan mereplikasi objek itu lagi.

  Untuk mereplikasi objek yang direplikasi sebelumnya, gunakan Replikasi Batch. Pelajari lebih lanjut mengonfigurasi Replikasi Batch di Mereplikasi objek yang ada.

- Replikasi Batch tidak mendukung replikasi ulang objek yang telah dihapus dengan ID versi objek dari bucket tujuan. Untuk mereplikasi ulang objek ini, Anda dapat menyalin objek sumber di tempat dengan tugas Penyalinan Batch. Menyalin objek tersebut di tempat akan membuat versi baru objek di bucket sumber dan memulai replikasi secara otomatis ke tujuan. Untuk informasi selengkapnya tentang cara menggunakan Penyalinan Batch, lihat, Contoh yang menggunakan Operasi Batch untuk menyalin objek.

- Secara default, saat mereplikasi dari yang berbeda Akun AWS, penanda hapus yang ditambahkan ke bucket sumber tidak akan direplikasi.

  Untuk informasi tentang cara mereplikasi penanda hapus, lihat Mereplikasi penanda hapus di antara bucket.

- Objek yang disimpan di S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive, S3 Intelligent-Tiering Archive Access, atau kelas penyimpanan S3 Intelligent-Tiering Deep Archive Access atau tingkatan. Anda tidak dapat mereplikasi objek ini sampai Anda mengembalikannya dan menyalinnya ke kelas penyimpanan yang berbeda.

  Untuk mempelajari lebih lanjut tentang S3 Glacier Flexible Retrieval dan S3 Glacier Deep Archive, lihat. Kelas penyimpanan untuk objek yang jarang diakses

  Untuk mempelajari lebih lanjut tentang S3 Intelligent-Tiering, lihat. Mengelola biaya penyimpanan dengan Amazon S3 Intelligent-Tiering

- Objek di bucket sumber yang pemilik bucket-nya tidak memiliki izin yang cukup untuk replikasi.

  Untuk informasi tentang bagaimana pemilik objek dapat memberikan izin kepada pemilik bucket, lihat Berikan izin lintas akun untuk unggah objek sekaligus memastikan bahwa pemilik bucket memiliki kendali penuh.

- Pembaruan untuk subsumber daya tingkat bucket.

Misalnya, jika Anda mengubah konfigurasi siklus aktif atau menambahkan konfigurasi pemberitahuan ke bucket sumber Anda, perubahan ini tidak diterapkan ke bucket tujuan. Fitur ini memungkinkan untuk memiliki konfigurasi berbeda pada bucket sumber dan tujuan.

- Tindakan yang dilakukan berdasarkan konfigurasi siklus aktif.

  Misalnya, jika konfigurasi siklus hidup diaktifkan hanya pada bucket sumber Anda, Amazon S3 membuat penanda hapus untuk objek yang kedaluwarsa tetapi tidak mereplikasi penanda tersebut. Jika Anda ingin konfigurasi siklus aktif yang sama diterapkan pada bucket sumber dan tujuan, aktifkan konfigurasi siklus aktif yang sama pada keduanya. Untuk informasi lebih lanjut tentang konfigurasi siklus aktif, lihat Mengelola siklus hidup objek.

- Saat Anda menggunakan aturan replikasi berbasis tag dengan replikasi langsung, objek baru harus ditandai dengan tag aturan replikasi yang cocok dalam operasi. `PutObject` Jika tidak, objek tidak akan direplikasi. Jika objek diberi tag setelah `PutObject` operasi, objek tersebut juga tidak akan direplikasi.

  Untuk mereplikasi objek yang telah ditandai setelah `PutObject` operasi, Anda harus menggunakan Replikasi Batch S3. Untuk informasi selengkapnya tentang Replikasi Batch, lihat Mereplikasi objek yang ada.

Persyaratan dan pertimbangan untuk replikasi

Replikasi Amazon S3 membutuhkan yang berikut:

- Pemilik bucket sumber harus Wilayah AWS mengaktifkan sumber dan tujuan untuk akun mereka. Pemilik bucket tujuan harus mengaktifkan Wilayah tujuan untuk akun mereka.

  Untuk informasi selengkapnya tentang mengaktifkan atau menonaktifkan Wilayah AWS, lihat Menentukan Wilayah AWS akun yang dapat digunakan dalam Panduan Referensi AWS Account Management .

- Bucket sumber dan tujuan harus memiliki Penentuan Versi aktif. Untuk informasi selengkapnya tentang penentuan versi, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.

- Amazon S3 harus memiliki izin untuk mereplikasi objek dari bucket sumber ke bucket tujuan atau bucket atas nama Anda. Untuk informasi selengkapnya tentang izin ini, lihat Menyiapkan izin untuk replikasi langsung.

- Apabila pemilik bucket sumber tidak memiliki objek dalam bucket, pemilik objek harus memberi pemilik bucket izin `READ` dan `READ_ACP` dengan daftar kontrol akses (ACL) objek. Untuk informasi selengkapnya, lihat Gambaran umum daftar kontrol akses (ACL).
- Jika bucket sumber mengaktifkan Kunci Objek S3, bucket tujuan juga harus mengaktifkan Kunci Objek S3.

  Untuk mengaktifkan replikasi pada bucket yang mengaktifkan Object Lock, Anda harus menggunakan AWS Command Line Interface, REST API, atau AWS SDKs. Untuk informasi umum selengkapnya, lihat Mengunci objek dengan Object Lock.

  > **Note**
  >
  > Anda harus memberikan dua izin baru pada bucket S3 sumber dalam peran AWS Identity and Access Management (IAM) yang Anda gunakan untuk menyiapkan replikasi. Dua izin baru adalah `s3:GetObjectRetention` dan `s3:GetObjectLegalHold`. Jika peran memiliki izin `s3:Get*`, itu memenuhi persyaratan. Untuk informasi selengkapnya, lihat Menyiapkan izin untuk replikasi langsung.

Untuk informasi selengkapnya, lihat Menyiapkan ikhtisar replikasi langsung.

Jika Anda mengatur konfigurasi replikasi dalam skenario lintas akun, yang bucket sumber dan tujuannya dimiliki oleh Akun AWS yang berbeda, persyaratan tambahan berikut ini berlaku:

- Pemilik bucket tujuan harus memberikan izin kepada pemilik bucket sumber untuk mereplikasi objek dengan kebijakan bucket. Untuk informasi selengkapnya, lihat Memberikan izin saat bucket sumber dan tujuan dimiliki oleh yang berbeda Akun AWS.
- Bucket tujuan tidak dapat dikonfigurasi sebagai bucket Pembayaran oleh Pemohon. Untuk informasi selengkapnya, lihat Menggunakan Requester Pays ember tujuan umum untuk transfer dan penggunaan penyimpanan.

Pertimbangan untuk replikasi

Sebelum Anda membuat konfigurasi replikasi, perhatikan pertimbangan berikut.

Topik

- Konfigurasi siklus hidup dan replika objek

- Konfigurasi Penentuan Versi dan konfigurasi replikasi
- Menggunakan Replikasi S3 dengan S3 Intelligent-Tiering
- Konfigurasi pencatatan dan konfigurasi replikasi
- CRR dan wilayah tujuan
- Replikasi Batch S3
- Kontrol Waktu Replikasi S3

Konfigurasi siklus hidup dan replika objek

Waktu yang diperlukan Amazon S3 untuk mereplikasi objek tergantung pada ukuran objek. Untuk objek besar, dibutuhkan waktu beberapa jam. Meskipun mungkin perlu beberapa saat sebelum replika tersedia di tujuan, dibutuhkan waktu yang sama untuk membuat replika seperti yang diperlukan untuk membuat objek yang sesuai di bucket sumber. Jika konfigurasi siklus hidup diaktifkan pada bucket tujuan, aturan siklus hidup menghormati waktu pembuatan asli objek, bukan saat replika tersedia di bucket tujuan.

Konfigurasi replikasi memerlukan bucket dengan dukungan Penentuan Versi. Saat Anda mengaktifkan Penentuan Versi di bucket, ingatlah hal berikut:

- Jika Anda memiliki konfigurasi siklus hidup Kedaluwarsa objek, setelah Anda mengaktifkan Penentuan Versi, tambahkan kebijakan `NonCurrentVersionExpiration` untuk mempertahankan perilaku penghapusan permanen yang sama seperti sebelum Anda mengaktifkan Penentuan Versi.
- Jika Anda memiliki konfigurasi siklus hidup Transisi, setelah Anda mengaktifkan Penentuan Versi, pertimbangkan untuk menambahkan kebijakan `NonCurrentVersionTransition`.

Konfigurasi Penentuan Versi dan konfigurasi replikasi

Bucket sumber dan tujuan harus memiliki dukungan Penentuan Versi ketika Anda mengonfigurasi replikasi pada bucket. Setelah Anda mengaktifkan Penentuan Versi pada bucket sumber dan tujuan dan mengonfigurasi replikasi pada bucket sumber, Anda akan menemui masalah berikut ini:

- Jika Anda mencoba menonaktifkan Penentuan Versi pada bucket sumber, Amazon S3 menampilkan pesan kesalahan. Anda harus menghapus konfigurasi replikasi sebelum Anda dapat menonaktifkan Penentuan Versi pada bucket sumber.

- Jika Anda menonaktifkan Penentuan Versi pada bucket tujuan, replikasi akan gagal. Objek sumber memiliki status replikasi `FAILED`.

Menggunakan Replikasi S3 dengan S3 Intelligent-Tiering

S3 Intelligent-Tiering adalah kelas penyimpanan yang dirancang untuk mengoptimalkan biaya penyimpanan dengan memindahkan data ke tingkat akses yang paling hemat biaya secara otomatis. Untuk biaya pemantauan dan otomatisasi objek bulanan yang murah, S3 Intelligent-Tiering memantau pola akses dan secara otomatis memindahkan objek yang belum diakses ke jenjang akses berbiaya rendah.

Mereplikasi objek yang disimpan dalam S3 Intelligent-Tiering dengan Replikasi Batch S3 atau memanggil CopyObject atau UploadPartCopy merupakan akses. Dalam kasus ini, objek sumber dari operasi salinan atau replikasi naik tingkat.

Untuk informasi selengkapnya tentang S3 Intelligent-Tiering lihat, Mengelola biaya penyimpanan dengan Amazon S3 Intelligent-Tiering.

Konfigurasi pencatatan dan konfigurasi replikasi

Jika Amazon S3 mengirimkan log ke bucket dengan dukungan replikasi, Amazon S3 mereplikasi objek log.

Jika log atau AWS CloudTrail log akses server diaktifkan di bucket sumber atau tujuan Anda, Amazon S3 menyertakan permintaan terkait replikasi di log. Misalnya, Amazon S3 mencatat setiap objek yang direplikasi.

CRR dan wilayah tujuan

Amazon S3 Cross-Region Replication (CRR) digunakan untuk menyalin objek di bucket S3 secara berbeda. Wilayah AWS Anda dapat memilih Wilayah untuk bucket tujuan Anda berdasarkan kebutuhan bisnis atau pertimbangan biaya Anda. Misalnya, biaya transfer data antar Wilayah berbeda tergantung Wilayah yang Anda pilih.

Misalkan Anda memilih US East (Virginia N.) (`us-east-1`) sebagai Wilayah untuk ember sumber Anda. Jika Anda memilih US West (Oregon) (`us-west-2`) sebagai Wilayah untuk ember tujuan Anda, Anda membayar lebih dari jika Anda memilih Wilayah AS Timur (Ohio) (`us-east-2`). Untuk informasi harga, lihat "Harga Transfer Data" di Harga Amazon S3.

Tidak ada biaya transfer data yang terkait dengan Replikasi Wilayah yang Sama (SRR).

Replikasi Batch S3

Untuk informasi tentang pertimbangan untuk Replikasi Batch, lihat. Pertimbangan Replikasi Batch S3

Kontrol Waktu Replikasi S3

Untuk informasi tentang praktik terbaik dan pertimbangan untuk Kontrol Waktu Replikasi S3 (S3 RTC), lihat. Praktik terbaik dan pedoman untuk S3 RTC

Menyiapkan ikhtisar replikasi langsung

> **Note**
>
> Objek yang ada sebelum Anda menyiapkan replikasi tidak direplikasi secara otomatis. Dengan kata lain, Amazon S3 tidak mereplikasi objek secara retroaktif. Untuk mereplikasi objek yang dibuat sebelum konfigurasi replikasi Anda, gunakan Replikasi Batch S3. Untuk informasi selengkapnya tentang mengonfigurasi Replikasi Batch, lihat. Mereplikasi objek yang ada

Untuk mengaktifkan replikasi langsung—Same-region Replication (SRR) atau Cross-Region Replication (CRR) —tambahkan konfigurasi replikasi ke bucket sumber Anda. Konfigurasi ini memberitahu Amazon S3 untuk mereplikasi objek seperti yang ditentukan. Dalam konfigurasi replikasi, Anda harus memberikan berikut ini:

- Bucket tujuan–Bucket tempat Anda ingin Amazon S3 mereplikasi objek.
- Objek yang ingin Anda replikasi — Anda dapat mereplikasi semua objek di bucket sumber atau subset objek. Anda mengidentifikasi subset dengan menyediakan awalan nama kunci, satu atau lebih tag objek, atau keduanya dalam konfigurasi.

  Misalnya, jika Anda mengonfigurasi aturan replikasi agar hanya mereplikasi objek dengan prefiks nama kunci `Tax/`, Amazon S3 akan mereplikasi objek dengan kunci seperti `Tax/doc1` atau `Tax/doc2`. Namun, aturan ini tidak mereplikasi objek dengan kunci `Legal/doc3`. Jika Anda menentukan prefiks dan satu atau beberapa tag, Amazon S3 hanya mereplikasi objek yang memiliki kunci prefiks dan tag tertentu.
- Peran AWS Identity and Access Management (IAM) - Amazon S3 mengasumsikan peran IAM ini untuk mereplikasi objek atas nama Anda. Untuk informasi selengkapnya tentang membuat peran IAM ini dan mengelola izin, lihat. Menyiapkan izin untuk replikasi langsung

Selain persyaratan minimum ini, Anda dapat memilih opsi berikut:

- Kelas penyimpanan replika–Secara default, Amazon S3 menyimpan replika objek menggunakan kelas penyimpanan yang sama dengan objek sumber. Anda dapat menentukan kelas penyimpanan yang berbeda untuk replika.
- Kepemilikan replika–Amazon S3 berasumsi bahwa replika objek terus dimiliki oleh pemilik objek sumber. Jadi, ketika mereplikasi objek, ini juga mereplikasi daftar kontrol akses (ACL) objek yang terkait atau pengaturan Kepemilikan Objek S3. Jika bucket sumber dan tujuan dimiliki oleh Akun AWS yang berbeda, Anda dapat mengonfigurasi replikasi untuk mengubah pemilik replika ke Akun AWS yang memiliki bucket tujuan. Untuk informasi selengkapnya, lihat the section called “Mengubah pemilik replika”.

Anda dapat mengonfigurasi replikasi dengan menggunakan konsol Amazon S3 AWS Command Line Interface ,AWS CLI() AWS SDKs, atau Amazon S3 REST API. Untuk panduan mendetail tentang cara mengatur replikasi, lihat. the section called “Panduan replikasi”

Amazon S3 menyediakan operasi REST API untuk mendukung pengaturan aturan replikasi. Untuk informasi selengkapnya, lihat topik berikut di Referensi API Amazon Simple Storage Service:

- PutBucketReplication
- GetBucketReplication
- DeleteBucketReplication

Topik

- Elemen file konfigurasi replikasi
- Menyiapkan izin untuk replikasi langsung
- Contoh untuk mengonfigurasi replikasi langsung

Elemen file konfigurasi replikasi

Amazon S3 menyimpan konfigurasi replikasi sebagai XML. Jika Anda mengonfigurasi replikasi secara terprogram melalui Amazon S3 REST API, Anda menentukan berbagai elemen konfigurasi replikasi Anda dalam file XHTML ini. Jika Anda mengonfigurasi replikasi melalui AWS Command Line Interface (AWS CLI), Anda menentukan konfigurasi replikasi menggunakan format JSON. Untuk contoh JSON, lihat penelusuran di. the section called “Panduan replikasi”

> **Note**
>
> Versi terbaru dari format XML-konfigurasi replikasi adalah V2. Konfigurasi replikasi XML V2 adalah yang berisi elemen `<Filter>` untuk aturan, dan aturan yang menentukan Kontrol Waktu Replikasi S3 (S3 RTC).
> Untuk melihat versi konfigurasi replikasi, Anda dapat menggunakan operasi API `GetBucketReplication`. Untuk informasi selengkapnya, lihat GetBucketReplication dalam Referensi API Amazon Simple Storage Service.
> Untuk kompatibilitas mundur, Amazon S3 terus mendukung format konfigurasi replikasi XMLV1. Jika Anda telah menggunakan format konfigurasi replikasi XMLV1, lihat Pertimbangan kompatibilitas mundur untuk pertimbangan kompatibilitas mundur.

Dalam file XHTML konfigurasi replikasi, Anda harus menentukan peran AWS Identity and Access Management (IAM) dan satu atau beberapa aturan, seperti yang ditunjukkan pada contoh berikut:

```
<ReplicationConfiguration>
    <Role>IAM-role-ARN</Role>
    <Rule>
        ...
    </Rule>
    <Rule>
         ...
    </Rule>
     ...
</ReplicationConfiguration>
```

Amazon S3 tidak dapat mereplikasi objek tanpa izin Anda. Anda memberikan izin ke Amazon S3 dengan peran IAM yang Anda tentukan dalam konfigurasi replikasi. Amazon S3 mengasumsikan peran IAM ini untuk mereplikasi objek atas nama Anda. Anda harus terlebih dahulu memberikan izin yang diperlukan untuk peran IAM. Untuk informasi selengkapnya tentang mengelola izin, lihat Menyiapkan izin untuk replikasi langsung.

Anda hanya menambahkan satu aturan dalam konfigurasi replikasi dalam skenario berikut:

- Anda ingin mereplikasi semua objek.
- Anda ingin mereplikasi hanya satu subset objek. Anda mengidentifikasi subset objek dengan menambahkan filter pada aturan. Dalam filter, Anda menentukan awalan key objek, tag,

atau kombinasi keduanya untuk mengidentifikasi subset objek yang diterapkan aturan. Filter menargetkan objek yang sesuai dengan nilai persis yang Anda tentukan.

Jika Anda ingin mereplikasi subset objek yang berbeda, Anda menambahkan beberapa aturan dalam konfigurasi replikasi. Di setiap aturan, Anda menentukan filter yang memilih subset yang berbeda. Misalnya, Anda dapat memilih untuk mereplikasi objek yang memiliki awalan kunci `tax/` atau `document/`. Untuk melakukan ini, Anda menambahkan dua aturan, satu yang menentukan filter awalan kunci `tax/` dan satu lagi yang menentukan awalan kunci `document/`. Untuk informasi selengkapnya tentang awalan kunci objek, lihat. Organisasi objek menggunakan prefiks

Bagian berikut memberikan informasi tambahan.

Topik

- Konfigurasi aturan dasar
- Opsional: Menentukan filter
- Konfigurasi tujuan tambahan
- Contoh konfigurasi replikasi
- Pertimbangan kompatibilitas mundur

Konfigurasi aturan dasar

Setiap aturan harus menyertakan status dan prioritas aturan. Aturan juga harus menunjukkan apakah akan mereplikasi penanda hapus.

- `<Status>`Elemen menunjukkan apakah aturan diaktifkan atau dinonaktifkan dengan menggunakan nilai `Enabled` atau`Disabled`. Jika aturan dinonaktifkan, Amazon S3 tidak akan melakukan tindakan yang ditentukan dalam aturan tersebut.
- `<Priority>`Elemen menunjukkan aturan mana yang diutamakan setiap kali dua atau lebih aturan replikasi bertentangan. Amazon S3 mencoba untuk mereplikasi objek sesuai dengan semua aturan replikasi. Namun, jika ada dua aturan atau lebih dengan bucket tujuan yang sama, objek akan direplikasi sesuai aturan dengan prioritas tertinggi. Semakin tinggi angkanya, semakin tinggi prioritasnya.
- `<DeleteMarkerReplication>`Elemen menunjukkan apakah akan mereplikasi penanda hapus dengan menggunakan nilai `Enabled` atau. `Disabled`

Dalam konfigurasi `<Destination>` elemen, Anda harus memberikan nama bucket tujuan atau bucket tempat Anda ingin Amazon S3 mereplikasi objek.

Contoh berikut menunjukkan persyaratan minimum untuk aturan V2. Untuk kompatibilitas mundur, Amazon S3 terus mendukung format XML V1. Untuk informasi selengkapnya, lihat Pertimbangan kompatibilitas mundur.

```
...
    <Rule>
        <ID>Rule-1</ID>
        <Status>Enabled-or-Disabled</Status>
        <Filter>
            <Prefix></Prefix>
        </Filter>
        <Priority>integer</Priority>
        <DeleteMarkerReplication>
           <Status>Enabled-or-Disabled</Status>
        </DeleteMarkerReplication>
        <Destination>
           <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
        </Destination>
    </Rule>
    <Rule>
         ...
    </Rule>
     ...
...
```

Anda juga dapat menentukan opsi konfigurasi lainnya. Misalnya, Anda dapat memilih untuk menggunakan kelas penyimpanan untuk replika objek yang berbeda dengan kelas untuk objek sumber.

Opsional: Menentukan filter

Untuk memilih subset objek yang menerapkan aturan, tambahkan filter opsional. Anda dapat memfilter berdasarkan awalan kunci objek, tag objek, atau kombinasi keduanya. Jika Anda memfilter kunci prefiks dan tag objek, Amazon S3 menggabungkan filter dengan menggunakan operator `AND` logis. Dengan kata lain, aturan berlaku untuk subset objek dengan awalan kunci khusus maupun tag tertentu.

Pemfilteran berdasarkan awalan kunci objek

Untuk menentukan aturan dengan filter berdasarkan key prefix objek, gunakan XHTML berikut. Anda hanya dapat menentukan satu awalan per aturan.

```
<Rule>
    ...
    <Filter>
        <Prefix>key-prefix</Prefix>
    </Filter>
    ...
</Rule>
...
```

Pemfilteran berdasarkan tag objek

Untuk menentukan aturan dengan filter berdasarkan tag objek, gunakan XHTML berikut. Anda dapat menentukan satu atau beberapa tag objek.

```
<Rule>
    ...
    <Filter>
        <And>
            <Tag>
                <Key>key1</Key>
                <Value>value1</Value>
            </Tag>
            <Tag>
                <Key>key2</Key>
                <Value>value2</Value>
            </Tag>
             ...
        </And>
    </Filter>
    ...
</Rule>
...
```

Filter dengan prefiks kunci dan tag objek

Untuk menentukan filter aturan dengan kombinasi key prefix dan tag objek, gunakan XHTML berikut. Anda membungkus filter ini dalam elemen induk <And>. Amazon S3 melakukan operasi AND logis untuk menggabungkan filter ini. Dengan kata lain, aturan berlaku untuk subset objek dengan awalan kunci khusus maupun tag tertentu.

```
<Rule>
    ...
    <Filter>
        <And>
            <Prefix>key-prefix</Prefix>
            <Tag>
                <Key>key1</Key>
                <Value>value1</Value>
            </Tag>
            <Tag>
                <Key>key2</Key>
                <Value>value2</Value>
            </Tag>
             ...
    </Filter>
    ...
</Rule>
...
```

Note

- Jika Anda menentukan aturan dengan `<Filter>` elemen kosong, aturan Anda berlaku untuk semua objek di bucket Anda.
- Saat Anda menggunakan aturan replikasi berbasis tag dengan replikasi langsung, objek baru harus ditandai dengan tag aturan replikasi yang cocok dalam operasi. `PutObject` Jika tidak, objek tidak akan direplikasi. Jika objek diberi tag setelah `PutObject` operasi, objek tersebut juga tidak akan direplikasi.

  Untuk mereplikasi objek yang telah ditandai setelah `PutObject` operasi, Anda harus menggunakan Replikasi Batch S3. Untuk informasi selengkapnya tentang Replikasi Batch, lihat Mereplikasi objek yang ada.

Konfigurasi tujuan tambahan

Dalam konfigurasi tujuan, Anda menentukan bucket tempat Anda ingin Amazon S3 mereplikasi objek. Anda dapat mengatur konfigurasi untuk mereplikasi objek dari satu bucket sumber ke satu atau beberapa bucket tujuan.

```
...
```

```
<Destination>
    <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
</Destination>
...
```

Anda dapat menambahkan opsi berikut di elemen `<Destination>`.

Topik

- Menentukan kelas penyimpanan
- Menambahkan beberapa bucket tujuan
- Menentukan parameter berbeda untuk setiap aturan replikasi dengan beberapa bucket tujuan
- Mengubah kepemilikan replika
- Mengaktifkan Kontrol Waktu Replikasi S3
- Replikasi objek yang dibuat dengan enkripsi sisi server dengan menggunakan AWS KMS

Menentukan kelas penyimpanan

Anda dapat menentukan kelas penyimpanan untuk replika objek. Secara default, Amazon S3 menggunakan kelas penyimpanan objek sumber untuk membuat replika objek, seperti dalam contoh berikut.

```
...
<Destination>
       <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
       <StorageClass>storage-class</StorageClass>
</Destination>
...
```

Menambahkan beberapa bucket tujuan

Anda dapat menambahkan beberapa bucket tujuan dalam satu konfigurasi replikasi, seperti berikut.

```
...
<Rule>
    <ID>Rule-1</ID>
    <Status>Enabled-or-Disabled</Status>
    <Priority>integer</Priority>
    <DeleteMarkerReplication>
       <Status>Enabled-or-Disabled</Status>
```

```
    </DeleteMarkerReplication>
    <Destination>
       <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket1</Bucket>
    </Destination>
</Rule>
<Rule>
    <ID>Rule-2</ID>
    <Status>Enabled-or-Disabled</Status>
    <Priority>integer</Priority>
    <DeleteMarkerReplication>
       <Status>Enabled-or-Disabled</Status>
    </DeleteMarkerReplication>
    <Destination>
       <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket2</Bucket>
    </Destination>
</Rule>
...
```

Menentukan parameter berbeda untuk setiap aturan replikasi dengan beberapa bucket tujuan

Ketika menambahkan beberapa bucket tujuan di satu konfigurasi replikasi, Anda dapat menentukan parameter berbeda untuk setiap aturan replikasi, seperti berikut.

```
...
<Rule>
    <ID>Rule-1</ID>
    <Status>Enabled-or-Disabled</Status>
    <Priority>integer</Priority>
    <DeleteMarkerReplication>
       <Status>Disabled</Status>
    </DeleteMarkerReplication>
      <Metrics>
    <Status>Enabled</Status>
    <EventThreshold>
      <Minutes>15</Minutes>
    </EventThreshold>
  </Metrics>
    <Destination>
       <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket1</Bucket>
    </Destination>
</Rule>
<Rule>
    <ID>Rule-2</ID>
```

```
    <Status>Enabled-or-Disabled</Status>
    <Priority>integer</Priority>
    <DeleteMarkerReplication>
       <Status>Enabled</Status>
    </DeleteMarkerReplication>
      <Metrics>
    <Status>Enabled</Status>
    <EventThreshold>
      <Minutes>15</Minutes>
    </EventThreshold>
  </Metrics>
  <ReplicationTime>
    <Status>Enabled</Status>
    <Time>
      <Minutes>15</Minutes>
    </Time>
  </ReplicationTime>
    <Destination>
       <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket2</Bucket>
    </Destination>
</Rule>
...
```

Mengubah kepemilikan replika

Jika bucket sumber dan tujuan tidak dimiliki oleh akun yang sama, Anda dapat mengubah kepemilikan replika menjadi bucket Akun AWS yang memiliki tujuan. Untuk melakukannya, tambahkan elemen `<AccessControlTranslation>`. Elemen ini mengambil nilai `Destination`.

```
...
<Destination>
   <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
   <Account>destination-bucket-owner-account-id</Account>
   <AccessControlTranslation>
       <Owner>Destination</Owner>
   </AccessControlTranslation>
</Destination>
...
```

Jika Anda tidak menambahkan `<AccessControlTranslation>` elemen ke konfigurasi replikasi, replika dimiliki oleh yang sama Akun AWS yang memiliki objek sumber. Untuk informasi selengkapnya, lihat Mengubah pemilik replika.

Mengaktifkan Kontrol Waktu Replikasi S3

Anda dapat mengaktifkan Kontrol Waktu Replikasi S3 (S3 RTC) dalam konfigurasi replikasi Anda. S3 RTC mereplikasi sebagian besar objek dalam hitungan detik dan 99,99 persen objek dalam waktu 15 menit (didukung oleh perjanjian tingkat layanan).

> **Note**
>
> Hanya nilai yang `<Minutes>15</Minutes>` diterima untuk `<Time>` elemen `<EventThreshold>` dan.

```
...
<Destination>
  <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
  <Metrics>
    <Status>Enabled</Status>
    <EventThreshold>
      <Minutes>15</Minutes>
    </EventThreshold>
  </Metrics>
  <ReplicationTime>
    <Status>Enabled</Status>
    <Time>
      <Minutes>15</Minutes>
    </Time>
  </ReplicationTime>
</Destination>
...
```

Untuk informasi selengkapnya, lihat Memenuhi persyaratan kepatuhan dengan Kontrol Waktu Replikasi S3. Untuk contoh API, lihat PutBucketReplication dalam Referensi API Amazon Simple Storage Service.

Replikasi objek yang dibuat dengan enkripsi sisi server dengan menggunakan AWS KMS

Bucket sumber Anda mungkin berisi objek yang dibuat dengan enkripsi sisi server menggunakan kunci AWS Key Management Service (AWS KMS) (SSE-KMS). Secara default, Amazon S3 tidak mereplikasi objek ini. Anda secara opsional dapat mengarahkan Amazon S3 untuk mereplikasi objek ini. Untuk melakukannya, ikut serta dalam fitur ini secara eksplisit terlebih dahulu dengan menambahkan elemen `<SourceSelectionCriteria>`. Kemudian berikan AWS KMS key (untuk

bucket tujuan) yang akan digunakan untuk mengenkripsi replika objek. Wilayah AWS Contoh berikut menunjukkan cara menentukan elemen-elemen ini.

```
...
<SourceSelectionCriteria>
  <SseKmsEncryptedObjects>
    <Status>Enabled</Status>
  </SseKmsEncryptedObjects>
</SourceSelectionCriteria>
<Destination>
  <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
  <EncryptionConfiguration>
    <ReplicaKmsKeyID>AWS KMS key ID to use for encrypting object replicas</
ReplicaKmsKeyID>
  </EncryptionConfiguration>
</Destination>
...
```

Untuk informasi selengkapnya, lihat Mereplikasi objek terenkripsi (SSE-S3, SSE-KMS, DSSE-KMS, SSE-C).

Contoh konfigurasi replikasi

Untuk memulai, Anda dapat menambahkan contoh konfigurasi replikasi berikut ke bucket Anda, sesuai kebutuhan.

> **Important**
>
> Untuk menambahkan konfigurasi replikasi ke bucket, Anda harus memiliki izin `iam:PassRole`. Dengan izin ini, Anda dapat meneruskan peran IAM yang memberikan izin replikasi Amazon S3. Anda menentukan peran IAM dengan menyediakan Amazon Resource Name (ARN) yang digunakan dalam elemen `<Role>` di XML konfigurasi replikasi. Untuk informasi selengkapnya, lihat Memberikan Izin Pengguna untuk Meneruskan Peran ke Layanan AWS di Panduan Pengguna IAM.

Example 1: Konfigurasi replikasi dengan satu aturan

Konfigurasi replikasi dasar berikut menentukan satu aturan. Aturan tersebut menentukan peran IAM yang dapat diasumsikan Amazon S3 dan satu bucket tujuan untuk replika objek. Nilai `<Status>` elemen `Enabled` menunjukkan bahwa aturan tersebut berlaku.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>

    <Destination>
      <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
    </Destination>
  </Rule>
</ReplicationConfiguration>
```

Untuk memilih subset objek yang akan direplikasi, Anda dapat menambahkan filter. Dalam konfigurasi berikut, filter menentukan awalan kunci objek. Aturan ini berlaku untuk objek yang memiliki awalan *Tax/* dalam nama kuncinya.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
       <Status>string</Status>
    </DeleteMarkerReplication>

    <Filter>
       <Prefix>Tax/</Prefix>
    </Filter>

    <Destination>
       <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
    </Destination>

  </Rule>
</ReplicationConfiguration>
```

Jika menentukan elemen `<Filter>`, Anda juga harus menyertakan elemen `<Priority>` dan `<DeleteMarkerReplication>`. Dalam contoh ini, nilai yang Anda tetapkan untuk `<Priority>` elemen tidak relevan karena hanya ada satu aturan.

Dalam konfigurasi berikut, filter menentukan satu awalan dan dua tag. Aturan ini berlaku untuk subset objek yang memiliki awalan dan tag kunci yang ditentukan. Secara khusus, ini berlaku untuk objek yang memiliki *Tax/* awalan dalam nama kunci mereka dan dua tag objek yang ditentukan. Dalam contoh ini, nilai yang Anda tetapkan untuk `<Priority>` elemen tidak relevan karena hanya ada satu aturan.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
       <Status>string</Status>
    </DeleteMarkerReplication>

    <Filter>
        <And>
          <Prefix>Tax/</Prefix>
          <Tag>
             <Tag>
                <Key>tagA</Key>
                <Value>valueA</Value>
             </Tag>
          </Tag>
          <Tag>
             <Tag>
                <Key>tagB</Key>
                <Value>valueB</Value>
             </Tag>
          </Tag>
       </And>

    </Filter>

    <Destination>
        <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
    </Destination>

  </Rule>
</ReplicationConfiguration>
```

Anda dapat menentukan kelas penyimpanan untuk replika objek sebagai berikut:

```
<?xml version="1.0" encoding="UTF-8"?>

<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>
    <Destination>
       <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
       <StorageClass>storage-class</StorageClass>
    </Destination>
  </Rule>
</ReplicationConfiguration>
```

Anda dapat menentukan kelas penyimpanan apa pun yang didukung Amazon S3.

Example 2: Konfigurasi replikasi dengan dua aturan

Example

Dalam konfigurasi replikasi berikut, aturan menentukan yang berikut:

- Setiap aturan memfilter pada awalan kunci yang berbeda sehingga setiap aturan berlaku untuk subset objek yang berbeda. Dalam contoh ini, Amazon S3 mereplikasi objek dengan nama kunci *Tax/doc1.pdf* dan *Project/project1.txt*, tetapi tidak mereplikasi objek dengan nama kunci *PersonalDoc/documentA*.
- Meskipun kedua aturan menentukan nilai untuk `<Priority>` elemen, prioritas aturan tidak relevan karena aturan berlaku untuk dua set objek yang berbeda. Contoh berikutnya menunjukkan apa yang terjadi ketika prioritas aturan diterapkan.
- Aturan kedua menetapkan kelas penyimpanan S3 Standard-IA untuk replika objek. Amazon S3 menggunakan kelas penyimpanan yang ditentukan untuk replika objek tersebut.

```
<?xml version="1.0" encoding="UTF-8"?>

<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>
```

```
    <Priority>1</Priority>
    <DeleteMarkerReplication>
       <Status>string</Status>
    </DeleteMarkerReplication>
    <Filter>
        <Prefix>Tax</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Destination>
      <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
    </Destination>
     ...
  </Rule>
 <Rule>
    <Status>Enabled</Status>
    <Priority>2</Priority>
    <DeleteMarkerReplication>
       <Status>string</Status>
    </DeleteMarkerReplication>
    <Filter>
        <Prefix>Project</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Destination>
      <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
     <StorageClass>STANDARD_IA</StorageClass>
    </Destination>
     ...
  </Rule>


</ReplicationConfiguration>
```

Example 3: Konfigurasi replikasi dengan dua aturan dengan awalan yang tumpang tindih

Dalam konfigurasi ini, dua aturan yang menentukan filter dengan awalan kunci yang tumpang tindih, *star/* dan *starship/*. Kedua aturan berlaku untuk objek dengan nama kunci *starship-x*. Dalam kasus ini, Amazon S3 menggunakan aturan prioritas untuk menentukan aturan mana yang diterapkan. Semakin tinggi angkanya, semakin tinggi prioritasnya.

```
<ReplicationConfiguration>

  <Role>arn:aws:iam::account-id:role/role-name</Role>
```

```
  <Rule>
    <Status>Enabled</Status>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
       <Status>string</Status>
    </DeleteMarkerReplication>
    <Filter>
        <Prefix>star</Prefix>
    </Filter>
    <Destination>
      <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
    </Destination>
  </Rule>
  <Rule>
    <Status>Enabled</Status>
    <Priority>2</Priority>
    <DeleteMarkerReplication>
       <Status>string</Status>
    </DeleteMarkerReplication>
    <Filter>
        <Prefix>starship</Prefix>
    </Filter>
    <Destination>
      <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
    </Destination>
  </Rule>
</ReplicationConfiguration>
```

Example 4: Contoh panduan

Untuk contoh panduan, lihat Contoh untuk mengonfigurasi replikasi langsung.

Untuk informasi lebih lanjut tentang struktur XML konfigurasi replikasi, lihat PutBucketReplication dalam Referensi API Amazon Simple Storage Service.

Pertimbangan kompatibilitas mundur

Versi terbaru dari format XML-konfigurasi replikasi adalah V2. Konfigurasi replikasi XML V2 adalah yang berisi elemen `<Filter>` untuk aturan, dan aturan yang menentukan Kontrol Waktu Replikasi S3 (S3 RTC).

Untuk melihat versi konfigurasi replikasi, Anda dapat menggunakan operasi API `GetBucketReplication`. Untuk informasi selengkapnya, lihat GetBucketReplication dalam Referensi API Amazon Simple Storage Service.

Untuk kompatibilitas mundur, Amazon S3 terus mendukung format konfigurasi replikasi XMLV1. Jika Anda telah menggunakan format konfigurasi replikasi XMLV1, pertimbangkan masalah berikut yang memengaruhi kompatibilitas mundur:

- Format konfigurasi replikasi XMLV2 mencakup `<Filter>` elemen untuk aturan. Dengan elemen `<Filter>`, Anda dapat menentukan filter objek berdasarkan awalan kunci objek, tag, atau keduanya untuk mencakup objek yang menerapkan aturan tersebut. Format konfigurasi replikasi XMLV1 mendukung penyaringan hanya berdasarkan key prefix. Dalam hal ini, Anda menambahkan `<Prefix>` elemen secara langsung sebagai elemen anak dari `<Rule>` elemen, seperti pada contoh berikut:

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>
    <Prefix>key-prefix</Prefix>
    <Destination>
       <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
    </Destination>

  </Rule>
</ReplicationConfiguration>
```

- Saat Anda menghapus objek dari bucket sumber tanpa menentukan ID versi objek, Amazon S3 menambahkan penanda hapus. Jika Anda menggunakan konfigurasi replikasi format XMLV1, Amazon S3 mereplikasi hanya menghapus penanda yang dihasilkan dari tindakan pengguna. Dengan kata lain, Amazon S3 mereplikasi penanda hapus hanya jika pengguna menghapus objek. Jika objek kedaluwarsa dihapus oleh Amazon S3 (sebagai bagian dari tindakan siklus hidup), Amazon S3 tidak mereplikasi penanda hapus.

  Dalam konfigurasi replikasi format XMLV2, Anda dapat mengaktifkan replikasi penanda hapus untuk aturan. non-tag-based Untuk informasi selengkapnya, lihat Mereplikasi penanda hapus di antara bucket.

Menyiapkan izin untuk replikasi langsung

Saat menyiapkan replikasi langsung di Amazon S3, Anda harus memperoleh izin yang diperlukan sebagai berikut:

- Amazon S3 membutuhkan izin untuk mereplikasi objek atas nama Anda. Anda memberikan izin ini dengan membuat peran AWS Identity and Access Management (IAM) dan kemudian menentukan peran tersebut dalam konfigurasi replikasi Anda.
- Jika bucket sumber dan tujuan tidak dimiliki oleh akun yang sama, pemilik bucket tujuan juga harus memberikan izin pemilik bucket sumber untuk menyimpan replika.

Topik

- Menyiapkan izin untuk membuat aturan replikasi
- Membuat peran IAM
- Memberikan izin saat bucket sumber dan tujuan dimiliki oleh yang berbeda Akun AWS
- Mengubah kepemilikan replika
- Memberikan izin untuk Operasi Batch S3

Menyiapkan izin untuk membuat aturan replikasi

Pengguna atau peran IAM yang akan Anda gunakan untuk membuat aturan replikasi memerlukan izin untuk membuat aturan replikasi untuk replikasi satu atau dua arah. Jika pengguna atau peran tidak memiliki izin ini, Anda tidak akan dapat membuat aturan replikasi. Untuk informasi selengkapnya, lihat Identitas IAM di Panduan Pengguna IAM.

Pengguna atau peran membutuhkan tindakan berikut:

- `iam:AttachRolePolicy`
- `iam:CreatePolicy`
- `iam:CreateServiceLinkedRole`
- `iam:PassRole`
- `iam:PutRolePolicy`
- `s3:GetBucketVersioning`
- `s3:GetObjectVersionAcl`

- s3:GetObjectVersionForReplication
- s3:GetReplicationConfiguration
- s3:PutReplicationConfiguration

Berikut ini adalah contoh kebijakan IAM yang mencakup tindakan ini.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetAccessPoint",
                "s3:GetAccountPublicAccessBlock",
                "s3:GetBucketAcl",
                "s3:GetBucketLocation",
                "s3:GetBucketPolicyStatus",
                "s3:GetBucketPublicAccessBlock",
                "s3:ListAccessPoints",
                "s3:ListAllMyBuckets",
                "s3:PutReplicationConfiguration",
                "s3:GetReplicationConfiguration",
                "s3:GetBucketVersioning",
                "s3:GetObjectVersionForReplication",
                "s3:GetObjectVersionAcl",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:GetObjectVersion",
                "s3:GetBucketOwnershipControls",
                "s3:PutBucketOwnershipControls",
                "s3:GetObjectLegalHold",
                "s3:GetObjectRetention",
                "s3:GetBucketObjectLockConfiguration"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket1-*",
                "arn:aws:s3:::amzn-s3-demo-bucket2-*/*"
            ]
        },
        {
            "Effect": "Allow",
```

```
            "Action": [
                "s3:List*AccessPoint*",
                "s3:GetMultiRegion*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:Get*",
                "iam:CreateServiceLinkedRole",
                "iam:CreateRole",
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::*:role/service-role/s3*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:List*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:PutRolePolicy",
                "iam:CreatePolicy"
              ],
            "Resource": [
                "arn:aws:iam::*:policy/service-role/s3*",
                "arn:aws:iam::*:role/service-role/s3*"
            ]
        }
    ]
}
```

Membuat peran IAM

Secara default, semua sumber daya Amazon S3—bucket, objek, dan subsumber terkait—bersifat privat dan hanya pemilik sumber daya yang bisa mengakses sumber daya. Amazon S3

membutuhkan izin untuk membaca dan mereplikasi objek dari bucket sumber. Anda memberikan izin ini dengan membuat peran IAM dan menentukan peran tersebut dalam konfigurasi replikasi Anda.

Bagian ini menjelaskan kebijakan kepercayaan dan kebijakan izin minimum yang diperlukan yang dilampirkan pada peran IAM ini. Contoh penelusuran memberikan step-by-step instruksi untuk membuat peran IAM. Untuk informasi selengkapnya, lihat Contoh untuk mengonfigurasi replikasi langsung.

Kebijakan kepercayaan mengidentifikasi identitas utama mana yang dapat mengambil peran IAM. Kebijakan izin menentukan tindakan yang dapat dilakukan peran IAM, sumber daya mana, dan dalam kondisi apa.

- Contoh berikut menunjukkan kebijakan kepercayaan di mana Anda mengidentifikasi Amazon S3 sebagai Layanan AWS prinsipal yang dapat mengambil peran:

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"s3.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

- Contoh berikut menunjukkan kebijakan kepercayaan tempat Anda mengidentifikasi Operasi Batch Amazon S3 dan S3 sebagai prinsip layanan yang dapat mengambil peran tersebut. Gunakan pendekatan ini jika Anda membuat pekerjaan Replikasi Batch. Untuk informasi selengkapnya, lihat Buat pekerjaan Replikasi Batch untuk aturan atau tujuan replikasi baru.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service": [
              "s3.amazonaws.com",
              "batchoperations.s3.amazonaws.com"
```

```
            ]
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

Untuk informasi selengkapnya tentang peran IAM, lihat Peran IAM dalam Panduan Pengguna IAM.

- Contoh berikut menunjukkan kebijakan izin, tempat Anda memberikan izin peran IAM untuk melakukan tugas replikasi atas nama Anda. Saat Amazon S3 memegang peran tersebut, Amazon S3 memiliki izin yang Anda tentukan dalam kebijakan ini. Dalam kebijakan ini, *amzn-s3-demo-source-bucket* adalah bucket sumber, dan *amzn-s3-demo-destination-bucket* merupakan bucket tujuan.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetReplicationConfiguration",
            "s3:ListBucket"
         ],
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObjectVersionForReplication",
            "s3:GetObjectVersionAcl",
            "s3:GetObjectVersionTagging"
         ],
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:ReplicateObject",
```

```
            "s3:ReplicateDelete",
            "s3:ReplicateTags"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
      }
   ]
}
```

Kebijakan izin memberikan izin untuk tindakan berikut:

- `s3:GetReplicationConfiguration`dan `s3:ListBucket` — Izin untuk tindakan ini di *amzn-s3-demo-source-bucket* bucket memungkinkan Amazon S3 untuk mengambil konfigurasi replikasi dan mencantumkan konten bucket. (Model izin saat ini memerlukan izin `s3:ListBucket` untuk mengakses penanda hapus.)
- `s3:GetObjectVersionForReplication` dan `s3:GetObjectVersionAcl`–Izin untuk tindakan ini diberikan pada semua objek untuk memungkinkan Amazon S3 mendapatkan versi objek tertentu dan daftar kontrol akses (ACL) yang terkait dengan objek.

- `s3:ReplicateObject`dan `s3:ReplicateDelete` — Izin untuk tindakan ini pada semua objek di *amzn-s3-demo-destination-bucket* bucket memungkinkan Amazon S3 mereplikasi objek atau menghapus penanda ke bucket tujuan. Untuk informasi tentang penanda hapus, lihat Bagaimana cara menghapus operasi yang memengaruhi replikasi.

  > **Note**
  >
  > Izin untuk `s3:ReplicateObject` tindakan pada *amzn-s3-demo-destination-bucket* bucket juga memungkinkan replikasi metadata seperti tag objek dan. ACLs Oleh karena itu, Anda tidak perlu secara eksplisit memberikan izin untuk tindakan `s3:ReplicateTags` tersebut.

- `s3:GetObjectVersionTagging`— Izin untuk tindakan ini pada objek di *amzn-s3-demo-source-bucket* bucket memungkinkan Amazon S3 membaca tag objek untuk replikasi. Untuk informasi selengkapnya tentang tag objek, lihat Mengategorikan penyimpanan Anda menggunakan tag. Jika Amazon S3 tidak memiliki `s3:GetObjectVersionTagging` izin, itu mereplikasi objek, tetapi bukan tag objek.

Untuk daftar tindakan Amazon S3, lihat Tindakan, sumber daya, dan kunci kondisi untuk Amazon S3 di Referensi Otorisasi Layanan.

Untuk informasi selengkapnya tentang izin operasi S3 API menurut jenis sumber daya S3, lihat. Izin yang diperlukan untuk operasi API Amazon S3

> **Important**
>
> Akun AWS Yang memiliki peran IAM harus memiliki izin untuk tindakan yang diberikannya ke peran IAM.
> Misalnya, anggaplah bucket sumber berisi objek yang dimiliki oleh Akun AWS lain. Pemilik objek harus secara eksplisit memberikan Akun AWS yang memiliki peran IAM izin yang diperlukan melalui daftar kontrol akses objek (). ACLs Jika tidak, Amazon S3 tidak dapat mengakses objek, dan replikasi objek akan gagal. Untuk informasi tentang izin ACL, lihat Gambaran umum daftar kontrol akses (ACL).
> Izin yang dijelaskan di sini terkait dengan konfigurasi replikasi minimum. Jika Anda memilih untuk menambahkan konfigurasi replikasi opsional, Anda harus memberikan izin tambahan ke Amazon S3:
>
> - Untuk mereplikasi objek terenkripsi, Anda juga perlu memberikan izin kunci AWS Key Management Service (AWS KMS) yang diperlukan. Untuk informasi selengkapnya, lihat the section called “Mereplikasi objek terenkripsi”.
> - Untuk menggunakan Object Lock dengan replikasi, Anda harus memberikan dua izin tambahan pada bucket S3 sumber dalam peran AWS Identity and Access Management (IAM) yang Anda gunakan untuk mengatur replikasi. Dua izin tambahan tersebut adalah `s3:GetObjectRetention` dan`s3:GetObjectLegalHold`. Jika peran tersebut memiliki pernyataan izin `s3:Get*`, pernyataan itu memenuhi persyaratan. Untuk informasi selengkapnya, lihat the section called “Menggunakan Kunci Objek dengan Replikasi S3”.

Memberikan izin saat bucket sumber dan tujuan dimiliki oleh yang berbeda Akun AWS

Jika bucket sumber dan tujuan tidak dimiliki oleh akun yang sama, pemilik bucket tujuan juga harus menambahkan kebijakan bucket untuk memberikan izin kepada pemilik bucket sumber agar melakukan tindakan replikasi, seperti yang ditunjukkan pada contoh berikut. Dalam contoh kebijakan ini, *`amzn-s3-demo-destination-bucket`* adalah bucket tujuan.

Anda juga dapat menggunakan konsol Amazon S3 untuk membuat kebijakan bucket ini secara otomatis untuk Anda. Untuk informasi selengkapnya, lihat Mengaktifkan penerimaan objek yang direplikasi dari keranjang sumber.

> **Note**
>
> Format ARN peran mungkin tampak berbeda. Jika peran dibuat dengan menggunakan konsol, format ARN adalah. `arn:aws:iam::account-ID:role/service-role/role-name` Jika peran dibuat dengan menggunakan AWS CLI, format ARN adalah. `arn:aws:iam::account-ID:role/role-name` Untuk informasi selengkapnya, lihat peran IAM dalam Panduan Pengguna IAM.

```
{
   "Version":"2012-10-17",
   "Id":"PolicyForDestinationBucket",
   "Statement":[
      {
         "Sid":"Permissions on objects",
         "Effect":"Allow",
         "Principal":{
            "AWS":"arn:aws:iam::source-bucket-account-ID:role/service-role/source-
account-IAM-role"
         },
         "Action":[
            "s3:ReplicateDelete",
            "s3:ReplicateObject"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
      },
      {
         "Sid":"Permissions on bucket",
         "Effect":"Allow",
         "Principal":{
            "AWS":"arn:aws:iam::source-bucket-account-ID:role/service-role/source-
account-IAM-role"
         },
         "Action": [
            "s3:List*",
            "s3:GetBucketVersioning",
            "s3:PutBucketVersioning"
         ],
         "Resource":"arn:aws:s3::amzn-s3-demo-destination-bucket"
      }
   ]
```

```
}
```

Sebagai contoh, lihat Mengkonfigurasi replikasi untuk bucket di akun yang berbeda.

Jika objek dalam bucket sumber ditandai, perhatikan hal berikut:

- Jika pemilik bucket sumber memberikan izin kepada Amazon S3 untuk tindakan `s3:GetObjectVersionTagging` dan `s3:ReplicateTags` guna mereplikasi tag objek (melalui peran IAM), Amazon S3 mereplikasi tag beserta objek. Untuk informasi tentang peran IAM, lihat Membuat peran IAM.
- Jika pemilik bucket tujuan tidak ingin mereplikasi tag, mereka dapat menambahkan pernyataan berikut ke kebijakan bucket tujuan untuk secara eksplisit menolak izin bagi tindakan `s3:ReplicateTags`. Dalam kebijakan ini, *`amzn-s3-demo-destination-bucket`* adalah bucket tujuan.

```
...
   "Statement":[
      {
         "Effect":"Deny",
         "Principal":{
            "AWS":"arn:aws:iam::source-bucket-account-id:role/service-role/source-account-IAM-role"
         },
         "Action":"s3:ReplicateTags",
         "Resource":"arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
      }
   ]
...
```

> **Note**
>
> - Jika Anda ingin mereplikasi objek terenkripsi, Anda juga harus memberikan izin kunci AWS Key Management Service (AWS KMS) yang diperlukan. Untuk informasi selengkapnya, lihat the section called “Mereplikasi objek terenkripsi”.
> - Untuk menggunakan Object Lock dengan replikasi, Anda harus memberikan dua izin tambahan pada bucket S3 sumber dalam peran AWS Identity and Access Management (IAM) yang Anda gunakan untuk mengatur replikasi. Dua izin tambahan tersebut adalah `s3:GetObjectRetention` dans`3:GetObjectLegalHold`. Jika peran tersebut

memiliki pernyataan izin `s3:Get*`, pernyataan itu memenuhi persyaratan. Untuk informasi selengkapnya, lihat the section called “Menggunakan Kunci Objek dengan Replikasi S3”.

Aktifkan menerima objek yang direplikasi dari bucket sumber

Alih-alih menambahkan kebijakan sebelumnya secara manual ke bucket tujuan, Anda dapat dengan cepat membuat kebijakan yang diperlukan untuk mengaktifkan penerimaan objek yang direplikasi dari bucket sumber melalui konsol Amazon S3.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Di daftar Bucket, pilih bucket yang ingin Anda gunakan sebagai bucket tujuan.
4. Pilih tab Manajemen, lalu gulir ke bawah ke Aturan replikasi.
5. Untuk Tindakan, pilih Menerima objek yang direplikasi.

   Ikuti petunjuknya dan masukkan Akun AWS ID akun bucket sumber, lalu pilih Buat kebijakan. Konsol menghasilkan kebijakan bucket Amazon S3 dan kebijakan kunci KMS.
6. Untuk menambahkan kebijakan ini ke kebijakan bucket yang ada, pilih Terapkan pengaturan atau pilih Salin untuk menyalin perubahan secara manual.
7. (Opsional) Salin AWS KMS kebijakan ke kebijakan kunci KMS yang Anda inginkan di AWS Key Management Service konsol.

Mengubah kepemilikan replika

Jika berbeda Akun AWS memiliki bucket sumber dan tujuan, Anda dapat memberi tahu Amazon S3 untuk mengubah kepemilikan replika ke bucket Akun AWS yang memiliki tujuan. Untuk informasi selengkapnya tentang penggantian pemilik, lihat Mengubah pemilik replika.

Memberikan izin untuk Operasi Batch S3

Replikasi Batch S3 memberi Anda cara untuk mereplikasi objek berikut:

- Objek yang ada sebelum konfigurasi replikasi ada
- Objek yang sebelumnya telah direplikasi
- Objek yang gagal replikasi

Anda dapat membuat tugas Replikasi Batch satu kali saat membuat aturan pertama dalam konfigurasi replikasi baru atau saat menambahkan tujuan baru ke konfigurasi yang ada melalui konsol Amazon S3. Anda juga dapat memulai Replikasi Batch untuk konfigurasi replikasi yang ada dengan membuat pekerjaan Operasi Batch.

Untuk contoh peran dan kebijakan IAM Replikasi Batch, lihat. Mengkonfigurasi peran IAM untuk Replikasi Batch S3

Contoh untuk mengonfigurasi replikasi langsung

Contoh berikut memberikan step-by-step panduan yang menunjukkan cara mengonfigurasi replikasi langsung untuk kasus penggunaan umum.

> **Note**
>
> Replikasi langsung mengacu pada Replikasi Wilayah yang Sama (SRR) dan Replikasi Lintas-Wilayah (CRR). Replikasi langsung tidak mereplikasi objek apa pun yang ada di bucket sebelum Anda mengatur replikasi. Untuk mereplikasi objek yang ada sebelum Anda mengatur replikasi, gunakan replikasi sesuai permintaan. Untuk menyinkronkan bucket dan mereplikasi objek yang ada sesuai permintaan, lihat. Mereplikasi objek yang ada

Contoh-contoh ini menunjukkan cara membuat konfigurasi replikasi dengan menggunakan konsol Amazon S3 AWS Command Line Interface ,AWS CLI(), AWS SDKs dan AWS SDK untuk Java (AWS SDK untuk .NET dan contoh ditampilkan).

Untuk informasi tentang menginstal dan mengonfigurasi AWS CLI, lihat topik berikut di Panduan AWS Command Line Interface Pengguna:

- Memulai dengan AWS CLI
- Konfigurasikan AWS CLI — Anda harus mengatur setidaknya satu profil. Jika Anda menjelajahi skenario lintas akun, siapkan dua profil.

Untuk informasi tentang AWS SDKs, lihat AWS SDK untuk Javadan AWS SDK untuk .NET.

> **Tip**
>
> Untuk step-by-step tutorial yang menunjukkan cara menggunakan replikasi langsung untuk mereplikasi data, lihat Tutorial: Mereplikasi data di dalam dan di antara Wilayah AWS menggunakan Replikasi S3.

Topik

- Mengkonfigurasi replikasi untuk bucket di akun yang sama
- Mengkonfigurasi replikasi untuk bucket di akun yang berbeda
- Memenuhi persyaratan kepatuhan dengan Kontrol Waktu Replikasi S3
- Mereplikasi objek terenkripsi (SSE-S3, SSE-KMS, DSSE-KMS, SSE-C)
- Mereplikasi perubahan metadata dengan sinkronisasi modifikasi replika
- Mereplikasi penanda hapus di antara bucket

Mengkonfigurasi replikasi untuk bucket di akun yang sama

Replikasi langsung adalah penyalinan objek otomatis dan asinkron di seluruh ember tujuan umum dalam hal yang sama atau berbeda. Wilayah AWS Replikasi langsung menyalin objek dan pembaruan objek yang baru dibuat dari bucket sumber ke bucket atau bucket tujuan. Untuk informasi selengkapnya, lihat Mereplikasi objek di dalam dan di seluruh Wilayah.

Saat mengonfigurasi replikasi, Anda menambahkan aturan replikasi ke bucket sumber. Aturan replikasi menentukan objek bucket sumber mana yang akan direplikasi dan bucket tujuan atau bucket tempat objek yang direplikasi disimpan. Anda dapat membuat aturan untuk mereplikasi semua objek dalam bucket atau subset objek dengan awalan nama kunci tertentu, satu atau beberapa tag objek, atau keduanya. Bucket tujuan bisa Akun AWS sama dengan bucket sumber, atau bisa juga di akun yang berbeda.

Jika Anda menentukan ID versi objek yang akan dihapus, Amazon S3 menghapus versi objek tersebut dalam bucket sumber. Tetapi ini tidak mereplikasi penghapusan di bucket tujuan. Dengan kata lain, itu tidak menghapus versi objek yang sama dari bucket tujuan. Ini melindungi data dari penghapusan berbahaya.

Ketika Anda menambahkan aturan replikasi ke bucket, aturan diaktifkan secara default, sehingga aturan mulai bekerja segera setelah Anda menyimpannya.

Dalam contoh ini, Anda mengatur replikasi langsung untuk bucket sumber dan tujuan yang dimiliki oleh yang sama. Akun AWS Contoh disediakan untuk menggunakan konsol Amazon S3, AWS Command Line Interface (AWS CLI), dan dan. AWS SDK untuk Java AWS SDK untuk .NET

Prasyarat

Sebelum Anda menggunakan prosedur berikut, pastikan Anda telah menyiapkan izin yang diperlukan untuk replikasi, tergantung pada apakah bucket sumber dan tujuan dimiliki oleh akun yang sama atau berbeda. Untuk informasi selengkapnya, lihat the section called “Menyiapkan izin”.

> **Note**
>
> - Jika Anda ingin mereplikasi objek terenkripsi, Anda juga harus memberikan izin kunci AWS Key Management Service (AWS KMS) yang diperlukan. Untuk informasi selengkapnya, lihat the section called “Mereplikasi objek terenkripsi”.
> - Untuk menggunakan Object Lock dengan replikasi, Anda harus memberikan dua izin tambahan pada bucket S3 sumber dalam peran AWS Identity and Access Management (IAM) yang Anda gunakan untuk mengatur replikasi. Dua izin tambahan tersebut adalah `s3:GetObjectRetention` dan`s3:GetObjectLegalHold`. Jika peran tersebut memiliki pernyataan izin `s3:Get*`, pernyataan itu memenuhi persyaratan. Untuk informasi selengkapnya, lihat the section called “Menggunakan Kunci Objek dengan Replikasi S3”.

Menggunakan konsol S3

Untuk mengonfigurasi aturan replikasi saat bucket tujuan Akun AWS sama dengan bucket sumber, ikuti langkah-langkah berikut.

Jika bucket tujuan berada di akun yang berbeda dengan bucket sumber, Anda harus menambahkan kebijakan bucket ke bucket tujuan untuk memberi pemilik akun bucket sumber izin untuk mereplikasi objek dalam bucket tujuan. Untuk informasi selengkapnya, lihat Memberikan izin saat bucket sumber dan tujuan dimiliki oleh yang berbeda Akun AWS.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Dalam daftar bucket, pilih nama bucket yang Anda inginkan.
4. Pilih tab Manajemen, gulir ke bawah ke Aturan replikasi, lalu pilih Buat aturan replikasi.

5. Di bagian Konfigurasi aturan replikasi, di bawah Nama aturan replikasi, masukkan nama untuk aturan Anda guna membantu mengidentifikasi aturan tersebut nanti. Nama wajib diisi dan harus unik dalam bucket.
6. Di bawah Status, Diaktifkan dipilih secara default. Aturan yang diaktifkan mulai berfungsi segera setelah Anda menyimpannya. Jika Anda ingin mengaktifkan aturan nanti, pilih Dinonaktifkan.
7. Jika bucket memiliki aturan replikasi yang ada, Anda diinstruksikan untuk menetapkan prioritas aturan. Anda harus menetapkan prioritas pada aturan untuk menghindari konflik yang disebabkan oleh objek yang dimasukkan dalam cakupan lebih dari satu aturan. Dalam kasus aturan yang tumpang tindih, Amazon S3 menggunakan prioritas aturan untuk menentukan aturan mana yang berlaku. Semakin tinggi angkanya, semakin tinggi prioritasnya. Untuk informasi selengkapnya tentang prioritas aturan, lihat Elemen file konfigurasi replikasi.
8. Di bawah Bucket sumber, Anda memiliki opsi berikut untuk mengatur sumber replikasi:

   - Untuk mereplikasi seluruh bucket, pilih Terapkan ke semua objek di bucket.
   - Untuk mereplikasi semua objek yang memiliki awalan yang sama, pilih Batasi cakupan aturan ini menggunakan satu atau beberapa filter. Ini membatasi replikasi ke semua objek yang memiliki nama yang dimulai dengan awalan yang Anda tentukan (misalnya `pictures`). Masukkan awalan di kotak Awalan.

     > **Note**
     >
     > Jika Anda memasukkan awalan yang merupakan nama folder, Anda harus menggunakan / (garis miring ke depan) sebagai karakter terakhir (misalnya, `pictures/`).

   - Untuk mereplikasi semua objek dengan satu atau beberapa tag objek, pilih Tambahkan tag dan masukkan pasangan nilai kunci ke dalam kotak. Ulangi prosedur untuk menambahkan tag lainnya. Anda dapat menggabungkan awalan dan tag. Untuk informasi selengkapnya tentang tag objek, lihat Mengategorikan penyimpanan Anda menggunakan tag.

   Skema XML konfigurasi replikasi baru mendukung pemfilteran awalan dan tag serta prioritas aturan. Untuk informasi selengkapnya tentang skema baru, lihat Pertimbangan kompatibilitas mundur. Untuk informasi selengkapnya tentang XML yang digunakan dengan API Amazon S3 yang berfungsi di balik antarmuka pengguna, lihat Elemen file konfigurasi replikasi. Skema baru dijelaskan sebagai konfigurasi replikasi XML V2.

9. Di bawah Tujuan, pilih bucket tempat Anda ingin Amazon S3 mereplikasi objek.

> **Note**
>
> Jumlah ember tujuan terbatas pada jumlah Wilayah AWS di partisi tertentu. Partisi adalah pengelompokan Wilayah. AWS Saat ini memiliki tiga partisi: `aws` (Wilayah Standar), `aws-cn` (Wilayah Tiongkok), dan `aws-us-gov` (AWS GovCloud (US) Wilayah). Untuk meminta peningkatan kuota bucket tujuan Anda, Anda dapat menggunakan kuota layanan.

- Untuk mereplikasi ke bucket atau beberapa bucket di akun Anda, pilih Pilih bucket di akun ini, dan masukkan atau telusuri nama bucket tujuan.
- Untuk mereplikasi ke bucket atau bucket yang berbeda Akun AWS, pilih Tentukan bucket di akun lain, lalu masukkan ID akun bucket tujuan dan nama bucket.

  Jika tujuan berada di akun yang berbeda dengan bucket sumber, Anda harus menambahkan kebijakan bucket ke bucket tujuan untuk memberi pemilik akun bucket sumber izin untuk mereplikasi objek. Untuk informasi selengkapnya, lihat Memberikan izin saat bucket sumber dan tujuan dimiliki oleh yang berbeda Akun AWS.

  Secara opsional, jika Anda ingin membantu menstandarkan kepemilikan objek baru di bucket tujuan, pilih Ubah kepemilikan objek ke pemilik bucket tujuan. Untuk informasi selengkapnya tentang metrik ini, lihat Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.

> **Note**
>
> Jika Penentuan Versi tidak diaktifkan pada bucket tujuan, Anda mendapatkan peringatan yang berisi tombol Aktifkan Penentuan Versi. Pilih tombol ini untuk mengaktifkan Penentuan Versi pada bucket.

10. Siapkan peran AWS Identity and Access Management (IAM) yang dapat diasumsikan Amazon S3 untuk mereplikasi objek atas nama Anda.

    Untuk menyiapkan peran IAM, di bagian peran IAM, pilih salah satu dari daftar dropdown peran IAM berikut ini:

- Kami sangat merekomendasikan Anda untuk memilih Buat peran baru agar Amazon S3 membuat peran IAM baru untuk Anda. Saat Anda menyimpan aturan, kebijakan baru akan dibuat untuk peran IAM yang sesuai dengan bucket sumber dan tujuan yang Anda pilih.
- Anda dapat memilih untuk menggunakan peran IAM yang sudah ada. Jika melakukannya, Anda harus memilih peran yang memberi Amazon S3 izin yang diperlukan untuk replikasi. Replikasi akan gagal jika peran ini tidak memberi Amazon S3 izin yang memadai untuk mengikuti aturan replikasi Anda.

> **Important**
>
> Saat menambahkan aturan replikasi ke bucket, Anda harus memiliki izin `iam:PassRole` untuk dapat meneruskan peran IAM yang memberi izin Replikasi Amazon S3. Untuk informasi selengkapnya, lihat Memberikan izin pengguna untuk meneruskan peran ke Layanan AWS di Panduan Pengguna IAM.

11. Untuk mereplikasi objek di bucket sumber yang dienkripsi dengan enkripsi sisi server dengan kunci AWS Key Management Service (AWS KMS) (SSE-KMS), di bawah Enkripsi, pilih Replikasi objek yang dienkripsi dengan. AWS KMS Di bawah AWS KMS kunci untuk mengenkripsi objek tujuan adalah kunci sumber yang dapat digunakan replikasi. Semua kunci sumber KMS disertakan secara default. Untuk mempersempit pilihan kunci KMS, Anda dapat memilih alias atau ID kunci.

    Objek yang dienkripsi oleh AWS KMS keys yang tidak Anda pilih tidak direplikasi. Kunci KMS atau kumpulan kunci KMS akan dipilih untuk Anda, tetapi Anda dapat memilih kunci KMS jika mau. Untuk informasi tentang menggunakan AWS KMS dengan replikasi, lihatMereplikasi objek terenkripsi (SSE-S3, SSE-KMS, DSSE-KMS, SSE-C).

    > **Important**
    >
    > Saat Anda mereplikasi objek yang dienkripsi AWS KMS, tingkat AWS KMS permintaan berlipat ganda di Wilayah sumber dan meningkat di Wilayah tujuan dengan jumlah yang sama. Tingkat panggilan yang meningkat ini disebabkan oleh cara data dienkripsi ulang dengan menggunakan kunci KMS yang Anda tentukan untuk Wilayah tujuan replikasi. AWS KMS AWS KMS memiliki kuota tarif permintaan per rekening panggilan per Region. Untuk informasi tentang default kuota, lihat Kuota AWS KMS -Permintaan per Detik: Bervariasi di Panduan Developer AWS Key Management Service .

Jika tingkat permintaan PUT objek Amazon S3 saat ini selama replikasi lebih dari setengah batas AWS KMS tarif default untuk akun Anda, kami sarankan Anda meminta kenaikan kuota tingkat AWS KMS permintaan Anda. Untuk meminta peningkatan, buat kasus di Pusat Dukungan di Hubungi Kami. Misalnya, misalkan tingkat permintaan PUT objek Anda saat ini adalah 1.000 permintaan per detik dan Anda gunakan AWS KMS untuk mengenkripsi objek Anda. Dalam hal ini, kami menyarankan agar Anda meminta Dukungan untuk meningkatkan batas AWS KMS tarif Anda menjadi 2.500 permintaan per detik, baik di Wilayah sumber maupun tujuan Anda (jika berbeda), untuk memastikan bahwa tidak ada pembatasan oleh. AWS KMS
Untuk melihat rasio permintaan PUT objek Anda di bucket sumber, lihat `PutRequests` di metrik CloudWatch permintaan Amazon untuk Amazon S3. Untuk informasi tentang melihat CloudWatch metrik, lihatMenggunakan konsol S3.

Jika Anda memilih untuk mereplikasi objek yang dienkripsi AWS KMS, lakukan hal berikut:

- Di bawah AWS KMS key untuk mengenkripsi objek tujuan, tentukan kunci KMS Anda dengan salah satu cara berikut:

  - Untuk memilih dari daftar kunci KMS yang tersedia, pilih Pilih dari AWS KMS keys, dan pilih Kunci KMS Anda dari daftar kunci yang tersedia.

    Kunci Kunci yang dikelola AWS (`aws/s3`) dan kunci terkelola pelanggan Anda muncul dalam daftar ini. Untuk informasi selengkapnya tentang CMK, lihat Kunci pelanggan dan AWS kunci di AWS Key Management Service Panduan Pengembang.
  - Untuk memasukkan kunci KMS Amazon Resource Name (ARN), pilih Masukkan ARN AWS KMS key , dan masukkan ARN kunci KMS Anda di bidang yang muncul. Ini akan mengenkripsi replika di bucket tujuan. Anda dapat menemukan ARN untuk kunci KMS Anda di Konsol IAM, di bawah kunci Enkripsi.
  - Untuk membuat kunci terkelola pelanggan baru di AWS KMS konsol, pilih Buat kunci KMS.

    Untuk informasi selengkapnya tentang membuat AWS KMS key, lihat Membuat kunci di Panduan AWS Key Management Service Pengembang.

> **Important**
>
> Anda hanya dapat menggunakan tombol KMS yang diaktifkan Wilayah AWS sama dengan bucket. Saat memilih Pilih dari kunci KMS Anda, konsol S3 hanya mencantumkan 100 kunci KMS per Wilayah. Jika Anda memiliki lebih dari 100 tombol KMS di Wilayah yang sama, Anda hanya dapat melihat 100 kunci KMS pertama di konsol S3. Untuk menggunakan kunci KMS yang tidak tercantum di konsol, pilih Masukkan ARN AWS KMS key , lalu masukkan ARN kunci KMS Anda. Saat Anda menggunakan enkripsi sisi server AWS KMS key untuk Amazon S3, Anda harus memilih kunci KMS enkripsi simetris. Amazon S3 hanya mendukung kunci KMS enkripsi simetris dan tidak mendukung kunci KMS asimetris. Untuk informasi selengkapnya, lihat Mengidentifikasi tombol KMS simetris dan asimetris dalam Panduan Pengembang AWS Key Management Service .

Untuk informasi selengkapnya tentang membuat AWS KMS key, lihat Membuat kunci di Panduan AWS Key Management Service Pengembang. Untuk informasi selengkapnya tentang penggunaan AWS KMS dengan Amazon S3, lihat. Menggunakan enkripsi sisi server dengan AWS KMS kunci (SSE-KMS)

12. Di Bawah Kelas penyimpanan tujuan, jika Anda ingin mereplikasi data Anda ke dalam kelas penyimpanan tertentu di tujuan, pilih Ubah kelas penyimpanan untuk objek yang direplikasi. Kemudian pilih kelas penyimpanan yang ingin Anda gunakan untuk objek yang direplikasi di tujuan. Jika Anda tidak memilih opsi ini, kelas penyimpanan untuk objek yang direplikasi adalah kelas yang sama dengan objek aslinya.
13. Anda memiliki opsi tambahan berikut saat mengatur Opsi replikasi tambahan:

    - Jika Anda ingin mengaktifkan Kontrol Waktu Replikasi S3 (S3 RTC) dalam konfigurasi replikasi Anda, pilih Kontrol Waktu Replikasi (RTC). Untuk informasi selengkapnya tentang metrik ini, lihat Memenuhi persyaratan kepatuhan dengan Kontrol Waktu Replikasi S3.
    - Jika Anda ingin mengaktifkan metrik replikasi S3 dalam konfigurasi replikasi Anda, pilih Metrik dan peristiwa replikasi. Untuk informasi selengkapnya, lihat Memantau replikasi dengan metrik, pemberitahuan acara, dan status.
    - Jika Anda ingin mengaktifkan replikasi penanda hapus dalam konfigurasi replikasi Anda, pilih Replikasi penanda hapus. Untuk informasi selengkapnya, lihat Mereplikasi penanda hapus di antara bucket.

- Jika Anda ingin mengaktifkan sinkronisasi modifikasi replika Amazon S3 dalam konfigurasi replikasi Anda, pilih Sinkronisasi modifikasi replika. Untuk informasi selengkapnya, lihat Mereplikasi perubahan metadata dengan sinkronisasi modifikasi replika.

> **Note**
>
> Ketika Anda menggunakan metrik Replikasi S3 atau S3 RTC, biaya tambahan berlaku.

14. Untuk menyelesaikan, pilih Simpan.
15. Setelah menyimpan aturan, Anda dapat mengedit, mengaktifkan, menonaktifkan, atau menghapus aturan Anda dengan memilih aturan dan memilih Edit aturan.

Menggunakan AWS CLI

Untuk menggunakan AWS CLI untuk mengatur replikasi ketika bucket sumber dan tujuan dimiliki oleh yang sama Akun AWS, Anda melakukan hal berikut:

- Buat ember sumber dan tujuan.
- Aktifkan pembuatan versi pada ember.
- Buat peran AWS Identity and Access Management (IAM) yang memberikan izin Amazon S3 untuk mereplikasi objek.
- Tambahkan konfigurasi replikasi ke bucket sumber.

Untuk memverifikasi konfigurasi, Anda mengujinya.

Untuk mengatur replikasi ketika bucket sumber dan tujuan dimiliki oleh yang sama Akun AWS

1. Atur profil kredensial untuk AWS CLI. Contoh ini menggunakan nama profil `acctA`. Untuk informasi tentang menyetel profil kredensi dan menggunakan profil bernama, lihat Pengaturan konfigurasi dan file kredensi di AWS Command Line Interface Panduan Pengguna.

> **Important**
>
> Profil yang Anda gunakan untuk contoh ini harus memiliki izin yang diperlukan. Misalnya, dalam konfigurasi replikasi, Anda menentukan peran IAM yang dapat diasumsikan oleh Amazon S3. Anda dapat melakukan ini hanya jika profil yang Anda gunakan memiliki

`iam:PassRole` izin. Untuk informasi selengkapnya, lihat Memberikan izin pengguna untuk meneruskan peran ke Layanan AWS dalam Panduan Pengguna IAM. Jika Anda menggunakan kredensial administrator untuk membuat profil bernama, Anda dapat melakukan semua tugas.

2. Buat bucket sumber dan aktifkan pembuatan versi di atasnya dengan menggunakan perintah berikut AWS CLI . Untuk menggunakan perintah ini, ganti *user input placeholders* dengan informasi Anda sendiri.

   `create-bucket`Perintah berikut membuat bucket sumber bernama *amzn-s3-demo-source-bucket* di Wilayah AS Timur (Virginia N.) (`us-east-1`):

   ```
   aws s3api create-bucket \
   --bucket amzn-s3-demo-source-bucket \
   --region us-east-1 \
   --profile acctA
   ```

   `put-bucket-versioning`Perintah berikut memungkinkan S3 Versioning pada bucket: *amzn-s3-demo-source-bucket*

   ```
   aws s3api put-bucket-versioning \
   --bucket amzn-s3-demo-source-bucket \
   --versioning-configuration Status=Enabled \
   --profile acctA
   ```

3. Buat bucket tujuan dan aktifkan pembuatan versi dengan menggunakan perintah berikut AWS CLI . Untuk menggunakan perintah ini, ganti *user input placeholders* dengan informasi Anda sendiri.

   > **Note**
   >
   > Untuk menyiapkan konfigurasi replikasi saat bucket sumber dan tujuan berada dalam kondisi yang sama Akun AWS, Anda menggunakan profil yang sama untuk bucket sumber dan tujuan. Contoh ini menggunakan `acctA`.
   > Untuk menguji konfigurasi replikasi ketika bucket dimiliki oleh yang berbeda Akun AWS, tentukan profil yang berbeda untuk setiap akun. Misalnya, gunakan `acctB` profil untuk bucket tujuan.

`create-bucket`Perintah berikut membuat bucket tujuan bernama *amzn-s3-demo-destination-bucket* di Wilayah AS Barat (Oregon) (`us-west-2`):

```
aws s3api create-bucket \
--bucket amzn-s3-demo-destination-bucket \
--region us-west-2 \
--create-bucket-configuration LocationConstraint=us-west-2 \
--profile acctA
```

`put-bucket-versioning`Perintah berikut memungkinkan S3 Versioning pada bucket: *amzn-s3-demo-destination-bucket*

```
aws s3api put-bucket-versioning \
--bucket amzn-s3-demo-destination-bucket \
--versioning-configuration Status=Enabled \
--profile acctA
```

4. Buat peran IAM. Anda menentukan peran ini dalam konfigurasi replikasi yang Anda tambahkan ke bucket sumber nanti. Amazon S3 mengasumsikan peran ini untuk mereplikasi objek atas nama Anda. Anda membuat peran IAM dalam dua langkah:

   - Buat peran.
   - Lampirkan kebijakan izin pada peran tersebut.

   a. Buat peran IAM.

      i. Salin kebijakan kepercayaan berikut dan simpan di berkas dengan nama `s3-role-trust-policy.json` di direktori saat ini di komputer lokal Anda. Kebijakan ini memberikan izin utama layanan Amazon S3 untuk mengambil peran.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"s3.amazonaws.com"
         },
```

```
            "Action":"sts:AssumeRole"
        }
    ]
}
```

ii. Jalankan perintah berikut untuk membuat peran.

```
$ aws iam create-role \
--role-name replicationRole \
--assume-role-policy-document file://s3-role-trust-policy.json  \
--profile acctA
```

b. Lampirkan kebijakan izin pada peran tersebut.

i. Salin kebijakan kepercayaan berikut dan simpan di berkas dengan nama `s3-role-permissions-policy.json` di direktori saat ini di komputer lokal Anda. Kebijakan ini memberikan izin untuk berbagai bucket dan tindakan objek Amazon S3.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObjectVersionForReplication",
            "s3:GetObjectVersionAcl",
            "s3:GetObjectVersionTagging"
         ],
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:ListBucket",
            "s3:GetReplicationConfiguration"
         ],
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket"
         ]
      },
      {
```

```
            "Effect":"Allow",
            "Action":[
               "s3:ReplicateObject",
               "s3:ReplicateDelete",
               "s3:ReplicateTags"
            ],
            "Resource":"arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
         }
      ]
}
```

> **Note**
>
> - Jika Anda ingin mereplikasi objek terenkripsi, Anda juga harus memberikan izin kunci AWS Key Management Service (AWS KMS) yang diperlukan. Untuk informasi selengkapnya, lihat the section called “Mereplikasi objek terenkripsi”.
> - Untuk menggunakan Object Lock dengan replikasi, Anda harus memberikan dua izin tambahan pada bucket S3 sumber dalam peran AWS Identity and Access Management (IAM) yang Anda gunakan untuk mengatur replikasi. Dua izin tambahan tersebut adalah `s3:GetObjectRetention` dan`s3:GetObjectLegalHold`. Jika peran tersebut memiliki pernyataan izin `s3:Get*`, pernyataan itu memenuhi persyaratan. Untuk informasi selengkapnya, lihat the section called “Menggunakan Kunci Objek dengan Replikasi S3”.

ii. Jalankan perintah berikut untuk membuat kebijakan dan melampirkannya ke peran. Ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
$ aws iam put-role-policy \
--role-name replicationRole \
--policy-document file://s3-role-permissions-policy.json \
--policy-name replicationRolePolicy \
--profile acctA
```

5. Tambahkan konfigurasi replikasi ke bucket sumber.

   a. Meskipun Amazon S3 API mengharuskan Anda menentukan konfigurasi replikasi sebagai XHTML, Anda harus menentukan konfigurasi replikasi sebagai JSON. AWS CLI Simpan

JSON berikut dalam file yang disebut `replication.json` ke direktori lokal di komputer Anda.

```
{
  "Role": "IAM-role-ARN",
  "Rules": [
    {
      "Status": "Enabled",
      "Priority": 1,
      "DeleteMarkerReplication": { "Status": "Disabled" },
      "Filter" : { "Prefix": "Tax"},
      "Destination": {
        "Bucket": "arn:aws:s3:::amzn-s3-demo-destination-bucket"
      }
    }
  ]
}
```

b. Perbarui JSON dengan mengganti nilai untuk *amzn-s3-demo-destination-bucket* dan *IAM-role-ARN* dengan informasi Anda sendiri. Simpan perubahan.

c. Jalankan `put-bucket-replication` perintah berikut untuk menambahkan konfigurasi replikasi ke bucket sumber Anda. Pastikan untuk memberikan nama bucket sumber:

```
$ aws s3api put-bucket-replication \
--replication-configuration file://replication.json \
--bucket amzn-s3-demo-source-bucket \
--profile acctA
```

Untuk mengambil konfigurasi replikasi, gunakan perintah: `get-bucket-replication`

```
$ aws s3api get-bucket-replication \
--bucket amzn-s3-demo-source-bucket \
--profile acctA
```

6. Uji pengaturan di konsol Amazon S3, dengan melakukan langkah-langkah berikut:

a. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

b. Di panel navigasi kiri, pilih Bucket. Dalam daftar Bucket tujuan umum, pilih bucket sumber.

c. Di bucket sumber, buat folder bernama*Tax*.

d. Tambahkan objek sampel ke *Tax* folder di ember sumber.

> **Note**
>
> Jumlah waktu yang dibutuhkan Amazon S3 untuk mereplikasi objek tergantung pada ukuran objek. Untuk informasi tentang cara melihat status replikasi, lihat Mendapatkan informasi status replikasi.

Di bucket tujuan, pastikan hal berikut:

- Bahwa Amazon S3 mereplikasi objek.
- Bahwa benda-benda itu adalah replika. Pada tab Properties untuk objek Anda, gulir ke bawah ke bagian Ikhtisar manajemen Object. Di bawah Konfigurasi manajemen, lihat nilai di bawah Status replikasi. Pastikan bahwa nilai ini diatur ke`REPLICA`.
- Bahwa replika dimiliki oleh akun bucket sumber. Anda dapat memverifikasi kepemilikan objek pada tab Izin untuk objek Anda.

  Jika bucket sumber dan tujuan dimiliki oleh akun yang berbeda, Anda dapat menambahkan konfigurasi opsional untuk memberi tahu Amazon S3 agar mengubah kepemilikan replika ke akun tujuan. Sebagai contoh, lihat Cara mengganti pemilik replika.

Menggunakan AWS SDKs

Gunakan contoh kode berikut untuk menambahkan konfigurasi replikasi ke bucket dengan AWS SDK untuk Java dan AWS SDK untuk .NET, masing-masing.

> **Note**
>
> - Jika Anda ingin mereplikasi objek terenkripsi, Anda juga harus memberikan izin kunci AWS Key Management Service (AWS KMS) yang diperlukan. Untuk informasi selengkapnya, lihat the section called “Mereplikasi objek terenkripsi”.
> - Untuk menggunakan Object Lock dengan replikasi, Anda harus memberikan dua izin tambahan pada bucket S3 sumber dalam peran AWS Identity and Access Management (IAM) yang Anda gunakan untuk mengatur replikasi. Dua izin tambahan tersebut adalah `s3:GetObjectRetention` dan`s3:GetObjectLegalHold`. Jika peran tersebut

memiliki pernyataan izin `s3:Get*`, pernyataan itu memenuhi persyaratan. Untuk informasi selengkapnya, lihat the section called “Menggunakan Kunci Objek dengan Replikasi S3”.

Java

Contoh berikut ini menambahkan konfigurasi replikasi ke bucket dan kemudian mengambil dan memverifikasi konfigurasi. Untuk petunjuk cara membuat dan menguji sampel yang berfungsi, lihat Memulai di Panduan AWS SDK untuk Java Pengembang.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.identitymanagement.AmazonIdentityManagement;
import
 com.amazonaws.services.identitymanagement.AmazonIdentityManagementClientBuilder;
import com.amazonaws.services.identitymanagement.model.CreateRoleRequest;
import com.amazonaws.services.identitymanagement.model.PutRolePolicyRequest;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.model.BucketReplicationConfiguration;
import com.amazonaws.services.s3.model.BucketVersioningConfiguration;
import com.amazonaws.services.s3.model.CreateBucketRequest;
import com.amazonaws.services.s3.model.DeleteMarkerReplication;
import com.amazonaws.services.s3.model.DeleteMarkerReplicationStatus;
import com.amazonaws.services.s3.model.ReplicationDestinationConfig;
import com.amazonaws.services.s3.model.ReplicationRule;
import com.amazonaws.services.s3.model.ReplicationRuleStatus;
import com.amazonaws.services.s3.model.SetBucketVersioningConfigurationRequest;
import com.amazonaws.services.s3.model.StorageClass;
import com.amazonaws.services.s3.model.replication.ReplicationFilter;
import com.amazonaws.services.s3.model.replication.ReplicationFilterPredicate;
import com.amazonaws.services.s3.model.replication.ReplicationPrefixPredicate;

import java.io.IOException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class CrossRegionReplication {
```

```
        public static void main(String[] args) throws IOException {
                Regions clientRegion = Regions.DEFAULT_REGION;
                String accountId = "*** Account ID ***";
                String roleName = "*** Role name ***";
                String sourceBucketName = "*** Source bucket name ***";
                String destBucketName = "*** Destination bucket name ***";
                String prefix = "Tax/";

                String roleARN = String.format("arn:aws:iam::%s:%s", accountId,
roleName);
                String destinationBucketARN = "arn:aws:s3:::" + destBucketName;

                AmazonS3 s3Client = AmazonS3Client.builder()
                                .withCredentials(new ProfileCredentialsProvider())
                                .withRegion(clientRegion)
                                .build();

                createBucket(s3Client, clientRegion, sourceBucketName);
                createBucket(s3Client, clientRegion, destBucketName);
                assignRole(roleName, clientRegion, sourceBucketName,
destBucketName);

                try {

                        // Create the replication rule.
                        List<ReplicationFilterPredicate> andOperands = new
ArrayList<ReplicationFilterPredicate>();
                        andOperands.add(new ReplicationPrefixPredicate(prefix));

                        Map<String, ReplicationRule> replicationRules = new
HashMap<String, ReplicationRule>();
                        replicationRules.put("ReplicationRule1",
                                        new ReplicationRule()
                                                        .withPriority(0)

.withStatus(ReplicationRuleStatus.Enabled)

.withDeleteMarkerReplication(
                                                                        new
DeleteMarkerReplication().withStatus(

   DeleteMarkerReplicationStatus.DISABLED))
```

```
                                                    .withFilter(new
ReplicationFilter().withPredicate(
                                                            new
ReplicationPrefixPredicate(prefix)))
                                                    .withDestinationConfig(new
ReplicationDestinationConfig()

.withBucketARN(destinationBucketARN)

.withStorageClass(StorageClass.Standard)));

                        // Save the replication rule to the source bucket.
                        s3Client.setBucketReplicationConfiguration(sourceBucketName,
                                        new BucketReplicationConfiguration()
                                                    .withRoleARN(roleARN)

.withRules(replicationRules));

                        // Retrieve the replication configuration and verify that
the configuration
                        // matches the rule we just set.
                        BucketReplicationConfiguration replicationConfig = s3Client

.getBucketReplicationConfiguration(sourceBucketName);
                        ReplicationRule rule =
replicationConfig.getRule("ReplicationRule1");
                        System.out.println("Retrieved destination bucket ARN: "
                                        +
rule.getDestinationConfig().getBucketARN());
                        System.out.println("Retrieved priority: " +
rule.getPriority());
                        System.out.println("Retrieved source-bucket replication rule
status: " + rule.getStatus());
                } catch (AmazonServiceException e) {
                        // The call was transmitted successfully, but Amazon S3
couldn't process
                        // it, so it returned an error response.
                        e.printStackTrace();
                } catch (SdkClientException e) {
                        // Amazon S3 couldn't be contacted for a response, or the
client
                        // couldn't parse the response from Amazon S3.
                        e.printStackTrace();
                }
```

```
        }

        private static void createBucket(AmazonS3 s3Client, Regions region, String
 bucketName) {
                CreateBucketRequest request = new CreateBucketRequest(bucketName,
 region.getName());
                s3Client.createBucket(request);
                BucketVersioningConfiguration configuration = new
 BucketVersioningConfiguration()
                                .withStatus(BucketVersioningConfiguration.ENABLED);

                SetBucketVersioningConfigurationRequest enableVersioningRequest =
 new SetBucketVersioningConfigurationRequest(
                                bucketName, configuration);
                s3Client.setBucketVersioningConfiguration(enableVersioningRequest);

        }

        private static void assignRole(String roleName, Regions region, String
 sourceBucket, String destinationBucket) {
                AmazonIdentityManagement iamClient =
 AmazonIdentityManagementClientBuilder.standard()
                                .withRegion(region)
                                .withCredentials(new ProfileCredentialsProvider())
                                .build();
                StringBuilder trustPolicy = new StringBuilder();
                trustPolicy.append("{\\r\\n   ");
                trustPolicy.append("\\\"Version\\\":\\\"2012-10-17\\\",\\r\\n   ");
                trustPolicy.append("\\\"Statement\\\":[\\r\\n      {\\r\\n
 ");
                trustPolicy.append("\\\"Effect\\\":\\\"Allow\\\",\\r\\n        \\
\"Principal\\\":{\\r\\n            ");
                trustPolicy.append("\\\"Service\\\":\\\"s3.amazonaws.com\\\"\\r\\n
       },\\r\\n         ");
                trustPolicy.append("\\\"Action\\\":\\\"sts:AssumeRole\\\"\\r\\n
  }\\r\\n   ]\\r\\n}");

                CreateRoleRequest createRoleRequest = new CreateRoleRequest()
                                .withRoleName(roleName)

 .withAssumeRolePolicyDocument(trustPolicy.toString());

                iamClient.createRole(createRoleRequest);
```

```
                StringBuilder permissionPolicy = new StringBuilder();
                permissionPolicy.append(
                                "{\\r\\n   \\\"Version\\\":\\\"2012-10-17\\\",\\r\\n
   \\\"Statement\\\":[\\r\\n      {\\r\\n         ");
                permissionPolicy.append(
                                "\\\"Effect\\\":\\\"Allow\\\",\\r\\n         \\
\"Action\\\":[\\r\\n             ");
                permissionPolicy.append("\\\"s3:GetObjectVersionForReplication\\\",\
\r\\n            ");
                permissionPolicy.append(
                                "\\\"s3:GetObjectVersionAcl\\\"\\r\\n         ],\\r\
\n         \\\"Resource\\\":[\\r\\n            ");
                permissionPolicy.append("\\\"arn:aws:s3:::");
                permissionPolicy.append(sourceBucket);
                permissionPolicy.append("/*\\\"\\r\\n         ]\\r\\n      },\\r\\n
     {\\r\\n         ");
                permissionPolicy.append(
                                "\\\"Effect\\\":\\\"Allow\\\",\\r\\n         \\
\"Action\\\":[\\r\\n            ");
                permissionPolicy.append(
                                "\\\"s3:ListBucket\\\",\\r\\n            \\
\"s3:GetReplicationConfiguration\\\"\\r\\n         ");
                permissionPolicy.append("],\\r\\n         \\\"Resource\\\":[\\r\\n
          \\\"arn:aws:s3:::");
                permissionPolicy.append(sourceBucket);
                permissionPolicy.append("\\r\\n         ");
                permissionPolicy
                                .append("]\\r\\n      },\\r\\n      {\\r\\n
 \\\"Effect\\\":\\\"Allow\\\",\\r\\n         ");
                permissionPolicy.append(
                                "\\\"Action\\\":[\\r\\n            \\
\"s3:ReplicateObject\\\",\\r\\n            ");
                permissionPolicy
                                .append("\\\"s3:ReplicateDelete\\\",\\r\\n
  \\\"s3:ReplicateTags\\\",\\r\\n            ");
                permissionPolicy.append("\\\"s3:GetObjectVersionTagging\\\"\\r\\n\\r
\\n         ],\\r\\n         ");
                permissionPolicy.append("\\\"Resource\\\":\\\"arn:aws:s3:::");
                permissionPolicy.append(destinationBucket);
                permissionPolicy.append("/*\\\"\\r\\n      }\\r\\n   ]\\r\\n}");

                PutRolePolicyRequest putRolePolicyRequest = new
 PutRolePolicyRequest()
                                .withRoleName(roleName)
```

```
                                .withPolicyDocument(permissionPolicy.toString())
                                .withPolicyName("crrRolePolicy");

                iamClient.putRolePolicy(putRolePolicyRequest);

        }
}
```

C#

Contoh AWS SDK untuk .NET kode berikut menambahkan konfigurasi replikasi ke bucket dan kemudian mengambilnya. Untuk menggunakan kode ini, berikan nama untuk bucket Anda dan Amazon Resource Name (ARN) untuk peran IAM Anda. Untuk informasi tentang menyiapkan dan menjalankan contoh kode, lihat Memulai AWS SDK untuk .NET di Panduan AWS SDK untuk .NET Pengembang.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class CrossRegionReplicationTest
    {
        private const string sourceBucket = "*** source bucket ***";
        // Bucket ARN example - arn:aws:s3:::destinationbucket
        private const string destinationBucketArn = "*** destination bucket ARN
 ***";
        private const string roleArn = "*** IAM Role ARN ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint sourceBucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
        public static void Main()
        {
            s3Client = new AmazonS3Client(sourceBucketRegion);
            EnableReplicationAsync().Wait();
        }
        static async Task EnableReplicationAsync()
        {
```

```
            try
            {
                ReplicationConfiguration replConfig = new ReplicationConfiguration
                {
                    Role = roleArn,
                    Rules =
                        {
                            new ReplicationRule
                            {
                                Prefix = "Tax",
                                Status = ReplicationRuleStatus.Enabled,
                                Destination = new ReplicationDestination
                                {
                                    BucketArn = destinationBucketArn
                                }
                            }
                        }
                };

                PutBucketReplicationRequest putRequest = new
PutBucketReplicationRequest
                {
                    BucketName = sourceBucket,
                    Configuration = replConfig
                };

                PutBucketReplicationResponse putResponse = await
s3Client.PutBucketReplicationAsync(putRequest);

                // Verify configuration by retrieving it.
                await RetrieveReplicationConfigurationAsync(s3Client);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
        }
```

```
        private static async Task RetrieveReplicationConfigurationAsync(IAmazonS3
 client)
        {
            // Retrieve the configuration.
            GetBucketReplicationRequest getRequest = new GetBucketReplicationRequest
            {
                BucketName = sourceBucket
            };
            GetBucketReplicationResponse getResponse = await
 client.GetBucketReplicationAsync(getRequest);
            // Print.
            Console.WriteLine("Printing replication configuration information...");
            Console.WriteLine("Role ARN: {0}", getResponse.Configuration.Role);
            foreach (var rule in getResponse.Configuration.Rules)
            {
                Console.WriteLine("ID: {0}", rule.Id);
                Console.WriteLine("Prefix: {0}", rule.Prefix);
                Console.WriteLine("Status: {0}", rule.Status);
            }
        }
    }
}
```

Mengkonfigurasi replikasi untuk bucket di akun yang berbeda

Replikasi langsung adalah penyalinan objek otomatis dan asinkron di seluruh ember dalam hal yang sama atau berbeda. Wilayah AWS Replikasi langsung menyalin objek dan pembaruan objek yang baru dibuat dari bucket sumber ke bucket atau bucket tujuan. Untuk informasi selengkapnya, lihat Mereplikasi objek di dalam dan di seluruh Wilayah.

Saat mengonfigurasi replikasi, Anda menambahkan aturan replikasi ke bucket sumber. Aturan replikasi menentukan objek bucket sumber mana yang akan direplikasi dan bucket tujuan atau bucket tempat objek yang direplikasi disimpan. Anda dapat membuat aturan untuk mereplikasi semua objek dalam bucket atau subset objek dengan awalan nama kunci tertentu, satu atau beberapa tag objek, atau keduanya. Bucket tujuan bisa Akun AWS sama dengan bucket sumber, atau bisa juga di akun yang berbeda.

Jika Anda menentukan ID versi objek yang akan dihapus, Amazon S3 menghapus versi objek tersebut dalam bucket sumber. Tetapi ini tidak mereplikasi penghapusan di bucket tujuan. Dengan kata lain, itu tidak menghapus versi objek yang sama dari bucket tujuan. Ini melindungi data dari penghapusan berbahaya.

Ketika Anda menambahkan aturan replikasi ke bucket, aturan diaktifkan secara default, sehingga aturan mulai bekerja segera setelah Anda menyimpannya.

Menyiapkan replikasi langsung ketika bucket sumber dan tujuan dimiliki oleh berbeda Akun AWS mirip dengan menyiapkan replikasi ketika kedua bucket dimiliki oleh akun yang sama. Namun, ada beberapa perbedaan saat Anda mengonfigurasi replikasi dalam skenario lintas akun:

- Pemilik bucket tujuan harus memberikan izin kepada pemilik bucket sumber untuk mereplikasi objek dalam kebijakan bucket tujuan.
- Jika Anda mereplikasi objek yang dienkripsi dengan enkripsi sisi server dengan kunci AWS Key Management Service (AWS KMS) (SSE-KMS) dalam skenario lintas akun, pemilik kunci KMS harus memberikan izin pemilik bucket sumber untuk menggunakan kunci KMS. Untuk informasi selengkapnya, lihat Memberikan izin tambahan untuk skenario lintas akun.
- Secara default, objek yang direplikasi dimiliki oleh pemilik bucket sumber. Dalam skenario lintas akun, Anda mungkin ingin mengonfigurasi replikasi untuk mengubah kepemilikan objek yang direplikasi menjadi pemilik bucket tujuan. Untuk informasi selengkapnya, lihat Mengubah pemilik replika.

Untuk mengonfigurasi replikasi saat bucket sumber dan tujuan dimiliki oleh yang berbeda Akun AWS

1. Dalam contoh ini, Anda membuat bucket sumber dan tujuan dalam dua yang berbeda Akun AWS. Anda harus memiliki dua profil kredensi yang ditetapkan untuk. AWS CLI Contoh ini menggunakan `acctA` dan `acctB` untuk nama-nama profil tersebut. Untuk informasi tentang menyetel profil kredensi dan menggunakan profil bernama, lihat Pengaturan konfigurasi dan file kredensi di AWS Command Line Interface Panduan Pengguna.
2. Ikuti step-by-step instruksi Mengkonfigurasi replikasi untuk bucket di akun yang sama dengan perubahan berikut:

   - Untuk semua AWS CLI perintah yang terkait dengan aktivitas bucket sumber (seperti membuat bucket sumber, mengaktifkan pembuatan versi, dan membuat peran IAM), gunakan profil. `acctA` Gunakan `acctB` profil untuk membuat bucket tujuan.
   - Pastikan kebijakan izin untuk peran IAM menentukan bucket sumber dan tujuan yang Anda buat untuk contoh ini.
3. Di konsol, tambahkan kebijakan bucket berikut pada bucket tujuan agar pemilik bucket sumber dapat mereplikasi objek. Untuk petunjuk, lihat Menambahkan kebijakan bucket dengan menggunakan konsol Amazon S3. Pastikan untuk mengedit kebijakan dengan memberikan Akun AWS ID pemilik bucket sumber, nama peran IAM, dan nama bucket tujuan.

> **Note**
>
> Untuk menggunakan contoh berikut, ganti *user input placeholders* dengan informasi Anda sendiri. Ganti *amzn-s3-demo-destination-bucket* dengan nama bucket tujuan Anda. Ganti *source-bucket-account-ID*`:role/service-role/`*source-account-IAM-role* di IAM Amazon Resource Name (ARN) dengan peran IAM yang Anda gunakan untuk konfigurasi replikasi ini.
> Jika Anda membuat peran layanan IAM secara manual, setel jalur peran di ARN IAM sebagai`role/service-role/`, seperti yang ditunjukkan dalam contoh kebijakan berikut. Untuk informasi selengkapnya, lihat IAM ARNs di Panduan Pengguna IAM.

```
{
   "Version":"2012-10-17",
   "Id":"",
   "Statement":[
      {
         "Sid":"Set-permissions-for-objects",
         "Effect":"Allow",
         "Principal":{
            "AWS":"arn:aws:iam::source-bucket-account-ID:role/service-role/source-account-IAM-role"
         },
         "Action":["s3:ReplicateObject", "s3:ReplicateDelete"],
         "Resource":"arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
      },
      {
         "Sid":"Set permissions on bucket",
         "Effect":"Allow",
         "Principal":{
            "AWS":"arn:aws:iam::source-bucket-account-ID:role/service-role/source-account-IAM-role"
         },
         "Action":["s3:GetBucketVersioning", "s3:PutBucketVersioning"],
         "Resource":"arn:aws:s3:::amzn-s3-demo-destination-bucket"
      }
   ]
}
```

4. (Opsional) Jika Anda mereplikasi objek yang dienkripsi dengan SSE-KMS, pemilik kunci KMS harus memberikan izin pemilik bucket sumber untuk menggunakan kunci KMS. Untuk informasi selengkapnya, lihat Memberikan izin tambahan untuk skenario lintas akun.
5. (Opsional) Dalam replikasi, pemilik objek sumber memiliki replika secara default. Jika bucket sumber dan tujuan dimiliki oleh yang berbeda Akun AWS, Anda dapat menambahkan pengaturan konfigurasi opsional untuk mengubah kepemilikan replika ke bucket Akun AWS yang memiliki tujuan. Ini termasuk pemberian izin `ObjectOwnerOverrideToBucketOwner`. Untuk informasi selengkapnya, lihat Mengubah pemilik replika.

Mengubah pemilik replika

Dalam replikasi, pemilik objek sumber juga memiliki replika secara default. Namun, ketika bucket sumber dan tujuan dimiliki oleh yang berbeda Akun AWS, Anda mungkin ingin mengubah kepemilikan replika. Misalnya, Anda mungkin ingin mengubah kepemilikan untuk membatasi akses ke replika objek. Dalam konfigurasi replikasi Anda, Anda dapat menambahkan pengaturan konfigurasi opsional untuk mengubah kepemilikan replika ke bucket Akun AWS yang memiliki tujuan.

Untuk mengubah pemilik replika, Anda melakukan hal berikut:

- Tambahkan opsi penggantian pemilik ke konfigurasi replikasi untuk memberi tahu Amazon S3 agar mengubah kepemilikan replika.
- Berikan Amazon S3 `s3:ObjectOwnerOverrideToBucketOwner` izin untuk mengubah kepemilikan replika.
- Tambahkan `s3:ObjectOwnerOverrideToBucketOwner` izin dalam kebijakan bucket tujuan untuk mengizinkan perubahan kepemilikan replika. `s3:ObjectOwnerOverrideToBucketOwner`Izin tersebut memungkinkan pemilik bucket tujuan untuk menerima kepemilikan replika objek.

Untuk informasi selengkapnya, lihat the section called “Pertimbangan untuk opsi penggantian kepemilikan” dan Menambahkan opsi penggantian pemilik ke konfigurasi replikasi. Untuk contoh kerja dengan step-by-step instruksi, lihatCara mengganti pemilik replika.

> **Important**
>
> Alih-alih menggunakan opsi penggantian pemilik, Anda dapat menggunakan setelan yang diberlakukan pemilik bucket untuk Kepemilikan Objek. Saat Anda menggunakan replikasi dan bucket sumber dan tujuan dimiliki oleh berbeda Akun AWS, pemilik bucket bucket tujuan

dapat menggunakan pengaturan yang diberlakukan pemilik bucket untuk Kepemilikan Objek untuk mengubah kepemilikan replika ke bucket Akun AWS yang memiliki tujuan. Pengaturan ini menonaktifkan daftar kontrol akses objek (ACLs).
Setelan yang diberlakukan pemilik bucket meniru perilaku override pemilik yang ada tanpa memerlukan izin. `s3:ObjectOwnerOverrideToBucketOwner` Semua objek yang direplikasi ke bucket tujuan dengan pengaturan yang diberlakukan pemilik bucket dimiliki oleh pemilik bucket tujuan. Untuk informasi selengkapnya tentang Kepemilikan Objek, lihat Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.

Pertimbangan untuk opsi penggantian kepemilikan

Ketika Anda mengonfigurasi opsi penggantian kepemilikan, pertimbangan berikut berlaku:

- Secara default, pemilik objek sumber juga memiliki replika. Amazon S3 mereplikasi versi objek dan ACL yang terkait dengannya.

  Jika Anda menambahkan opsi penggantian pemilik ke konfigurasi replikasi Anda, Amazon S3 hanya mereplikasi versi objek, bukan ACL. Selain itu, Amazon S3 tidak mereplikasi perubahan berikutnya ke objek sumber ACL. Amazon S3 menetapkan ACL pada replika yang memberikan kontrol penuh kepada pemilik bucket tujuan.
- Saat Anda memperbarui konfigurasi replikasi untuk mengaktifkan atau menonaktifkan penggantian pemilik, perilaku berikut akan terjadi:
  - Jika Anda menambahkan opsi penggantian pemilik ke konfigurasi replikasi:

    Saat Amazon S3 mereplikasi versi objek, ia membuang ACL yang terkait dengan objek sumber. Sebagai gantinya, Amazon S3 menetapkan ACL pada replika, memberikan kontrol penuh kepada pemilik bucket tujuan. Amazon S3 tidak mereplikasi perubahan selanjutnya ke objek sumber ACL. Namun, perubahan ACL ini tidak berlaku untuk versi objek yang direplikasi sebelum Anda menetapkan opsi penggantian pemilik. Pembaruan ACL pada objek sumber yang direplikasi sebelum penggantian pemilik telah ditetapkan untuk terus direplikasi (karena objek dan replikanya terus memiliki pemilik yang sama).
  - Jika Anda menghapus opsi penggantian pemilik dari konfigurasi replikasi:

    Amazon S3 mereplikasi objek baru yang muncul di bucket sumber dan yang ACLs terkait dengan bucket tujuan. Untuk objek yang direplikasi sebelum Anda menghapus penggantian pemilik, Amazon S3 tidak mereplikasi ACLs karena perubahan kepemilikan objek yang dibuat Amazon

S3 tetap berlaku. Artinya, ACLs letakkan pada versi objek yang direplikasi ketika override pemilik disetel terus tidak direplikasi.

Menambahkan opsi penggantian pemilik ke konfigurasi replikasi

> **Warning**
>
> Tambahkan opsi penggantian pemilik hanya jika bucket sumber dan tujuan dimiliki oleh yang berbeda. Akun AWS Amazon S3 tidak memeriksa apakah bucket dimiliki oleh akun yang sama atau berbeda. Jika Anda menambahkan override pemilik saat kedua bucket dimiliki oleh yang sama Akun AWS, Amazon S3 menerapkan penggantian pemilik. Opsi ini memberikan izin penuh kepada pemilik bucket tujuan dan tidak mereplikasi pembaruan berikutnya ke daftar kontrol akses objek sumber (). ACLs Pemilik replika dapat langsung mengubah ACL yang terkait dengan replika dengan permintaan `PutObjectAcl`, tetapi tidak melalui replikasi.

Untuk menentukan opsi penggantian pemilik, tambahkan berikut ini ke setiap elemen `Destination`:

- Elemen `AccessControlTranslation`, yang memberi tahu Amazon S3 untuk mengubah kepemilikan replika
- `Account`Elemen, yang menentukan Akun AWS pemilik bucket tujuan

```
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
    ...
    <Destination>
      ...
      <AccessControlTranslation>
           <Owner>Destination</Owner>
       </AccessControlTranslation>
      <Account>destination-bucket-owner-account-id</Account>
    </Destination>
  </Rule>
</ReplicationConfiguration>
```

Contoh konfigurasi replikasi berikut memberitahu Amazon S3 untuk mereplikasi objek yang memiliki key *`Tax`* prefix ke bucket tujuan dan mengubah *`amzn-s3-demo-destination-bucket`*

kepemilikan replika. Untuk menggunakan contoh ini, ganti *user input placeholders*dengan informasi Anda sendiri.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
   <Role>arn:aws:iam::account-id:role/role-name</Role>
   <Rule>
      <ID>Rule-1</ID>
      <Priority>1</Priority>
      <Status>Enabled</Status>
      <DeleteMarkerReplication>
         <Status>Disabled</Status>
      </DeleteMarkerReplication>
      <Filter>
         <Prefix>Tax</Prefix>
      </Filter>
      <Destination>
         <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
         <Account>destination-bucket-owner-account-id</Account>
         <AccessControlTranslation>
            <Owner>Destination</Owner>
         </AccessControlTranslation>
      </Destination>
   </Rule>
</ReplicationConfiguration>
```

Memberi Amazon S3 izin untuk mengubah kepemilikan replika

Berikan izin Amazon S3 untuk mengubah kepemilikan replika dengan menambahkan izin untuk `s3:ObjectOwnerOverrideToBucketOwner` tindakan dalam kebijakan izin yang terkait dengan peran (IAM). AWS Identity and Access Management Peran ini adalah peran IAM yang Anda tentukan dalam konfigurasi replikasi yang memungkinkan Amazon S3 untuk mengasumsikan dan mereplikasi objek atas nama Anda. Untuk menggunakan contoh berikut, ganti *amzn-s3-demo-destination-bucket* dengan nama bucket tujuan.

```
...
{
    "Effect":"Allow",
         "Action":[
       "s3:ObjectOwnerOverrideToBucketOwner"
    ],
    "Resource":"arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
```

```
}
...
```

Menambahkan izin dalam kebijakan bucket tujuan untuk memungkinkan perubahan kepemilikan replika

Pemilik bucket tujuan harus memberikan izin bucket sumber untuk mengubah kepemilikan replika. Pemilik bucket tujuan memberi pemilik bucket sumber izin untuk tindakan `s3:ObjectOwnerOverrideToBucketOwner`. Izin ini memungkinkan pemilik bucket tujuan untuk menerima kepemilikan replika objek. Contoh pernyataan kebijakan bucket berikut menunjukkan cara melakukannya. Untuk menggunakan contoh ini, ganti *`user input placeholders`*dengan informasi Anda sendiri.

```
...
{
    "Sid":"1",
    "Effect":"Allow",
    "Principal":{"AWS":"source-bucket-account-id"},
    "Action":["s3:ObjectOwnerOverrideToBucketOwner"],
    "Resource":"arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
}
...
```

Cara mengganti pemilik replika

Jika bucket sumber dan tujuan dalam konfigurasi replikasi dimiliki oleh berbeda Akun AWS, Anda dapat memberi tahu Amazon S3 untuk mengubah kepemilikan replika ke bucket yang memiliki tujuan. Akun AWS Contoh berikut menunjukkan cara menggunakan konsol Amazon S3, AWS Command Line Interface (AWS CLI), dan AWS SDKs untuk mengubah kepemilikan replika.

Menggunakan konsol S3

Untuk step-by-step instruksi, lihatMengkonfigurasi replikasi untuk bucket di akun yang sama. Topik ini memberikan instruksi untuk menyiapkan konfigurasi replikasi ketika bucket sumber dan tujuan dimiliki oleh yang sama dan berbeda. Akun AWS

Menggunakan AWS CLI

Prosedur berikut menunjukkan cara mengubah kepemilikan replika dengan menggunakan. AWS CLI Dalam prosedur ini, Anda melakukan hal berikut:

- Buat ember sumber dan tujuan.
- Aktifkan pembuatan versi pada ember.
- Buat peran AWS Identity and Access Management (IAM) yang memberikan izin Amazon S3 untuk mereplikasi objek.
- Tambahkan konfigurasi replikasi ke bucket sumber.
- Dalam konfigurasi replikasi, Anda mengarahkan Amazon S3 untuk mengubah kepemilikan replika.
- Anda menguji konfigurasi replikasi Anda.

Untuk mengubah kepemilikan replika saat bucket sumber dan tujuan dimiliki oleh different Akun AWS ()AWS CLI

Untuk menggunakan AWS CLI perintah contoh dalam prosedur ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

1. Dalam contoh ini, Anda membuat bucket sumber dan tujuan dalam dua yang berbeda Akun AWS. Untuk bekerja dengan dua akun ini, konfigurasikan AWS CLI dengan dua profil bernama. Contoh ini menggunakan profil yang, masing-masing, diberi nama *`acctA`* dan *`acctB`*. Untuk informasi tentang menyetel profil kredensi dan menggunakan profil bernama, lihat Pengaturan konfigurasi dan file kredensi di AWS Command Line Interface Panduan Pengguna.

   > **Important**
   >
   > Profil yang Anda gunakan untuk prosedur ini harus memiliki izin yang diperlukan. Misalnya, dalam konfigurasi replikasi, Anda menentukan peran IAM yang dapat diasumsikan oleh Amazon S3. Anda dapat melakukan ini hanya jika profil yang Anda gunakan memiliki `iam:PassRole` izin. Jika Anda menggunakan kredensi pengguna administrator untuk membuat profil bernama, maka Anda dapat melakukan semua tugas dalam prosedur ini. Untuk informasi selengkapnya, lihat Memberikan izin pengguna untuk meneruskan peran ke Layanan AWS di Panduan Pengguna IAM.

2. Buat bucket sumber dan aktifkan Penentuan Versi. Contoh ini membuat bucket sumber bernama *`amzn-s3-demo-source-bucket`* di Wilayah AS Timur (Virginia N.) (`us-east-1`).

   ```
   aws s3api create-bucket \
   --bucket amzn-s3-demo-source-bucket \
   --region us-east-1 \
   --profile acctA
   ```

```
aws s3api put-bucket-versioning \
--bucket amzn-s3-demo-source-bucket \
--versioning-configuration Status=Enabled \
--profile acctA
```

3. Buat bucket tujuan dan aktifkan Penentuan Versi. Contoh ini membuat bucket tujuan bernama *amzn-s3-demo-destination-bucket* di Wilayah AS Barat (Oregon) (us-west-2). Gunakan Akun AWS profil yang berbeda dari yang Anda gunakan untuk bucket sumber.

```
aws s3api create-bucket \
--bucket amzn-s3-demo-destination-bucket \
--region us-west-2 \
--create-bucket-configuration LocationConstraint=us-west-2 \
--profile acctB
```

```
aws s3api put-bucket-versioning \
--bucket amzn-s3-demo-destination-bucket \
--versioning-configuration Status=Enabled \
--profile acctB
```

4. Anda harus menambahkan izin dalam kebijakan bucket tujuan untuk memungkinkan perubahan kepemilikan replika.

   a. Simpan kebijakan berikut ke file bernama*destination-bucket-policy*.json. Pastikan untuk mengganti *user input placeholders* dengan informasi Anda sendiri.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "destination_bucket_policy_sid",
      "Principal": {
        "AWS": "source-bucket-owner-account-id"
      },
      "Action": [
        "s3:ReplicateObject",
        "s3:ReplicateDelete",
        "s3:ObjectOwnerOverrideToBucketOwner",
        "s3:ReplicateTags",
        "s3:GetObjectVersionTagging"
      ],
```

```
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
      ]
    }
  ]
}
```

b. Tambahkan kebijakan sebelumnya ke bucket tujuan dengan menggunakan perintah berikut: `put-bucket-policy`

```
aws s3api put-bucket-policy --region $ {destination-region} --bucket $ {amzn-s3-demo-destination-bucket} --policy file://destination_bucket_policy.json
```

5. Buat peran IAM. Anda menentukan peran ini dalam konfigurasi replikasi yang Anda tambahkan ke bucket sumber nanti. Amazon S3 mengasumsikan peran ini untuk mereplikasi objek atas nama Anda. Anda membuat peran IAM dalam dua langkah:

   - Buat peran.
   - Lampirkan kebijakan izin pada peran tersebut.

   a. Buat peran IAM.

      i. Salin kebijakan kepercayaan berikut dan simpan di berkas dengan nama *s3-role-trust-policy*`.json` di direktori saat ini di komputer lokal Anda. Kebijakan ini memberi Amazon S3 izin untuk mengasumsikan peran tersebut.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"s3.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

ii. Jalankan AWS CLI `create-role` perintah berikut untuk membuat peran IAM:

```
$ aws iam create-role \
--role-name replicationRole \
--assume-role-policy-document file://s3-role-trust-policy.json  \
--profile acctA
```

Catat Nama Sumber Daya Amazon (ARN) dari peran IAM yang Anda buat. Anda akan membutuhkan ARN ini di langkah selanjutnya.

b. Lampirkan kebijakan izin pada peran tersebut.

i. Salin kebijakan kepercayaan berikut dan simpan di berkas dengan nama *s3-role-perm-pol-changeowner*`.json` di direktori saat ini di komputer lokal Anda. Kebijakan ini memberikan izin untuk berbagai bucket dan tindakan objek Amazon S3. Pada langkah-langkah berikut, Anda melampirkan kebijakan ini ke peran IAM yang Anda buat sebelumnya.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObjectVersionForReplication",
            "s3:GetObjectVersionAcl"
         ],
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:ListBucket",
            "s3:GetReplicationConfiguration"
         ],
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket"
         ]
      },
      {
```

```
            "Effect":"Allow",
            "Action":[
               "s3:ReplicateObject",
               "s3:ReplicateDelete",
               "s3:ObjectOwnerOverrideToBucketOwner",
               "s3:ReplicateTags",
               "s3:GetObjectVersionTagging"
            ],
            "Resource":"arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
         }
      ]
}
```

ii. Untuk melampirkan kebijakan izin sebelumnya ke peran, jalankan perintah berikut: `put-role-policy`

```
$ aws iam put-role-policy \
--role-name replicationRole \
--policy-document file://s3-role-perm-pol-changeowner.json \
--policy-name replicationRolechangeownerPolicy \
--profile acctA
```

6. Menambahkan konfigurasi replikasi ke bucket sumber.

a. AWS CLI Memerlukan menentukan konfigurasi replikasi sebagai JSON. Simpan JSON berikut dalam file yang disebut *`replication`*`.json` ke direktori lokal di komputer Anda. Dalam konfigurasi, `AccessControlTranslation` menentukan perubahan kepemilikan replika dari pemilik bucket sumber ke pemilik bucket tujuan.

```
{
   "Role":"IAM-role-ARN",
   "Rules":[
      {
         "Status":"Enabled",
         "Priority":1,
         "DeleteMarkerReplication":{
            "Status":"Disabled"
         },
         "Filter":{
         },
         "Status":"Enabled",
         "Destination":{
            "Bucket":"arn:aws:s3:::amzn-s3-demo-destination-bucket",
```

```
            "Account":"destination-bucket-owner-account-id",
            "AccessControlTranslation":{
               "Owner":"Destination"
            }
         }
      }
   ]
}
```

b. Edit JSON dengan memberikan nilai untuk nama bucket tujuan, ID akun pemilik bucket tujuan, dan file. *IAM-role-ARN* Ganti *IAM-role-ARN* dengan ARN peran IAM yang Anda buat sebelumnya. Simpan perubahan.

c. Untuk menambahkan konfigurasi replikasi ke bucket sumber, jalankan perintah berikut:

```
$ aws s3api put-bucket-replication \
--replication-configuration file://replication.json \
--bucket amzn-s3-demo-source-bucket \
--profile acctA
```

7. Uji konfigurasi replikasi Anda dengan memeriksa kepemilikan replika di konsol Amazon S3.

a. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

b. Tambahkan objek ke bucket sumber. Verifikasi bahwa bucket tujuan berisi replika objek dan kepemilikan replika telah berubah menjadi bucket Akun AWS yang memiliki tujuan.

Menggunakan AWS SDKs

Untuk contoh kode untuk menambahkan konfigurasi replikasi, lihatMenggunakan AWS SDKs. Anda harus memodifikasi konfigurasi replikasi dengan tepat. Untuk informasi konseptual, lihat Mengubah pemilik replika.

Memenuhi persyaratan kepatuhan dengan Kontrol Waktu Replikasi S3

Kontrol Waktu Replikasi S3 (S3 RTC) membantu Anda memenuhi persyaratan kepatuhan atau bisnis untuk replikasi data dan memberikan visibilitas ke dalam waktu replikasi Amazon S3. S3 RTC mereplikasi sebagian besar objek yang Anda unggah ke Amazon S3 dalam hitungan detik, dan 99,9 persen objek tersebut dalam waktu 15 menit.

Secara default, S3 RTC mencakup dua cara untuk melacak kemajuan replikasi:

- Metrik Replikasi S3 — Anda dapat menggunakan metrik Replikasi S3 untuk memantau jumlah total operasi S3 API yang menunggu replikasi, ukuran total objek yang menunggu replikasi, waktu replikasi maksimum ke Wilayah tujuan, dan jumlah total operasi yang gagal replikasi. Kemudian, Anda dapat memantau setiap set data yang akan direplikasi secara terpisah. Anda juga dapat mengaktifkan metrik Replikasi S3 secara independen dari S3 RTC. Untuk informasi selengkapnya, lihat the section called “Menggunakan metrik Replikasi S3”.

  Aturan replikasi dengan Kontrol Waktu Replikasi S3 (S3 RTC) memungkinkan publikasi metrik Replikasi S3. Metrik replikasi tersedia dalam 15 menit setelah S3 RTC diaktifkan. Metrik replikasi tersedia melalui konsol Amazon S3, Amazon S3 API, AWS SDKs the, the AWS Command Line Interface ()AWS CLI, dan Amazon. CloudWatch Untuk informasi selengkapnya tentang CloudWatch metrik, lihatMemantau metrik dengan Amazon CloudWatch. Untuk informasi selengkapnya tentang melihat metrik replikasi melalui konsol Amazon S3, lihat. Melihat metrik replikasi

  Metrik Replikasi S3 ditagih dengan tarif yang sama dengan metrik kustom Amazon. CloudWatch Untuk informasi, lihat harga Amazon CloudWatch.
- Pemberitahuan Acara Amazon S3 — S3 RTC menyediakan `OperationMissedThreshold` dan `OperationReplicatedAfterThreshold` peristiwa yang memberi tahu pemilik bucket jika replikasi objek melebihi atau terjadi setelah ambang batas 15 menit. Dengan S3 RTC, Pemberitahuan Acara Amazon S3 dapat memberi tahu Anda dalam contoh langka ketika objek tidak bereplikasi dalam waktu 15 menit dan ketika objek tersebut bereplikasi setelah ambang batas 15 menit.

  Metrik replikasi tersedia dalam 15 menit setelah S3 RTC diaktifkan. Pemberitahuan Acara Amazon S3 tersedia melalui Amazon SQS, Amazon SNS, atau. AWS Lambda Untuk informasi selengkapnya, lihat the section called “Menerima peristiwa kegagalan replikasi”.

Praktik terbaik dan pedoman untuk S3 RTC

Saat mereplikasi data di Amazon S3 dengan S3 Replication Time Control (S3 RTC) diaktifkan, ikuti panduan praktik terbaik ini untuk mengoptimalkan kinerja replikasi untuk beban kerja Anda.

Topik

- Replikasi Amazon S3 dan pedoman kinerja tingkat permintaan
- Memperkirakan tingkat permintaan replikasi Anda
- Melebihi kuota kecepatan transfer data S3 RTC

- AWS KMS tingkat permintaan replikasi objek terenkripsi

Replikasi Amazon S3 dan pedoman kinerja tingkat permintaan

Ketika mengunggah dan mengambil penyimpanan dari Amazon S3, aplikasi Anda dapat mencapai ribuan transaksi per detik dalam kinerja permintaan. Misalnya, aplikasi dapat mencapai setidaknya 3.500PUT/COPYPOST/DELETEatau GET HEAD 5.500/permintaan per detik per awalan dalam bucket S3, termasuk permintaan yang dibuat oleh S3 Replication atas nama Anda. Tidak ada batas jumlah awalan dalam bucket. Anda dapat meningkatkan kinerja baca atau tulis dengan cara menyelaraskan pembacaan. Misalnya, jika Anda membuat 10 awalan dalam bucket S3 untuk memparalelkan pembacaan, Anda dapat menskalakan kinerja baca menjadi 55.000 permintaan baca per detik.

Amazon S3 secara otomatis menskalakan sebagai respons terhadap tarif permintaan berkelanjutan di atas pedoman ini, atau tarif permintaan berkelanjutan bersamaan dengan permintaan. LIST Meski Amazon S3 secara internal mengoptimalkan tingkat permintaan baru, Anda mungkin menerima respons permintaan HTTP 503 untuk sementara hingga optimasi selesai. Perilaku ini mungkin terjadi dengan peningkatan tingkat permintaan per detik, atau saat Anda pertama kali mengaktifkan S3 RTC. Selama periode ini, latensi replikasi Anda dapat meningkat. Perjanjian tingkat layanan (SLA) S3 RTC tidak berlaku untuk periode waktu ketika pedoman kinerja Amazon S3 pada permintaan per detik terlampaui.

SLA S3 RTC juga tidak berlaku selama periode waktu dimana kecepatan transfer data replikasi Anda melebihi kuota default 1 gigabit per detik (Gbps). Jika Anda mengharapkan tingkat transfer replikasi Anda melebihi 1 Gbps, Anda dapat menghubungi AWS Dukungan Center atau menggunakan Service Quotas untuk meminta peningkatan kuota transfer rate replikasi Anda.

Memperkirakan tingkat permintaan replikasi Anda

Total tingkat permintaan Anda termasuk permintaan yang dibuat oleh replikasi Amazon S3 atas nama Anda harus berada dalam pedoman tarif permintaan Amazon S3 untuk bucket sumber replikasi dan tujuan. Untuk setiap objek yang direplikasi, replikasi Amazon S3 menghasilkan hingga GET lima permintaanHEAD/dan PUT satu permintaan ke bucket sumber, dan PUT satu permintaan ke setiap bucket tujuan.

Misalnya, jika Anda berharap untuk mereplikasi 100 objek per detik, replikasi Amazon S3 dapat melakukan PUT 100 permintaan tambahan atas nama Anda dengan total PUT 200 permintaan per detik ke bucket S3 sumber. Replikasi Amazon S3 juga dapat melakukan hingga GET 500/ HEAD permintaan (GETHEAD5/permintaan untuk setiap objek yang direplikasi.)

> **Note**
>
> Anda dikenakan biaya hanya untuk satu PUT permintaan per objek yang direplikasi. Untuk informasi selengkapnya, lihat informasi harga di Amazon S3 FAQs tentang replikasi.

Melebihi kuota kecepatan transfer data S3 RTC

Jika Anda mengharapkan kecepatan transfer data S3 RTC melebihi kuota default 1 Gbps, hubungi AWS Dukungan Center atau gunakan Service Quotas untuk meminta peningkatan kuota transfer rate replikasi Anda.

AWS KMS tingkat permintaan replikasi objek terenkripsi

Saat Anda mereplikasi objek yang dienkripsi dengan enkripsi sisi server dengan kunci AWS Key Management Service (AWS KMS) (SSE-KMS), kuota permintaan per detik berlaku. AWS KMS AWS KMS mungkin menolak permintaan yang valid karena tingkat permintaan Anda melebihi kuota untuk jumlah permintaan per detik. Ketika permintaan dibatasi, AWS KMS mengembalikan kesalahan. `ThrottlingException` Kuota tarif AWS KMS permintaan berlaku untuk permintaan yang Anda buat secara langsung dan permintaan yang dibuat oleh replikasi Amazon S3 atas nama Anda.

Misalnya, jika Anda berharap untuk mereplikasi 1.000 objek per detik, Anda dapat mengurangi 2.000 permintaan dari kuota tingkat AWS KMS permintaan Anda. Tingkat permintaan per detik yang dihasilkan tersedia untuk AWS KMS beban kerja Anda tidak termasuk replikasi. Anda dapat menggunakan metrik AWS KMS permintaan di Amazon CloudWatch untuk memantau total tingkat AWS KMS permintaan pada Anda Akun AWS.

Untuk meminta peningkatan kuota AWS KMS permintaan per detik, hubungi AWS Dukungan Pusat atau gunakan Service Quotas.

Mengaktifkan Kontrol Waktu Replikasi S3

Anda dapat mulai menggunakan Kontrol Waktu Replikasi S3 (S3 RTC) dengan aturan replikasi baru atau yang sudah ada. Anda dapat memilih untuk menerapkan aturan replikasi ke seluruh bucket, atau ke objek dengan awalan atau tag tertentu. Saat Anda mengaktifkan S3 RTC, metrik Replikasi S3 juga diaktifkan pada aturan replikasi Anda.

Anda dapat mengonfigurasi S3 RTC dengan menggunakan konsol Amazon S3, API Amazon S3, AWS SDKs the, dan (). AWS Command Line Interface AWS CLI

Menggunakan konsol S3

Untuk step-by-step instruksi, lihatMengkonfigurasi replikasi untuk bucket di akun yang sama. Topik ini memberikan instruksi untuk mengaktifkan S3 RTC dalam konfigurasi replikasi Anda ketika bucket sumber dan tujuan dimiliki oleh yang sama dan berbeda. Akun AWS

Menggunakan AWS CLI

Untuk menggunakan AWS CLI untuk mereplikasi objek dengan S3 RTC diaktifkan, Anda membuat bucket, mengaktifkan pembuatan versi pada bucket, membuat peran IAM yang memberikan izin Amazon S3 untuk mereplikasi objek, dan menambahkan konfigurasi replikasi ke bucket sumber. Konfigurasi replikasi harus mengaktifkan S3 RTC, seperti yang ditunjukkan pada contoh berikut.

Untuk step-by-step petunjuk untuk mengatur konfigurasi replikasi Anda dengan menggunakan AWS CLI, lihatMengkonfigurasi replikasi untuk bucket di akun yang sama.

Contoh konfigurasi replikasi berikut memungkinkan dan menetapkan `ReplicationTime` dan `EventThreshold` nilai-nilai untuk aturan replikasi. Mengaktifkan dan mengatur nilai-nilai ini memungkinkan S3 RTC pada aturan.

```
{
    "Rules": [
        {
            "Status": "Enabled",
            "Filter": {
                "Prefix": "Tax"
            },
            "DeleteMarkerReplication": {
                "Status": "Disabled"
            },
            "Destination": {
                "Bucket": "arn:aws:s3:::amzn-s3-demo-destination-bucket",
                "Metrics": {
                    "Status": "Enabled",
                    "EventThreshold": {
                        "Minutes": 15
                    }
                },
                "ReplicationTime": {
                    "Status": "Enabled",
                    "Time": {
                        "Minutes": 15
                    }
```

```
                }
            },
            "Priority": 1
        }
    ],
    "Role": "IAM-Role-ARN"
}
```

> **Important**
>
> `Metrics:EventThreshold:Minutes`dan hanya `ReplicationTime:Time:Minutes` dapat memiliki 15 nilai yang valid.

Menggunakan AWS SDK for Java

Contoh Java berikut menambahkan konfigurasi replikasi dengan S3 Replication Time Control (S3 RTC) diaktifkan.

```
import software.amazon.awssdk.auth.credentials.AwsBasicCredentials;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.model.DeleteMarkerReplication;
import software.amazon.awssdk.services.s3.model.Destination;
import software.amazon.awssdk.services.s3.model.Metrics;
import software.amazon.awssdk.services.s3.model.MetricsStatus;
import software.amazon.awssdk.services.s3.model.PutBucketReplicationRequest;
import software.amazon.awssdk.services.s3.model.ReplicationConfiguration;
import software.amazon.awssdk.services.s3.model.ReplicationRule;
import software.amazon.awssdk.services.s3.model.ReplicationRuleFilter;
import software.amazon.awssdk.services.s3.model.ReplicationTime;
import software.amazon.awssdk.services.s3.model.ReplicationTimeStatus;
import software.amazon.awssdk.services.s3.model.ReplicationTimeValue;

public class Main {

  public static void main(String[] args) {
    S3Client s3 = S3Client.builder()
      .region(Region.US_EAST_1)
      .credentialsProvider(() -> AwsBasicCredentials.create(
          "AWS_ACCESS_KEY_ID",
          "AWS_SECRET_ACCESS_KEY")
      )
```

```
      .build();

    ReplicationConfiguration replicationConfig = ReplicationConfiguration
      .builder()
      .rules(
          ReplicationRule
            .builder()
            .status("Enabled")
            .priority(1)
            .deleteMarkerReplication(
                DeleteMarkerReplication
                    .builder()
                    .status("Disabled")
                    .build()
            )
            .destination(
                Destination
                    .builder()
                    .bucket("destination_bucket_arn")
                    .replicationTime(
                        ReplicationTime.builder().time(
                            ReplicationTimeValue.builder().minutes(15).build()
                        ).status(
                            ReplicationTimeStatus.ENABLED
                        ).build()
                    )
                    .metrics(
                        Metrics.builder().eventThreshold(
                            ReplicationTimeValue.builder().minutes(15).build()
                        ).status(
                            MetricsStatus.ENABLED
                        ).build()
                    )
                    .build()
            )
            .filter(
                ReplicationRuleFilter
                    .builder()
                    .prefix("testtest")
                    .build()
            )
        .build())
        .role("role_arn")
        .build();
```

```
    // Put replication configuration
    PutBucketReplicationRequest putBucketReplicationRequest =
 PutBucketReplicationRequest
      .builder()
      .bucket("source_bucket")
      .replicationConfiguration(replicationConfig)
      .build();

    s3.putBucketReplication(putBucketReplicationRequest);
  }
}
```

Mereplikasi objek terenkripsi (SSE-S3, SSE-KMS, DSSE-KMS, SSE-C)

> **Important**
>
> Amazon S3 sudah menerapkan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3) sebagai tingkat dasar enkripsi bagi setiap bucket di Amazon S3. Mulai 5 Januari 2023, semua unggahan objek baru ke Amazon S3 secara otomatis akan dienkripsi tanpa biaya tambahan dan tidak akan berdampak pada kinerja. Status enkripsi otomatis untuk konfigurasi enkripsi default bucket S3 dan untuk unggahan objek baru tersedia di AWS CloudTrail log, S3 Inventory, S3 Storage Lens, konsol Amazon S3, dan sebagai header respons API Amazon S3 tambahan di dan. AWS Command Line Interface AWS SDKs Untuk informasi selengkapnya, lihat FAQ enkripsi default.

Ada beberapa pertimbangan khusus ketika Anda mereplikasi objek yang telah dienkripsi dengan menggunakan enkripsi di sisi server. Amazon S3 mendukung tipe enkripsi di sisi server berikut:

- Enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3)
- Enkripsi sisi server dengan kunci AWS Key Management Service (AWS KMS) (SSE-KMS)
- Enkripsi sisi server dua lapis dengan kunci (DSSE-KMS) AWS KMS
- Enkripsi di sisi server dengan kunci yang disediakan pelanggan (SSE-C)

Untuk informasi lebih lanjut tentang enkripsi di sisi server, lihat the section called “enkripsi di sisi server”.

Topik ini menjelaskan izin yang Anda perlukan untuk mengarahkan Amazon S3 guna mereplikasi objek yang telah dienkripsi menggunakan enkripsi di sisi server. Topik ini juga menyediakan elemen konfigurasi tambahan yang dapat Anda tambahkan dan contoh kebijakan AWS Identity and Access Management (IAM) yang memberikan izin yang diperlukan untuk mereplikasi objek terenkripsi.

Untuk contoh dengan step-by-step instruksi, lihatMengaktifkan replikasi untuk objek terenkripsi. Untuk informasi tentang pembuatan konfigurasi replikasi, lihat Mereplikasi objek di dalam dan di seluruh Wilayah.

> **Note**
>
> Anda dapat menggunakan Multi-wilayah AWS KMS keys di Amazon S3. Namun, Amazon S3 saat ini memperlakukan kunci multi-Wilayah selayaknya kunci satu Wilayah, dan tidak menggunakan fitur multi-Wilayah dari kunci tersebut. Untuk informasi selengkapnya, lihat Menggunakan kunci Multi-wilayah di Panduan AWS Key Management Service Pengembang.

Topik

- Bagaimana enkripsi bucket default memengaruhi replikasi
- Mereplikasi objek yang dienkripsi dengan SSE-C
- Mereplikasi objek yang dienkripsi dengan SSE-S3, SSE-KMS, atau DSSE-KMS
- Mengaktifkan replikasi untuk objek terenkripsi

Bagaimana enkripsi bucket default memengaruhi replikasi

Saat Anda mengaktifkan enkripsi default untuk bucket tujuan replikasi, perilaku enkripsi berikut berlaku:

- Jika objek dalam bucket sumber tidak dienkripsi, objek replika dalam bucket tujuan akan dienkripsi menggunakan pengaturan enkripsi default dari bucket tujuan. Akibatnya, tag entitas (ETags) dari objek sumber berbeda ETags dari objek replika. Jika Anda memiliki aplikasi yang menggunakan ETags, Anda harus memperbarui aplikasi tersebut untuk memperhitungkan perbedaan ini.
- Jika objek dalam bucket sumber dienkripsi dengan menggunakan enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3), enkripsi sisi server dengan kunci () (SSE-KMS), atau enkripsi sisi server dua lapis dengan AWS Key Management Service kunci (DSSE-KMS), objek replika di bucket tujuan menggunakan jenis enkripsi yang sama dengan AWS KMS objek sumber.AWS KMS Pengaturan enkripsi default bucket tujuan tidak digunakan.

Mereplikasi objek yang dienkripsi dengan SSE-C

Dengan menggunakan enkripsi sisi server dengan kunci yang disediakan pelanggan (SSE-C), Anda dapat mengelola kunci enkripsi milik Anda sendiri. Dengan SSE-C, Anda mengelola kunci sedangkan Amazon S3 mengelola proses enkripsi dan dekripsi. Anda harus memberikan kunci enkripsi sebagai bagian dari permintaan Anda, tetapi Anda tidak perlu menulis kode apa pun untuk melakukan enkripsi atau dekripsi objek. Saat mengunggah sebuah objek, Amazon S3 mengenkripsi objek menggunakan kunci yang Anda berikan. Amazon S3 kemudian membersihkan kunci itu dari memori. Saat mengambil sebuah objek, Anda harus memberikan kunci enkripsi yang sama sebagai bagian dari permintaan Anda. Untuk informasi selengkapnya, lihat the section called “Kunci enkripsi yang disediakan pelanggan (SSE-C)”.

S3 Replikasi mendukung objek yang dienkripsi dengan SSE-C. Anda dapat mengonfigurasi replikasi objek SSE-C di konsol Amazon S3 atau dengan cara yang sama seperti Anda mengonfigurasi replikasi untuk objek yang tidak terenkripsi. AWS SDKs Tidak ada izin SSE-C tambahan di luar apa yang saat ini diperlukan untuk replikasi.

Replikasi S3 secara otomatis mereplikasi objek terenkripsi SSE-C yang baru diunggah jika memenuhi syarat, seperti yang ditentukan dalam konfigurasi Replikasi S3 Anda. Untuk mereplikasi objek yang ada di bucket Anda, gunakan Replikasi Batch S3. Untuk informasi selengkapnya tentang mereplikasi objek, lihat the section called “Menyiapkan replikasi langsung” dan the section called “Mereplikasi objek yang ada”.

Tidak ada biaya tambahan untuk mereplikasi objek SSE-C. Untuk detail tentang harga replikasi, lihat harga Amazon S3.

Mereplikasi objek yang dienkripsi dengan SSE-S3, SSE-KMS, atau DSSE-KMS

Secara default, Amazon S3 tidak mereplikasi objek yang dienkripsi dengan SSE-KMS atau DSSE-KMS. Bagian ini menjelaskan elemen konfigurasi tambahan yang dapat Anda tambahkan ke Amazon S3 langsung untuk mereplikasi objek tersebut.

Untuk contoh dengan step-by-step instruksi, lihatMengaktifkan replikasi untuk objek terenkripsi. Untuk informasi tentang pembuatan konfigurasi replikasi, lihat Mereplikasi objek di dalam dan di seluruh Wilayah.

Menentukan informasi tambahan dalam konfigurasi replikasi

Dalam konfigurasi replikasi, Anda melakukan hal berikut:

- Dalam `Destination` elemen dalam konfigurasi replikasi Anda, tambahkan ID kunci terkelola AWS KMS pelanggan simetris yang Anda ingin Amazon S3 gunakan untuk mengenkripsi replika objek, seperti yang ditunjukkan dalam contoh konfigurasi replikasi berikut.
- Pilih ikut secara eksplisit dengan mengaktifkan replikasi objek yang dienkripsi menggunakan kunci KMS (SSE-KMS atau DSSE-KMS). Untuk memilih ikut, tambahkan elemen `SourceSelectionCriteria`, seperti yang ditunjukkan dalam konfigurasi replikasi contoh berikut.

```
<ReplicationConfiguration>
   <Rule>
      ...
      <SourceSelectionCriteria>
         <SseKmsEncryptedObjects>
           <Status>Enabled</Status>
         </SseKmsEncryptedObjects>
      </SourceSelectionCriteria>

      <Destination>
          ...
          <EncryptionConfiguration>
             <ReplicaKmsKeyID>AWS KMS key ARN or Key Alias ARN that's in the same
 Wilayah AWS as the destination bucket.</ReplicaKmsKeyID>
          </EncryptionConfiguration>
       </Destination>
      ...
   </Rule>
</ReplicationConfiguration>
```

Important

- Kunci KMS harus dibuat Wilayah AWS sama dengan bucket tujuan.
- Kunci KMS harus valid. Operasi API `PutBucketReplication` tidak memeriksa validitas kunci KMS. Jika menggunakan kunci KMS yang tidak valid, Anda akan menerima kode status `200 OK` HTTP sebagai respons, tetapi replikasi gagal.

Contoh berikut menunjukkan konfigurasi replikasi yang meliputi elemen konfigurasi opsional. Konfigurasi replikasi ini memiliki satu aturan. Aturan ini berlaku untuk objek dengan awalan kunci

Tax. Amazon S3 menggunakan AWS KMS key ID yang ditentukan untuk mengenkripsi replika objek ini.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration>
   <Role>arn:aws:iam::account-id:role/role-name</Role>
   <Rule>
      <ID>Rule-1</ID>
      <Priority>1</Priority>
      <Status>Enabled</Status>
      <DeleteMarkerReplication>
         <Status>Disabled</Status>
      </DeleteMarkerReplication>
      <Filter>
         <Prefix>Tax</Prefix>
      </Filter>
      <Destination>
         <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
         <EncryptionConfiguration>
            <ReplicaKmsKeyID>AWS KMS key ARN or Key Alias ARN that's in the same
 Wilayah AWS as the destination bucket.</ReplicaKmsKeyID>
         </EncryptionConfiguration>
      </Destination>
      <SourceSelectionCriteria>
         <SseKmsEncryptedObjects>
            <Status>Enabled</Status>
         </SseKmsEncryptedObjects>
      </SourceSelectionCriteria>
   </Rule>
</ReplicationConfiguration>
```

Memberikan izin tambahan untuk peran IAM

Untuk mereplikasi objek yang dienkripsi saat istirahat menggunakan SSE-S3, SSE-KMS, atau DSSE-KMS, berikan izin tambahan berikut ke peran (IAM) yang Anda tentukan dalam konfigurasi replikasi. AWS Identity and Access Management Anda memberikan izin ini dengan memperbarui kebijakan izin yang terkait dengan peran IAM.

- Tindakan **`s3:GetObjectVersionForReplication`** untuk objek sumber–Tindakan ini memungkinkan Amazon S3 mereplikasi objek tidak terenkripsi dan objek yang dibuat dengan enkripsi di sisi server menggunakan SSE-S3, SSE-KMS, atau DSSE-KMS.

> **Note**
>
> Kami menyarankan Anda menggunakan tindakan `s3:GetObjectVersionForReplication` alih-alih tindakan `s3:GetObjectVersion` karena `s3:GetObjectVersionForReplication` hanya memberi Amazon S3 izin minimum yang diperlukan untuk replikasi. Selain itu, tindakan `s3:GetObjectVersion` memungkinkan replikasi objek tidak terenkripsi dan terenkripsi SSE-S3, tetapi bukan objek yang dienkripsi menggunakan kunci KMS (SSE-KMS atau DSSE-KMS).

- **`kms:Decrypt`**dan **`kms:Encrypt`** AWS KMS tindakan untuk kunci KMS
  - Anda harus memberikan izin `kms:Decrypt` untuk AWS KMS key yang digunakan untuk mendekripsi objek sumber.
  - Anda harus memberikan izin `kms:Encrypt` untuk AWS KMS key yang digunakan untuk mengenkripsi replika objek.
- Tindakan **`kms:GenerateDataKey`** untuk mereplikasi objek plaintext–Jika Anda mereplikasi objek plaintext ke bucket dengan enkripsi SSE-KMS atau DSSE-KMS yang diaktifkan secara default, Anda harus menyertakan izin `kms:GenerateDataKey` untuk konteks enkripsi tujuan dan kunci KMS dalam kebijakan IAM.

Kami menyarankan Anda membatasi izin ini hanya untuk bucket tujuan dan objek dengan menggunakan AWS KMS tombol kondisi. Akun AWS Yang memiliki peran IAM harus memiliki izin untuk `kms:Encrypt` dan `kms:Decrypt` tindakan untuk kunci KMS yang tercantum dalam kebijakan. Jika kunci KMS dimiliki oleh orang lain Akun AWS, pemilik kunci KMS harus memberikan izin ini kepada Akun AWS yang memiliki peran IAM. Untuk informasi selengkapnya tentang mengelola akses ke kunci KMS ini, lihat Menggunakan kebijakan IAM dengan Panduan AWS KMSAWS Key Management Service Pengembang.

Replikasi dan Kunci Bucket S3

Untuk menggunakan replikasi dengan S3 Bucket Key, AWS KMS key kebijakan untuk kunci KMS yang digunakan untuk mengenkripsi replika objek harus menyertakan `kms:Decrypt` izin untuk prinsipal panggilan. Panggilan ke `kms:Decrypt` memverifikasi integritas Kunci Bucket S3 sebelum menggunakannya. Untuk informasi selengkapnya, lihat Menggunakan kunci Bucket S3 dengan replikasi.

Saat Kunci Bucket S3 diaktifkan untuk bucket sumber atau tujuan, konteks enkripsi akan menjadi Amazon Resource Name (ARN) bucket, bukan ARN objek (misalnya, `arn:aws:s3:::bucket_ARN`). Anda harus memperbarui kebijakan IAM Anda untuk menggunakan ARN bucket untuk konteks enkripsi:

```
"kms:EncryptionContext:aws:s3:arn": [
"arn:aws:s3:::bucket_ARN"
]
```

Untuk informasi selengkapnya, lihat Konteks enkripsi (`x-amz-server-side-encryption-context`) (di bagian "Menggunakan API REST") dan Perubahan yang perlu diperhatikan sebelum mengaktifkan Kunci Bucket S3.

Contoh kebijakan: Menggunakan SSE-S3 dan SSE-KMS dengan replikasi

Contoh kebijakan IAM berikut menunjukkan pernyataan untuk menggunakan SSE-S3 dan SSE-KMS dengan replikasi.

Example – Menggunakan SSE-KMS dengan bucket tujuan terpisah

Contoh kebijakan berikut menunjukkan pernyataan untuk menggunakan SSE-KMS dengan bucket tujuan terpisah.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Action": ["kms:Decrypt"],
         "Effect": "Allow",
         "Condition": {
             "StringLike": {
                  "kms:ViaService": "s3.source-bucket-region.amazonaws.com",
                  "kms:EncryptionContext:aws:s3:arn": [
                      "arn:aws:s3:::amzn-s3-demo-source-bucket/key-prefix1*"
                  ]
             }
          },
          "Resource": [
             "List of AWS KMS key ARNs that are used to encrypt source objects."
          ]
      },
      {
```

```
         "Action": ["kms:Encrypt"],
         "Effect": "Allow",
         "Condition": {
            "StringLike": {
               "kms:ViaService": "s3.destination-bucket-1-region.amazonaws.com",
               "kms:EncryptionContext:aws:s3:arn": [
                  "arn:aws:s3:::amzn-s3-demo-destination-bucket1/key-prefix1*"
               ]
         }
         },
         "Resource": [
            "AWS KMS key ARNs (in the same Wilayah AWS as destination bucket 1). Used
 to encrypt object replicas created in destination bucket 1."
         ]
      },
      {
         "Action": ["kms:Encrypt"],
         "Effect": "Allow",
         "Condition": {
            "StringLike": {
               "kms:ViaService": "s3.destination-bucket-2-region.amazonaws.com",
               "kms:EncryptionContext:aws:s3:arn": [
                  "arn:aws:s3:::amzn-s3-demo-destination-bucket2/key-prefix1*"
               ]
         }
         },
         "Resource": [
            "AWS KMS key ARNs (in the same Wilayah AWS as destination bucket 2). Used
 to encrypt object replicas created in destination bucket 2."
         ]
      }
   ]
}
```

Example – Mereplikasi objek yang dibuat dengan SSE-S3 dan SSE-KMS

Berikut ini adalah kebijakan IAM lengkap yang memberikan izin yang diperlukan untuk mereplikasi objek tidak terenkripsi, objek yang dibuat dengan SSE-S3, dan objek yang dibuat dengan SSE-KMS.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
```

```
        "Effect":"Allow",
        "Action":[
           "s3:GetReplicationConfiguration",
           "s3:ListBucket"
        ],
        "Resource":[
           "arn:aws:s3:::amzn-s3-demo-source-bucket"
        ]
     },
     {
        "Effect":"Allow",
        "Action":[
           "s3:GetObjectVersionForReplication",
           "s3:GetObjectVersionAcl"
        ],
        "Resource":[
           "arn:aws:s3:::amzn-s3-demo-source-bucket/key-prefix1*"
        ]
     },
     {
        "Effect":"Allow",
        "Action":[
           "s3:ReplicateObject",
           "s3:ReplicateDelete"
        ],
        "Resource":"arn:aws:s3:::amzn-s3-demo-destination-bucket/key-prefix1*"
     },
     {
        "Action":[
           "kms:Decrypt"
        ],
        "Effect":"Allow",
        "Condition":{
           "StringLike":{
              "kms:ViaService":"s3.source-bucket-region.amazonaws.com",
              "kms:EncryptionContext:aws:s3:arn":[
                 "arn:aws:s3:::amzn-s3-demo-source-bucket/key-prefix1*"
              ]
           }
        },
        "Resource":[
          "List of the AWS KMS key ARNs that are used to encrypt source objects."
        ]
     },
```

```
      {
         "Action":[
            "kms:Encrypt"
         ],
         "Effect":"Allow",
         "Condition":{
            "StringLike":{
               "kms:ViaService":"s3.destination-bucket-region.amazonaws.com",
               "kms:EncryptionContext:aws:s3:arn":[
                  "arn:aws:s3:::amzn-s3-demo-destination-bucket/prefix1*"
               ]
            }
         },
         "Resource":[
            "AWS KMS key ARNs (in the same Wilayah AWS as the destination bucket) to
 use for encrypting object replicas"
         ]
      }
   ]
}
```

Example – Mereplikasi objek dengan Kunci Bucket S3

Berikut ini adalah kebijakan IAM lengkap yang memberikan izin yang diperlukan untuk mereplikasi objek dengan Kunci Bucket S3.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetReplicationConfiguration",
            "s3:ListBucket"
         ],
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObjectVersionForReplication",
```

```
            "s3:GetObjectVersionAcl"
         ],
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket/key-prefix1*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:ReplicateObject",
            "s3:ReplicateDelete"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-destination-bucket/key-prefix1*"
      },
      {
         "Action":[
            "kms:Decrypt"
         ],
         "Effect":"Allow",
         "Condition":{
            "StringLike":{
               "kms:ViaService":"s3.source-bucket-region.amazonaws.com",
               "kms:EncryptionContext:aws:s3:arn":[
                  "arn:aws:s3:::amzn-s3-demo-source-bucket"
               ]
            }
         },
         "Resource":[
           "List of the AWS KMS key ARNs that are used to encrypt source objects."
         ]
      },
      {
         "Action":[
            "kms:Encrypt"
         ],
         "Effect":"Allow",
         "Condition":{
            "StringLike":{
               "kms:ViaService":"s3.destination-bucket-region.amazonaws.com",
               "kms:EncryptionContext:aws:s3:arn":[
                  "arn:aws:s3:::amzn-s3-demo-destination-bucket"
               ]
            }
         },
```

```
            "Resource":[
               "AWS KMS key ARNs (in the same Wilayah AWS as the destination bucket) to
 use for encrypting object replicas"
            ]
         }
   ]
}
```

Memberikan izin tambahan untuk skenario lintas akun

Dalam skenario lintas akun, di mana bucket sumber dan tujuan dimiliki oleh yang berbeda Akun AWS, Anda dapat menggunakan kunci KMS untuk mengenkripsi replika objek. Namun, pemilik kunci KMS harus memberikan izin pemilik bucket sumber untuk menggunakan kunci KMS.

> **Note**
>
> Jika Anda perlu mereplikasi data lintas akun SSE-KMS, maka aturan replikasi Anda harus menentukan kunci yang dikelola pelanggan untuk akun tujuan. AWS KMS Kunci yang dikelola AWSjangan izinkan penggunaan lintas akun, dan karenanya tidak dapat digunakan untuk melakukan replikasi lintas akun.

Untuk memberikan izin kepada pemilik bucket sumber untuk menggunakan kunci KMS (konsol AWS KMS)

1. Masuk ke AWS Management Console dan buka AWS KMS konsol di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Untuk melihat tombol di akun yang Anda buat dan kelola, di panel navigasi pilih CMK.
4. Pilih tombol KMS.
5. Di bawah bagian Konfigurasi umum, pilih tab Kebijakan kunci.
6. Gulir ke bawah ke Lainnya Akun AWS.
7. Pilih Tambahkan lainnya Akun AWS.

   Kotak Akun AWS dialog Lainnya muncul.
8. Di kotak dialog, pilih Tambahkan yang lain Akun AWS. Untuk arn:aws:iam::, masukkan ID akun bucket sumber.

9. Pilih Simpan perubahan.

Untuk memberikan izin kepada pemilik bucket sumber untuk menggunakan kunci KMS (AWS CLI)

- Untuk informasi tentang perintah `put-key-policy` AWS Command Line Interface (AWS CLI), lihat put-key-policydi AWS CLI Command Reference. Untuk informasi tentang operasi API `PutKeyPolicy` yang mendasarinya, lihat PutKeyPolicy di Referensi API AWS Key Management Service.

AWS KMS pertimbangan kuota transaksi

Saat Anda menambahkan banyak objek baru dengan AWS KMS enkripsi setelah mengaktifkan Replikasi Lintas Wilayah (CRR), Anda mungkin mengalami pelambatan (kesalahan HTTP). `503 Service Unavailable` Throttling terjadi ketika jumlah AWS KMS transaksi per detik melebihi kuota saat ini. Untuk informasi selengkapnya, lihat Kuota di Panduan Developer AWS Key Management Service .

Untuk meminta peningkatan kuota, gunakan Kuota Layanan. Untuk informasi selengkapnya, lihat Meminta peningkatan kuota. Jika Service Quotas tidak didukung di Wilayah Anda, buka kasing. AWS Dukungan

Mengaktifkan replikasi untuk objek terenkripsi

Secara default, Amazon S3 tidak mereplikasi objek yang dienkripsi dengan menggunakan enkripsi sisi server dengan () kunci (SSE-KMS) atau enkripsi sisi server AWS Key Management Service dua lapis dengan kunci (DSSE-KMS AWS KMS). AWS KMS Untuk mereplikasi objek yang dienkripsi dengan SSE-KMS atau DSS-KMS, Anda harus memodifikasi konfigurasi replikasi bucket agar Amazon S3 dapat mereplikasi objek tersebut. Contoh ini menjelaskan cara menggunakan konsol Amazon S3 dan AWS Command Line Interface (AWS CLI) untuk mengubah konfigurasi replikasi bucket guna mengaktifkan replikasi objek terenkripsi.

> **Note**
>
> Saat Kunci Bucket S3 diaktifkan untuk bucket sumber atau tujuan, konteks enkripsi akan menjadi Amazon Resource Name (ARN) bucket, bukan ARN objek. Anda harus memperbarui kebijakan IAM Anda untuk menggunakan ARN bucket untuk konteks enkripsi. Untuk informasi selengkapnya, lihat Replikasi dan Kunci Bucket S3.

> **Note**
>
> Anda dapat menggunakan Multi-wilayah AWS KMS keys di Amazon S3. Namun, Amazon S3 saat ini memperlakukan kunci multi-Wilayah selayaknya kunci satu Wilayah, dan tidak menggunakan fitur multi-Wilayah dari kunci tersebut. Untuk informasi selengkapnya, lihat Menggunakan kunci Multi-wilayah di Panduan AWS Key Management Service Pengembang.

Menggunakan konsol S3

Untuk step-by-step instruksi, lihatMengkonfigurasi replikasi untuk bucket di akun yang sama. Topik ini memberikan instruksi untuk menyetel konfigurasi replikasi ketika bucket sumber dan tujuan dimiliki oleh yang sama dan berbeda. Akun AWS

Menggunakan AWS CLI

Untuk mereplikasi objek terenkripsi dengan AWS CLI, Anda melakukan hal berikut:

- Buat bucket sumber dan tujuan, lalu aktifkan Penentuan Versi pada bucket ini.
- Buat peran layanan AWS Identity and Access Management (IAM) yang memberikan izin Amazon S3 untuk mereplikasi objek. Izin peran IAM mencakup izin yang diperlukan untuk mereplikasi objek yang dienkripsi.
- Tambahkan konfigurasi replikasi ke bucket sumber. Konfigurasi replikasi menyediakan informasi yang terkait dengan replikasi objek yang dienkripsi dengan menggunakan kunci KMS.
- Tambahkan objek terenkripsi ke bucket sumber.
- Uji penyiapan untuk mengonfirmasi bahwa objek terenkripsi Anda sedang direplikasi ke bucket tujuan.

Prosedur berikut memandu Anda menjalankan proses ini.

Untuk mereplikasi objek yang dienkripsi di sisi server (AWS CLI)

Untuk menggunakan contoh dalam prosedur ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

1. Dalam contoh ini, Anda membuat bucket source (*`amzn-s3-demo-source-bucket`*) dan destination (*`amzn-s3-demo-destination-bucket`*) secara bersamaan Akun AWS. Anda juga menetapkan profil kredensial untuk AWS CLI. Contoh ini menggunakan nama profil *`acctA`*.

Untuk informasi tentang menyetel profil kredensi dan menggunakan profil bernama, lihat Pengaturan konfigurasi dan file kredensi di AWS Command Line Interface Panduan Pengguna.

2. Gunakan perintah berikut untuk membuat bucket *amzn-s3-demo-source-bucket* dan mengaktifkan Penentuan Versi di dalamnya. Contoh perintah berikut membuat bucket *amzn-s3-demo-source-bucket* di Wilayah AS Timur (Virginia Utara) (us-east-1).

```
aws s3api create-bucket \
--bucket amzn-s3-demo-source-bucket \
--region us-east-1 \
--profile acctA
```

```
aws s3api put-bucket-versioning \
--bucket amzn-s3-demo-source-bucket \
--versioning-configuration Status=Enabled \
--profile acctA
```

3. Gunakan perintah berikut untuk membuat bucket *amzn-s3-demo-destination-bucket* dan mengaktifkan Penentuan Versi di dalamnya. Contoh perintah berikut membuat bucket *amzn-s3-demo-destination-bucket* di Wilayah AS Barat (Oregon) (us-west-2).

> **Note**
>
> Untuk menyiapkan konfigurasi replikasi saat bucket *amzn-s3-demo-source-bucket* dan *amzn-s3-demo-destination-bucket* berada di Akun AWS yang sama, Anda menggunakan profil yang sama. Contoh ini menggunakan *acctA*. Untuk mengonfigurasi replikasi ketika bucket dimiliki oleh yang berbeda Akun AWS, Anda menentukan profil yang berbeda untuk masing-masing.

```
aws s3api create-bucket \
--bucket amzn-s3-demo-destination-bucket \
--region us-west-2 \
--create-bucket-configuration LocationConstraint=us-west-2 \
--profile acctA
```

```
aws s3api put-bucket-versioning \
--bucket amzn-s3-demo-destination-bucket \
```

```
--versioning-configuration Status=Enabled \
--profile acctA
```

4. Di samping itu, Anda membuat peran layanan IAM. Anda akan menentukan peran ini dalam konfigurasi replikasi yang Anda tambahkan ke *amzn-s3-demo-source-bucket* bucket nanti. Amazon S3 mengasumsikan peran ini untuk mereplikasi objek atas nama Anda. Anda membuat peran IAM dalam dua langkah:

   - Membuat peran layanan.
   - Lampirkan kebijakan izin pada peran tersebut.

   a. Untuk membuat peran layanan IAM, lakukan hal berikut:

      i. Salin kebijakan kepercayaan berikut dan simpan ke file bernama *s3-role-trust-policy-kmsobj*`.json` di direktori saat ini di komputer lokal Anda. Kebijakan ini memberikan izin utama layanan Amazon S3 untuk mengambil peran sehingga Amazon S3 dapat melakukan tugas atas nama Anda.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"s3.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

      ii. Gunakan perintah berikut ini untuk membuat peran tersebut:

```
$ aws iam create-role \
--role-name replicationRolekmsobj \
--assume-role-policy-document file://s3-role-trust-policy-kmsobj.json  \
--profile acctA
```

   b. Selanjutnya, Anda melampirkan kebijakan izin pada peran tersebut. Kebijakan ini memberikan izin untuk berbagai bucket dan tindakan objek Amazon S3.

i. Salin kebijakan kepercayaan berikut dan simpan di berkas dengan nama *s3-role-permissions-policykmsobj*.json di direktori saat ini di komputer lokal Anda. Anda akan membuat peran IAM dan melampirkan kebijakannya nanti.

> **Important**
>
> Dalam kebijakan izin, Anda menentukan AWS KMS kunci IDs yang akan digunakan untuk enkripsi *amzn-s3-demo-source-bucket* dan *amzn-s3-demo-destination-bucket* bucket. Anda harus membuat dua kunci KMS terpisah untuk *amzn-s3-demo-source-bucket* dan *amzn-s3-demo-destination-bucket* ember. AWS KMS keys tidak dibagikan Wilayah AWS di luar tempat mereka diciptakan.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Action":[
            "s3:ListBucket",
            "s3:GetReplicationConfiguration",
            "s3:GetObjectVersionForReplication",
            "s3:GetObjectVersionAcl",
            "s3:GetObjectVersionTagging"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket",
            "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
         ]
      },
      {
         "Action":[
            "s3:ReplicateObject",
            "s3:ReplicateDelete",
            "s3:ReplicateTags"
         ],
         "Effect":"Allow",
         "Condition":{
            "StringLikeIfExists":{
```

```
                "s3:x-amz-server-side-encryption":[
                   "aws:kms",
                   "AES256",
                   "aws:kms:dsse"
                ],
                "s3:x-amz-server-side-encryption-aws-kms-key-id":[
                   "AWS KMS key IDs(in ARN format) to use for encrypting
object replicas"
                ]
             }
          },
          "Resource":"arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
       },
       {
          "Action":[
             "kms:Decrypt"
          ],
          "Effect":"Allow",
          "Condition":{
             "StringLike":{
                "kms:ViaService":"s3.us-east-1.amazonaws.com",
                "kms:EncryptionContext:aws:s3:arn":[
                   "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
                ]
             }
          },
          "Resource":[
             "AWS KMS key IDs(in ARN format) used to encrypt source
objects."
          ]
       },
       {
          "Action":[
             "kms:Encrypt"
          ],
          "Effect":"Allow",
          "Condition":{
             "StringLike":{
                "kms:ViaService":"s3.us-west-2.amazonaws.com",
                "kms:EncryptionContext:aws:s3:arn":[
                   "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
                ]
             }
          },
```

```
         "Resource":[
            "AWS KMS key IDs (in ARN format) to use for encrypting object
 replicas"
         ]
      }
   ]
}
```

ii. Buat kebijakan dan lampirkan ke peran tersebut.

```
$ aws iam put-role-policy \
--role-name replicationRolekmsobj \
--policy-document file://s3-role-permissions-policykmsobj.json \
--policy-name replicationRolechangeownerPolicy \
--profile acctA
```

5. Selanjutnya, tambahkan konfigurasi replikasi berikut ke bucket *amzn-s3-demo-source-bucket*. Ini memberi tahu Amazon S3 untuk mereplikasi objek dengan awalan *Tax/* ke bucket *amzn-s3-demo-destination-bucket*.

> **Important**
>
> Dalam konfigurasi replikasi, Anda menentukan peran IAM yang dapat diasumsikan oleh Amazon S3. Anda dapat melakukannya hanya jika Anda memiliki izin `iam:PassRole`. Profil yang Anda tentukan dalam perintah CLI harus memiliki izin ini. Untuk informasi selengkapnya, lihat Memberikan izin pengguna untuk meneruskan peran ke Layanan AWS di Panduan Pengguna IAM.

```
<ReplicationConfiguration>
 <Role>IAM-Role-ARN</Role>
 <Rule>
   <Priority>1</Priority>
   <DeleteMarkerReplication>
      <Status>Disabled</Status>
   </DeleteMarkerReplication>
   <Filter>
      <Prefix>Tax</Prefix>
   </Filter>
   <Status>Enabled</Status>
   <SourceSelectionCriteria>
```

```
      <SseKmsEncryptedObjects>
        <Status>Enabled</Status>
      </SseKmsEncryptedObjects>
    </SourceSelectionCriteria>
    <Destination>
      <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
      <EncryptionConfiguration>
        <ReplicaKmsKeyID>AWS KMS key IDs to use for encrypting object replicas</
ReplicaKmsKeyID>
      </EncryptionConfiguration>
    </Destination>
  </Rule>
</ReplicationConfiguration>
```

Untuk menambahkan konfigurasi replikasi ke bucket *amzn-s3-demo-source-bucket*, lakukan hal berikut:

a. Ini AWS CLI mengharuskan Anda untuk menentukan konfigurasi replikasi sebagai JSON. Simpan JSON berikut dalam file (*replication*.json) dalam direktori saat ini di komputer lokal Anda.

```
{
   "Role":"IAM-Role-ARN",
   "Rules":[
      {
         "Status":"Enabled",
         "Priority":1,
         "DeleteMarkerReplication":{
            "Status":"Disabled"
         },
         "Filter":{
            "Prefix":"Tax"
         },
         "Destination":{
            "Bucket":"arn:aws:s3:::amzn-s3-demo-destination-bucket",
            "EncryptionConfiguration":{
               "ReplicaKmsKeyID":"AWS KMS key IDs (in ARN format) to use for
 encrypting object replicas"
            }
         },
         "SourceSelectionCriteria":{
            "SseKmsEncryptedObjects":{
```

```
                "Status":"Enabled"
            }
         }
      }
   ]
}
```

b. Edit JSON guna memberikan nilai untuk bucket *amzn-s3-demo-destination-bucket*, *AWS KMS key IDs (in ARN format)*, dan *IAM-role-ARN*. Simpan perubahan.

c. Gunakan perintah berikut untuk menambahkan konfigurasi replikasi ke bucket *amzn-s3-demo-source-bucket* Anda. Pastikan untuk memberikan nama bucket *amzn-s3-demo-source-bucket*.

```
$ aws s3api put-bucket-replication \
--replication-configuration file://replication.json \
--bucket amzn-s3-demo-source-bucket \
--profile acctA
```

6. Uji konfigurasi untuk memverifikasi bahwa objek terenkripsi direplikasi. Di konsol Amazon S3, lakukan berikut ini:

a. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

b. Di bucket *amzn-s3-demo-source-bucket*, buat folder bernama *Tax*.

c. Tambahkan objek sampel ke folder. Pastikan untuk memilih opsi enkripsi dan tentukan kunci KMS Anda untuk mengenkripsi objek.

d. Pastikan bahwa bucket *amzn-s3-demo-destination-bucket* berisi replika objek dan dienkripsi menggunakan kunci KMS yang Anda tentukan dalam konfigurasi. Untuk informasi selengkapnya, lihat the section called “Mendapatkan status replikasi”.

Menggunakan AWS SDKs

Untuk contoh kode yang menunjukkan cara menambahkan konfigurasi replikasi, lihat Menggunakan AWS SDKs. Anda harus memodifikasi konfigurasi replikasi dengan tepat.

Mereplikasi perubahan metadata dengan sinkronisasi modifikasi replika

Sinkronisasi modifikasi replika Amazon S3 dapat membantu Anda menyimpan metadata objek seperti tag, daftar kontrol akses (ACLs), dan pengaturan Kunci Objek direplikasi antara replika dan objek

sumber. Secara default, Amazon S3 mereplikasi metadata dari objek sumber ke replika saja. Saat sinkronisasi modifikasi replika diaktifkan, Amazon S3 mereplikasi perubahan metadata yang dibuat pada salinan replika kembali ke objek sumber, membuat replikasi dua arah (replikasi dua arah).

Mengaktifkan sinkronisasi modifikasi replika

Anda dapat menggunakan sinkronisasi modifikasi replika Amazon S3 dengan aturan replikasi baru atau yang sudah ada. Anda dapat menerapkannya ke seluruh ember atau objek yang memiliki awalan tertentu.

Untuk mengaktifkan sinkronisasi modifikasi replika dengan menggunakan konsol Amazon S3, lihat. Contoh untuk mengonfigurasi replikasi langsung Topik ini memberikan petunjuk untuk mengaktifkan sinkronisasi modifikasi replika dalam konfigurasi replikasi Anda saat bucket sumber dan tujuan dimiliki oleh yang sama atau berbeda. Akun AWS

Untuk mengaktifkan sinkronisasi modifikasi replika dengan menggunakan AWS Command Line Interface (AWS CLI), Anda harus menambahkan konfigurasi replikasi ke bucket yang berisi replika yang diaktifkan. `ReplicaModifications` Untuk menyiapkan replikasi dua arah, buat aturan replikasi dari bucket sumber (*amzn-s3-demo-source-bucket*) ke bucket yang berisi replika (). *amzn-s3-demo-destination-bucket* Kemudian, buat aturan replikasi kedua dari bucket yang berisi replika (*amzn-s3-demo-destination-bucket*) ke bucket sumber (*amzn-s3-demo-source-bucket*). Ember sumber dan tujuan bisa sama atau berbeda Wilayah AWS.

> **Note**
>
> Anda harus mengaktifkan sinkronisasi modifikasi replika pada bucket sumber dan tujuan untuk mereplikasi perubahan metadata replika seperti daftar kontrol akses objek (ACLs), tag objek, atau pengaturan Kunci Objek pada objek yang direplikasi. Seperti semua aturan replikasi, Anda dapat menerapkan aturan ini ke seluruh bucket atau subset objek yang difilter oleh awalan atau tag objek.

Dalam contoh konfigurasi berikut, Amazon S3 mereplikasi perubahan metadata di bawah awalan *Tax* ke bucket*amzn-s3-demo-source-bucket*, yang berisi objek sumber.

```
{
    "Rules": [
        {
            "Status": "Enabled",
            "Filter": {
```

```
                "Prefix": "Tax"
            },
            "SourceSelectionCriteria": {
                "ReplicaModifications":{
                    "Status": "Enabled"
                }
            },
            "Destination": {
                "Bucket": "arn:aws:s3:::amzn-s3-demo-source-bucket"
            },
            "Priority": 1
        }
    ],
    "Role": "IAM-Role-ARN"
}
```

Untuk petunjuk lengkap tentang membuat aturan replikasi dengan menggunakan AWS CLI, lihatMengkonfigurasi replikasi untuk bucket di akun yang sama.

Mereplikasi penanda hapus di antara bucket

Secara default, saat Replikasi S3 diaktifkan dan objek dihapus di bucket sumber, Amazon S3 menambahkan penanda hapus hanya dalam bucket sumber. Tindakan ini membantu melindungi data di bucket tujuan dari penghapusan yang tidak disengaja atau berbahaya. Jika replikasi penanda hapus diaktifkan, penanda ini akan disalin ke bucket tujuan, dan Amazon S3 berperilaku seolah-olah objek telah dihapus di bucket sumber dan tujuan. Untuk informasi selengkapnya tentang cara kerja penanda hapus, lihat Bekerja dengan penanda hapus.

> **Note**
>
> - Hapus replikasi penanda tidak didukung untuk aturan replikasi berbasis tag. Hapus replikasi penanda juga tidak mematuhi perjanjian tingkat layanan (SLA) 15 menit yang diberikan saat Anda menggunakan Kontrol Waktu Replikasi S3 (S3 RTC).
> - Jika Anda tidak menggunakan konfigurasi replikasi terbaru versi XHTML, operasi delete mempengaruhi replikasi secara berbeda. Untuk informasi selengkapnya, lihat Bagaimana cara menghapus operasi yang memengaruhi replikasi.
> - Jika Anda mengaktifkan replikasi penanda hapus dan bucket sumber Anda memiliki aturan kedaluwarsa Siklus Hidup S3, penanda hapus yang ditambahkan oleh aturan kedaluwarsa Siklus Hidup S3 tidak akan direplikasi ke bucket tujuan.

Mengaktifkan replikasi penanda hapus

Anda dapat mulai menggunakan replikasi penanda hapus dengan aturan replikasi baru atau yang sudah ada. Anda dapat menerapkan replikasi penanda hapus ke seluruh bucket atau objek yang memiliki awalan tertentu.

Untuk mengaktifkan replikasi penanda hapus dengan menggunakan konsol Amazon S3, lihat. Menggunakan konsol S3 Topik ini memberikan petunjuk untuk mengaktifkan replikasi penanda hapus dalam konfigurasi replikasi Anda saat bucket sumber dan tujuan dimiliki oleh yang sama atau berbeda. Akun AWS

Untuk mengaktifkan replikasi penanda hapus dengan menggunakan AWS Command Line Interface (AWS CLI), Anda harus menambahkan konfigurasi replikasi ke bucket sumber dengan `DeleteMarkerReplication` diaktifkan, seperti yang ditunjukkan pada contoh konfigurasi berikut.

Dalam contoh konfigurasi replikasi berikut, penanda hapus direplikasi ke bucket tujuan *amzn-s3-demo-destination-bucket* untuk objek di bawah awalan. *Tax*

```
{
    "Rules": [
        {
            "Status": "Enabled",
            "Filter": {
                "Prefix": "Tax"
            },
            "DeleteMarkerReplication": {
                "Status": "Enabled"
            },
            "Destination": {
                "Bucket": "arn:aws:s3:::amzn-s3-demo-destination-bucket"
            },
            "Priority": 1
        }
    ],
    "Role": "IAM-Role-ARN"
}
```

Untuk petunjuk lengkap tentang membuat aturan replikasi melalui AWS CLI, lihatMengkonfigurasi replikasi untuk bucket di akun yang sama.

Mengelola atau menjeda replikasi langsung

Replikasi langsung adalah penyalinan objek otomatis dan asinkron di seluruh ember dalam hal yang sama atau berbeda. Wilayah AWS Setelah menyiapkan konfigurasi replikasi, Amazon S3 mereplikasi objek yang baru dibuat dan pembaruan objek dari bucket sumber ke satu atau beberapa bucket tujuan yang ditentukan.

Anda menggunakan konsol Amazon S3 untuk menambahkan aturan replikasi ke bucket sumber. Aturan replikasi menentukan objek bucket sumber mana yang akan direplikasi dan bucket tujuan atau bucket tempat objek yang direplikasi disimpan. Untuk informasi selengkapnya tentang replikasi, lihat Mereplikasi objek di dalam dan di seluruh Wilayah.

Anda dapat mengelola aturan replikasi pada halaman Replikasi di konsol Amazon S3. Anda dapat menambahkan, melihat, mengedit, mengaktifkan, menonaktifkan, atau menghapus aturan replikasi. Anda juga dapat mengubah prioritas aturan replikasi Anda. Untuk informasi tentang menambahkan aturan replikasi ke bucket, lihat Menggunakan konsol S3.

Untuk mengelola aturan replikasi bucket dengan menggunakan konsol Amazon S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Pada tab Bucket tujuan umum, pilih nama bucket yang Anda inginkan.
4. Pilih tab Manajemen, lalu gulir ke bawah ke Aturan replikasi.
5. Anda dapat mengubah aturan replikasi Anda dengan cara berikut:
   - Untuk mengaktifkan atau menonaktifkan aturan replikasi, pilih tombol opsi di sebelah kiri aturan. Pada menu Tindakan, pilih Aktifkan aturan atau Nonaktifkan aturan. Anda juga dapat menonaktifkan, mengaktifkan, atau menghapus semua aturan di bucket dari menu Tindakan.

> **Note**
>
> Jika Anda menonaktifkan aturan replikasi dan kemudian mengaktifkan kembali aturan tersebut, objek baru atau yang diubah yang tidak direplikasi saat aturan dinonaktifkan tidak akan direplikasi secara otomatis saat aturan diaktifkan kembali. Untuk mereplikasi objek tersebut, Anda harus menggunakan Replikasi Batch S3. Untuk informasi selengkapnya, lihat the section called “Mereplikasi objek yang ada”.

- Untuk mengubah prioritas aturan, pilih tombol opsi di sebelah kiri aturan, lalu pilih Edit aturan.

  Anda menetapkan prioritas aturan untuk menghindari konflik yang disebabkan oleh objek yang termasuk dalam cakupan lebih dari satu aturan. Dalam kasus aturan yang tumpang tindih, Amazon S3 menggunakan prioritas aturan untuk menentukan aturan mana yang berlaku. Semakin tinggi angkanya, semakin tinggi prioritasnya. Untuk informasi selengkapnya tentang prioritas aturan, lihat Elemen file konfigurasi replikasi.

Menjeda atau menghentikan replikasi

Untuk menjeda sementara replikasi dan membuatnya dilanjutkan secara otomatis nanti, Anda dapat menggunakan `aws:s3:bucket-pause-replication` tindakan di. AWS Fault Injection ServiceUntuk informasi selengkapnya, lihat aws:s3:bucket-pause-replicationdan Jeda Replikasi S3 di Panduan Pengguna.AWS Fault Injection Service

Untuk menghentikan replikasi di Amazon S3, sebaiknya nonaktifkan aturan replikasi Anda. Jika Anda menonaktifkan aturan replikasi dan kemudian mengaktifkan kembali aturan tersebut, objek baru atau yang diubah yang tidak direplikasi saat aturan dinonaktifkan tidak akan direplikasi secara otomatis saat aturan diaktifkan kembali. Untuk mereplikasi objek tersebut, Anda harus menggunakan Replikasi Batch S3. Untuk informasi selengkapnya, lihat the section called “Mereplikasi objek yang ada”.

Replikasi juga akan berhenti jika Anda menghapus peran AWS Identity and Access Management (IAM), izin AWS Key Management Service (AWS KMS), atau izin kebijakan bucket yang memberikan Amazon S3 izin yang diperlukan. Namun, kami tidak merekomendasikan pendekatan ini karena menyebabkan replikasi gagal. Amazon S3 melaporkan status replikasi objek yang terdampak sebagai `FAILED`. Jika izin kemudian dipulihkan, objek yang ditandai sebagai `FAILED` tidak direplikasi secara otomatis. Untuk mereplikasi objek tersebut, Anda harus menggunakan Replikasi Batch S3.

Mereplikasi objek yang ada dengan Batch Replication

Replikasi Batch S3 berbeda dari replikasi langsung, yang secara terus menerus dan otomatis mereplikasi objek baru di seluruh bucket Amazon S3. Sebaliknya, Replikasi Batch S3 terjadi sesuai permintaan pada objek yang ada. Anda dapat menggunakan Replikasi Batch S3 untuk mereplikasi jenis objek berikut:

- Objek yang ada sebelum konfigurasi replikasi ada
- Objek yang sebelumnya telah direplikasi

- Objek yang gagal replikasi

Anda dapat mereplikasi objek ini sesuai permintaan dengan menggunakan pekerjaan Operasi Batch.

Untuk memulai dengan Batch Replication, Anda dapat:

- Memulai Replikasi Batch untuk aturan atau tujuan replikasi baru — Anda dapat membuat tugas Replikasi Batch satu kali saat membuat aturan pertama dalam konfigurasi replikasi baru atau saat menambahkan bucket tujuan baru ke konfigurasi yang ada melalui konsol Amazon S3.
- Memulai Replikasi Batch untuk konfigurasi replikasi yang ada — Anda dapat membuat pekerjaan Replikasi Batch baru dengan menggunakan Operasi Batch S3 melalui konsol Amazon S3, AWS Command Line Interface (AWS CLI), API REST Amazon S3, atau AWS SDKs Amazon S3 REST.

Ketika tugas Replikasi Batch selesai, Anda menerima laporan penyelesaian. Untuk informasi lebih lanjut tentang cara menggunakan laporan ini untuk memeriksa pekerjaan, lihatMelacak status tugas dan laporan penyelesaian.

Pertimbangan Replikasi Batch S3

Sebelum menggunakan Replikasi Batch S3, tinjau daftar pertimbangan berikut:

- Bucket sumber Anda harus memiliki konfigurasi replikasi yang sudah ada. Untuk mengaktifkan replikasi, lihat Menyiapkan ikhtisar replikasi langsung dan Contoh untuk mengonfigurasi replikasi langsung.
- Jika Siklus Hidup S3 dikonfigurasi untuk bucket, sebaiknya nonaktifkan aturan siklus hidup Anda saat tugas Replikasi Batch aktif. Melakukannya membantu memastikan kesetaraan antara ember sumber dan tujuan. Jika tidak, bucket ini bisa menyimpang, dan bucket tujuan tidak akan menjadi replika yang tepat dari bucket sumber. Sebagai contoh, pertimbangkan alur perencanaan berikut ini:
  - Bucket sumber Anda memiliki beberapa versi objek dan penanda hapus pada objek tersebut.
  - Bucket sumber dan tujuan Anda memiliki konfigurasi siklus hidup untuk menghapus penanda hapus yang kedaluwarsa.

  Dalam skenario ini, Replikasi Batch mungkin mereplikasi penanda hapus ke bucket tujuan sebelum mereplikasi versi objek. Perilaku ini dapat mengakibatkan konfigurasi siklus hidup Anda menandai penanda hapus sebagai kedaluwarsa dan penanda hapus dihapus dari keranjang tujuan sebelum versi objek direplikasi.

- Peran AWS Identity and Access Management (IAM) yang Anda tentukan untuk menjalankan pekerjaan Operasi Batch harus memiliki izin yang diperlukan untuk melakukan operasi Replikasi Batch yang mendasarinya. Untuk informasi selengkapnya tentang cara membuat peran IAM, lihat Mengkonfigurasi peran IAM untuk Replikasi Batch S3.
- Replikasi Batch memerlukan manifes, yang dapat dihasilkan oleh Amazon S3. Manifes yang dihasilkan harus disimpan Wilayah AWS sama dengan bucket sumber. Jika Anda memilih untuk tidak membuat manifes, Anda dapat menyediakan laporan Inventaris Amazon S3 atau file CSV yang berisi objek yang ingin direplikasi. Untuk informasi selengkapnya, lihat the section called "Menentukan manifes untuk tugas Replikasi Batch".
- Batch Replication tidak mendukung replikasi ulang objek yang dihapus dengan menentukan ID versi objek dari bucket tujuan. Untuk mereplikasi ulang objek ini, Anda dapat menyalin objek sumber di tempat dengan tugas Penyalinan Batch. Menyalin objek tersebut di tempat akan membuat versi baru objek di bucket sumber dan secara otomatis memulai replikasi ke bucket tujuan. Menghapus dan membuat ulang bucket tujuan tidak memulai replikasi.

  Untuk informasi selengkapnya tentang Batch Copy, lihatContoh yang menggunakan Operasi Batch untuk menyalin objek.
- Jika Anda menggunakan aturan replikasi pada bucket sumber, pastikan untuk memperbarui konfigurasi replikasi Anda dengan memberikan peran IAM yang dilampirkan pada aturan replikasi izin yang tepat untuk mereplikasi objek. Peran IAM ini harus memiliki izin yang diperlukan untuk melakukan replikasi pada bucket sumber dan tujuan.
- Jika Anda mengirimkan beberapa pekerjaan Replikasi Batch untuk bucket yang sama dalam jangka waktu singkat, Amazon S3 menjalankan pekerjaan tersebut secara bersamaan.
- Jika Anda mengirimkan beberapa pekerjaan Replikasi Batch untuk dua bucket yang berbeda, ketahuilah bahwa Amazon S3 mungkin tidak menjalankan semua pekerjaan secara bersamaan. Jika Anda melebihi jumlah pekerjaan Replikasi Batch yang dapat berjalan pada satu waktu di akun Anda, Amazon S3 menghentikan sementara pekerjaan dengan prioritas lebih rendah untuk mengerjakan pekerjaan dengan prioritas yang lebih tinggi. Setelah pekerjaan prioritas yang lebih tinggi selesai, setiap pekerjaan yang dijeda menjadi aktif kembali.
- Replikasi Batch tidak didukung untuk objek yang disimpan di kelas penyimpanan S3 Glacier Flexible Retrieval dan S3 Glacier Deep Archive.
- Untuk mereplikasi objek S3 Intelligent-Tiering yang disimpan di tingkat penyimpanan Akses Arsip atau Akses Arsip Dalam, Anda harus terlebih dahulu memulai permintaan pemulihan dan menunggu hingga objek dipindahkan ke tingkat Akses Sering.
- Satu pekerjaan Batch Replication dapat mendukung manifes dengan hingga 20 miliar objek.

Menentukan manifes untuk tugas Replikasi Batch

Manifes adalah objek Amazon S3 yang berisi kunci objek yang harus ditindaklanjuti Amazon S3. Jika Anda ingin membuat pekerjaan Replikasi Batch, Anda harus menyediakan manifes buatan pengguna atau meminta Amazon S3 menghasilkan manifes berdasarkan konfigurasi replikasi Anda.

Jika Anda menyediakan manifes buatan pengguna, manifes tersebut harus dalam bentuk laporan Inventaris Amazon S3 atau file CSV. Jika objek dalam manifes Anda berada dalam bucket berversi, Anda harus menentukan versi IDs untuk objek tersebut. Hanya objek dengan ID versi yang ditentukan dalam manifes yang akan direplikasi. Untuk mempelajari penentuan manifes lebih lanjut, lihat Menentukan manifes.

Jika Anda memilih agar Amazon S3 menghasilkan file manifes atas nama Anda, objek yang terdaftar menggunakan bucket sumber, awalan, dan tag yang sama dengan konfigurasi replikasi di bucket sumber. Dengan manifes yang dihasilkan, Amazon S3 mereplikasi semua versi objek yang memenuhi syarat.

> **Note**
>
> Jika Anda memilih agar Amazon S3 menghasilkan manifes, manifes harus disimpan Wilayah AWS sama dengan bucket sumber.

Filter untuk tugas Replikasi Batch

Saat membuat pekerjaan Replikasi Batch, Anda dapat secara opsional menentukan filter tambahan, seperti tanggal pembuatan objek dan status replikasi, untuk mengurangi cakupan pekerjaan.

Anda dapat memfilter objek yang akan mereplikasi berdasarkan nilai `ObjectReplicationStatuses`, dengan memberikan satu atau beberapa nilai berikut:

- `"NONE"`–Menunjukkan bahwa Amazon S3 belum pernah mencoba mereplikasi objek sebelumnya.
- `"FAILED"`— Menunjukkan bahwa Amazon S3 telah mencoba, tetapi gagal, untuk mereplikasi objek sebelumnya.
- `"COMPLETED"`–Menunjukkan bahwa Amazon S3 telah berhasil mereplikasi objek sebelumnya.
- `"REPLICA"`— Menunjukkan bahwa objek ini adalah replika yang telah direplikasi Amazon S3 dari ember sumber lain.

Untuk informasi selengkapnya tentang status replikasi, lihat Mendapatkan informasi status replikasi.

Jika Anda tidak memfilter tugas Replikasi Batch, Operasi Batch mencoba mereplikasi semua objek (apa pun objek`ObjectReplicationStatus`) dalam manifes Anda yang cocok dengan aturan dalam konfigurasi replikasi Anda, kecuali untuk objek tertentu yang tidak direplikasi secara default. Untuk informasi selengkapnya, lihat the section called “Apa yang tidak direplikasi dengan konfigurasi replikasi?”

Bergantung pada tujuan Anda, Anda dapat menetapkan `ObjectReplicationStatuses` ke satu atau beberapa nilai berikut:

- Untuk mereplikasi hanya objek yang ada yang belum pernah direplikasi, hanya sertakan. `"NONE"`
- Untuk mencoba lagi mereplikasi hanya objek yang sebelumnya gagal direplikasi, hanya sertakan. `"FAILED"`
- Untuk mereplikasi objek yang ada dan mencoba lagi mereplikasi objek yang sebelumnya gagal direplikasi, sertakan keduanya dan. `"NONE" "FAILED"`
- Untuk mengisi ulang bucket tujuan dengan objek yang telah direplikasi ke tujuan lain, sertakan. `"COMPLETED"`
- Untuk mereplikasi objek yang sebelumnya direplikasi, sertakan. `"REPLICA"`

Laporan penyelesaian Replikasi Batch

Saat Anda membuat tugas Replikasi Batch, Anda dapat meminta laporan penyelesaian CSV. Laporan ini menunjukkan objek, kode keberhasilan atau kegagalan replikasi, output, dan deskripsi. Untuk informasi selengkapnya tentang laporan pelacakan dan penyelesaian pekerjaan, lihatLaporan penyelesaian.

Untuk daftar kode kegagalan replikasi dan deskripsi, lihat. Penyebab kegagalan replikasi Amazon S3

Untuk informasi tentang pemecahan masalah Replikasi Batch, lihat. Kesalahan Replikasi Batch

Memulai Replikasi Batch

Untuk mempelajari cara menggunakan Replikasi Batch lebih lanjut, lihat Tutorial: Mereplikasi objek yang ada di bucket Amazon S3 Anda dengan Replikasi Batch S3.

Mengkonfigurasi peran IAM untuk Replikasi Batch S3

Karena Replikasi Batch Amazon S3 adalah jenis pekerjaan Operasi Batch, Anda harus membuat peran AWS Identity and Access Management (IAM) untuk memberikan izin Operasi Batch untuk melakukan tindakan atas nama Anda. Anda juga harus melampirkan kebijakan IAM Replikasi Batch ke peran IAM Operasi Batch.

Gunakan prosedur berikut untuk membuat kebijakan dan peran IAM yang memberikan izin Operasi Batch untuk memulai pekerjaan Replikasi Batch.

Untuk membuat kebijakan untuk Replikasi Batch

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di bagian Manajemen akses, pilih Kebijakan.
3. Pilih Buat kebijakan.
4. Pada halaman Tentukan izin, pilih JSON.
5. Masukkan salah satu kebijakan berikut, tergantung pada apakah manifes Anda dihasilkan oleh Amazon S3 atau apakah Anda menyediakan manifes Anda sendiri. Untuk informasi selengkapnya tentang manifest, lihat Menentukan manifes untuk tugas Replikasi Batch.

   Sebelum menggunakan kebijakan ini, ganti kebijakan berikut dengan nama bucket sumber replikasi, bucket manifes, dan bucket laporan penyelesaian. *`user input placeholders`*

   > **Note**
   >
   > Peran IAM Anda untuk Replikasi Batch memerlukan izin yang berbeda, tergantung pada apakah Anda membuat manifes atau memasok, jadi pastikan Anda memilih kebijakan yang sesuai dari contoh berikut.

   Kebijakan jika menggunakan dan menyimpan manifes yang dihasilkan Amazon S3

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Action":[
            "s3:InitiateReplication"
```

```
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
         ]
      },
      {
         "Action":[
            "s3:GetReplicationConfiguration",
            "s3:PutInventoryConfiguration"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket"
         ]
      },
      {
         "Action":[
            "s3:GetObject",
            "s3:GetObjectVersion"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject"
         ],
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-completion-report-bucket/*",
            "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
         ]
      }
   ]
}
```

Kebijakan jika menggunakan manifes yang disediakan pengguna

```
{
   "Version":"2012-10-17",
```

```
   "Statement":[
      {
         "Action":[
            "s3:InitiateReplication"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
         ]
      },
            {
         "Action":[
            "s3:GetObject",
            "s3:GetObjectVersion"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject"
         ],
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-completion-report-bucket/*"
         ]
      }
   ]
}
```

6. Pilih Berikutnya.
7. Tentukan nama untuk kebijakan tersebut, lalu pilih Buat kebijakan.

Untuk membuat peran IAM untuk Replikasi Batch

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di bawah Manajemen akses, pilih Peran.
3. Pilih Buat peran.

4. Pilih Layanan AWSsebagai jenis entitas tepercaya. Di bagian Use case, pilih S3 sebagai layanan, dan Operasi Batch S3 sebagai kasus penggunaan.
5. Pilih Berikutnya. Halaman Tambahkan izin muncul. Di kotak pencarian, cari kebijakan yang Anda buat di prosedur sebelumnya. Pilih kotak centang di sebelah nama kebijakan, lalu pilih Berikutnya.
6. Pada halaman Nama, tinjau, dan buat, tentukan nama untuk peran IAM Anda.
7. Di bagian Langkah 1: Kepercayaan identitas, verifikasi bahwa peran IAM Anda menggunakan kebijakan kepercayaan berikut:

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"batchoperations.s3.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

8. Di bagian Langkah 2: Tambahkan izin, verifikasi bahwa peran IAM Anda menggunakan kebijakan yang Anda buat sebelumnya.
9. Pilih Buat peran.

Buat pekerjaan Replikasi Batch untuk aturan atau tujuan replikasi baru

Di Amazon S3, replikasi langsung tidak mereplikasi objek apa pun yang sudah ada di bucket sumber Anda sebelum Anda membuat konfigurasi replikasi. Replikasi langsung secara otomatis hanya mereplikasi objek baru dan yang diperbarui yang ditulis ke bucket setelah konfigurasi replikasi dibuat. Untuk mereplikasi objek yang sudah ada, Anda dapat menggunakan Replikasi Batch S3 untuk mereplikasi objek ini sesuai permintaan.

Saat membuat aturan pertama dalam konfigurasi replikasi langsung baru atau menambahkan bucket tujuan baru ke konfigurasi replikasi yang ada melalui konsol Amazon S3, Anda dapat membuat pekerjaan Replikasi Batch secara opsional. Anda dapat menggunakan tugas Replikasi Batch ini untuk mereplikasi objek yang ada di bucket sumber ke bucket tujuan.

Untuk menggunakan Replikasi Batch untuk konfigurasi yang ada tanpa menambahkan bucket tujuan baru, lihatMembuat tugas Replikasi Batch untuk aturan replikasi yang ada.

Prasyarat

Sebelum membuat tugas Replikasi Batch, Anda harus membuat peran Operasi Batch AWS Identity and Access Management (IAM) untuk memberikan izin Amazon S3 untuk melakukan tindakan atas nama Anda. Untuk informasi selengkapnya, lihat Mengkonfigurasi peran IAM untuk Replikasi Batch S3.

Menggunakan Replikasi Batch untuk aturan atau tujuan replikasi baru melalui konsol Amazon S3

Saat membuat aturan pertama dalam konfigurasi replikasi baru atau menambahkan bucket tujuan baru ke konfigurasi yang ada melalui konsol Amazon S3, Anda dapat memilih untuk membuat tugas Replikasi Batch untuk mereplikasi objek yang ada di bucket sumber.

Untuk membuat pekerjaan Replikasi Batch saat membuat atau memperbarui konfigurasi replikasi

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Dalam daftar Bucket tujuan umum, pilih nama bucket yang berisi objek yang ingin Anda tiru.
4. Untuk membuat aturan replikasi baru atau mengedit aturan yang ada, pilih tab Manajemen, dan gulir ke bawah ke Aturan replikasi:
   - Untuk membuat aturan replikasi baru, pilih Buat aturan replikasi. Untuk contoh cara mengatur aturan replikasi dasar, lihatContoh untuk mengonfigurasi replikasi langsung.
   - Untuk mengedit aturan replikasi yang ada, pilih tombol opsi di sebelah nama aturan, lalu pilih Edit aturan.
5. Buat aturan replikasi baru Anda atau edit tujuan untuk aturan replikasi yang ada, lalu pilih Simpan.

   Setelah membuat aturan pertama dalam konfigurasi replikasi baru atau mengedit konfigurasi yang ada untuk menambahkan tujuan baru, dialog Replikasi objek lama? akan muncul, yang memberi Anda opsi untuk membuat tugas Replikasi Batch.
6. Jika Anda ingin membuat dan menjalankan pekerjaan ini sekarang, pilih Ya, replikasi objek yang ada.

Jika Anda ingin membuat pekerjaan Replikasi Batch di lain waktu, pilih Tidak, jangan mereplikasi objek yang ada.

7. Jika Anda memilih Ya, mereplikasi objek yang ada, halaman pekerjaan Buat Operasi Batch akan muncul. Pekerjaan Replikasi Batch S3 memiliki pengaturan berikut:

   Opsi Job run

   Jika Anda ingin pekerjaan Replikasi Batch S3 segera berjalan, pilih Jalankan pekerjaan secara otomatis saat sudah siap. Jika Anda ingin menjalankan pekerjaan di lain waktu, pilih Tunggu untuk menjalankan pekerjaan ketika sudah siap.

   > **Note**
   >
   > Jika Anda memilih Jalankan pekerjaan secara otomatis saat sudah siap, Anda tidak akan dapat membuat dan menyimpan manifes Operasi Batch. Untuk menyimpan manifes Operasi Batch, pilih Tunggu untuk menjalankan pekerjaan ketika sudah siap.

   Manifes Operasi Batch

   Jika Anda memilih Tunggu untuk menjalankan pekerjaan saat sudah siap, bagian manifes Operasi Batch akan muncul. Manifes adalah daftar semua objek berisi tindakan tertentu yang ingin Anda jalankan. Anda dapat memilih untuk menyimpan manifes. Mirip dengan file Persediaan S3, manifes akan disimpan sebagai file CSV dan disimpan dalam bucket. Untuk mempelajari manifes Operasi Batch lebih lanjut, lihat Menentukan manifes.

   Laporan penyelesaian

   Operasi Batch S3 mengeksekusi satu tugas untuk setiap objek yang ditentukan dalam manifes. Laporan penyelesaian memberikan cara mudah untuk melihat hasil tugas Anda dalam format terkonsolidasi tanpa perlu pengaturan tambahan. Anda dapat meminta laporan penyelesaian untuk semua tugas atau hanya untuk tugas yang gagal. Untuk mempelajari laporan penyelesaian lebih lanjut, lihat Laporan penyelesaian.

   Izin

   Salah satu penyebab paling umum kegagalan replikasi adalah izin yang tidak memadai dalam peran yang disediakan AWS Identity and Access Management (IAM). Untuk informasi lebih lanjut tentang pembuatan peran, lihat Mengkonfigurasi peran IAM untuk Replikasi Batch

S3. Pastikan Anda membuat atau memilih peran IAM yang memiliki izin yang diperlukan untuk Replikasi Batch.

8. Pilih Simpan.

Membuat tugas Replikasi Batch untuk aturan replikasi yang ada

Di Amazon S3, replikasi langsung tidak mereplikasi objek apa pun yang sudah ada di bucket sumber Anda sebelum Anda membuat konfigurasi replikasi. Replikasi langsung secara otomatis hanya mereplikasi objek baru dan yang diperbarui yang ditulis ke bucket setelah konfigurasi replikasi dibuat. Untuk mereplikasi objek yang sudah ada, Anda dapat menggunakan Replikasi Batch S3 untuk mereplikasi objek ini sesuai permintaan.

Anda dapat mengonfigurasi Replikasi Batch S3 untuk konfigurasi replikasi yang ada dengan menggunakan konsol AWS SDKs, AWS Command Line Interface (AWS CLI), atau Amazon S3. Untuk gambaran umum tentang Replikasi Batch, lihatMereplikasi objek yang ada dengan Batch Replication.

Ketika tugas Replikasi Batch selesai, Anda menerima laporan penyelesaian. Untuk informasi selengkapnya tentang cara menggunakan laporan untuk memeriksa tugas, lihat Melacak status tugas dan laporan penyelesaian.

Prasyarat

Sebelum membuat tugas Replikasi Batch, Anda harus membuat peran Operasi Batch AWS Identity and Access Management (IAM) untuk memberikan izin Amazon S3 untuk melakukan tindakan atas nama Anda. Untuk informasi selengkapnya, lihat Mengkonfigurasi peran IAM untuk Replikasi Batch S3.

Menggunakan konsol S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Operasi Batch.
3. Pilih Buat tugas.
4. Verifikasi bahwa Wilayah AWSbagian tersebut menunjukkan Wilayah tempat Anda ingin membuat pekerjaan.

5. Di bagian Manifest, tentukan format manifes yang ingin Anda gunakan. Manifes adalah daftar semua objek berisi tindakan tertentu yang ingin Anda jalankan. Untuk mempelajari manifes Operasi Batch lebih lanjut, lihat Menentukan manifes.

   - Jika Anda memiliki manifes yang sudah siap, pilih laporan persediaan S3 (manifest.json) atau CSV. Jika manifes Anda ada di bucket berversi, Anda dapat menentukan ID versi untuk manifes tersebut. Jika Anda tidak menentukan ID versi, Operasi Batch menggunakan versi manifes Anda saat ini. Untuk informasi selengkapnya tentang membuat manifes, lihatMenentukan manifes.

     > **Note**
     >
     > Jika objek dalam manifes Anda berada dalam bucket berversi, Anda harus menentukan versi IDs untuk objek tersebut. Untuk informasi selengkapnya, lihat Menentukan manifes.

   - Untuk membuat manifes berdasarkan konfigurasi replikasi Anda, pilih Buat manifes menggunakan konfigurasi Replikasi S3. Kemudian pilih keranjang sumber konfigurasi replikasi Anda.

6. (Opsional) Jika Anda memilih Buat manifes menggunakan konfigurasi Replikasi S3, Anda dapat menyertakan filter tambahan, seperti tanggal pembuatan objek dan status replikasi. Untuk contoh cara memfilter berdasarkan status replikasi, lihatMenentukan manifes untuk tugas Replikasi Batch.
7. (Opsional) Jika Anda memilih Buat manifes menggunakan konfigurasi Replikasi S3, Anda dapat menyimpan manifes yang dihasilkan. Untuk menyimpan manifes ini, pilih manifes Simpan Operasi Batch. Kemudian tentukan bucket tujuan untuk manifes dan pilih apakah akan mengenkripsi manifes.

   > **Note**
   >
   > Manifes yang dihasilkan harus disimpan Wilayah AWS sama dengan bucket sumber.

8. Pilih Berikutnya.
9. Pada halaman Operasi, pilih Replikasi, lalu pilih Berikutnya.
10. (Opsional) Berikan Deskripsi.
11. Sesuaikan Prioritas tugas jika diperlukan. Angka yang lebih tinggi menunjukkan prioritas yang lebih tinggi. Amazon S3 mencoba menjalankan tugas dengan prioritas lebih tinggi sebelum

tugas dengan prioritas lebih rendah. Untuk informasi selengkapnya tentang prioritas tugas, lihat Menetapkan prioritas tugas.

12. (Opsional) Buat laporan penyelesaian. Untuk menghasilkan laporan ini, pilih Hasilkan laporan penyelesaian.

    Jika Anda memilih untuk membuat laporan penyelesaian, Anda harus memilih untuk melaporkan Hanya tugas gagal atau Semua tugas, dan menyediakan bucket tujuan untuk laporan tersebut.

13. Di bagian Izin, pastikan Anda memilih peran IAM yang memiliki izin yang diperlukan untuk Replikasi Batch. Salah satu penyebab paling umum kegagalan replikasi adalah izin yang tidak memadai dalam peran IAM yang disediakan. Untuk informasi lebih lanjut tentang pembuatan peran, lihat Mengkonfigurasi peran IAM untuk Replikasi Batch S3.
14. (Opsional) Tambahkan tag pekerjaan ke tugas Replikasi Batch.
15. Pilih Selanjutnya.
16. Tinjau konfigurasi pekerjaan Anda, lalu pilih Buat pekerjaan.

Menggunakan AWS CLI manifes dengan S3

`create-job`Perintah contoh berikut membuat pekerjaan Replikasi Batch S3 dengan menggunakan manifes yang dihasilkan S3 untuk. Akun AWS *111122223333* Contoh ini mereplikasi objek yang ada dan objek yang sebelumnya gagal untuk mereplikasi. Untuk informasi tentang pemfilteran berdasarkan status replikasi, lihat. Menentukan manifes untuk tugas Replikasi Batch

Untuk menggunakan perintah ini, ganti *user input placeholders* dengan informasi Anda sendiri. Ganti peran IAM `role/`*batch-Replication-IAM-policy* dengan peran IAM yang sebelumnya Anda buat. Untuk informasi selengkapnya, lihat Mengkonfigurasi peran IAM untuk Replikasi Batch S3.

```
aws s3control create-job --account-id 111122223333 \
--operation '{"S3ReplicateObject":{}}' \
--report '{"Bucket":"arn:aws:s3:::amzn-s3-demo-completion-report-bucket",\
"Prefix":"batch-replication-report", \
"Format":"Report_CSV_20180820","Enabled":true,"ReportScope":"AllTasks"}' \
--manifest-generator '{"S3JobManifestGenerator": {"ExpectedBucketOwner":
 "111122223333", \
"SourceBucket": "arn:aws:s3:::amzn-s3-demo-source-bucket", \
"EnableManifestOutput": false, "Filter": {"EligibleForReplication": true, \
"ObjectReplicationStatuses": ["NONE","FAILED"]}}}' \
--priority 1 \
```

```
--role-arn arn:aws:iam::111122223333:role/batch-Replication-IAM-policy \
--no-confirmation-required \
--region source-bucket-region
```

> **Note**
>
> Anda harus memulai pekerjaan dari yang Wilayah AWS sama dengan bucket sumber replikasi.

Setelah berhasil memulai tugas Replikasi Batch, Anda menerima ID tugas sebagai respons. Anda dapat memantau pekerjaan ini dengan menggunakan `describe-job` perintah berikut. Untuk menggunakan perintah ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3control describe-job --account-id 111122223333 --job-id job-id --region source-bucket-region
```

Menggunakan AWS CLI dengan manifes yang disediakan pengguna

Contoh berikut membuat pekerjaan Replikasi Batch S3 dengan menggunakan manifes yang ditentukan pengguna untuk. Akun AWS *111122223333* Jika objek dalam manifes Anda berada dalam bucket berversi, Anda harus menentukan versi IDs untuk objek tersebut. Hanya objek dengan ID versi yang ditentukan dalam manifes yang akan direplikasi. Untuk informasi selengkapnya tentang membuat manifes, lihatMenentukan manifes.

Untuk menggunakan perintah ini, ganti *`user input placeholders`* dengan informasi Anda sendiri. Ganti peran IAM `role/`*`batch-Replication-IAM-policy`* dengan peran IAM yang sebelumnya Anda buat. Untuk informasi selengkapnya, lihat Mengkonfigurasi peran IAM untuk Replikasi Batch S3.

```
aws s3control create-job --account-id 111122223333 \
--operation '{"S3ReplicateObject":{}}' \
--report '{"Bucket":"arn:aws:s3:::amzn-s3-demo-completion-report-bucket",\
"Prefix":"batch-replication-report", \
"Format":"Report_CSV_20180820","Enabled":true,"ReportScope":"AllTasks"}' \
--manifest '{"Spec":{"Format":"S3BatchOperations_CSV_20180820",\
"Fields":["Bucket","Key","VersionId"]},\
"Location":{"ObjectArn":"arn:aws:s3:::amzn-s3-demo-manifest-bucket/manifest.csv",\
"ETag":"Manifest Etag"}}' \
```

```
--priority 1 \
--role-arn arn:aws:iam::111122223333:role/batch-Replication-IAM-policy \
--no-confirmation-required \
--region source-bucket-region
```

> **Note**
>
> Anda harus memulai pekerjaan dari yang Wilayah AWS sama dengan bucket sumber replikasi.

Setelah berhasil memulai tugas Replikasi Batch, Anda menerima ID tugas sebagai respons. Anda dapat memantau pekerjaan ini dengan menggunakan `describe-job` perintah berikut.

```
aws s3control describe-job --account-id 111122223333 --job-id job-id --region source-bucket-region
```

Memecahkan masalah replikasi

Bagian ini mencantumkan tips pemecahan masalah untuk Replikasi Amazon S3 dan informasi tentang kesalahan Replikasi Batch S3.

Topik

- Kiat pemecahan masalah untuk Replikasi S3
- Kesalahan Replikasi Batch

Kiat pemecahan masalah untuk Replikasi S3

Jika replika objek tidak muncul di bucket tujuan setelah Anda mengonfigurasi replikasi, gunakan kiat pemecahan masalah ini untuk mengidentifikasi dan memperbaiki masalah.

- Mayoritas objek bereplikasi dalam 15 menit. Waktu yang dibutuhkan Amazon S3 untuk mereplikasi suatu objek tergantung pada beberapa faktor, termasuk pasangan Wilayah sumber dan tujuan, serta ukuran objek. Untuk objek besar, replikasi dapat memakan waktu hingga beberapa jam. Untuk visibilitas waktu replikasi, Anda dapat menggunakan Kontrol Waktu Replikasi S3 (S3 RTC).

  Jika objek yang direplikasi berukuran besar, tunggu beberapa saat sebelum memeriksa apakah objek tersebut muncul di tempat tujuan. Anda juga dapat memeriksa status replikasi objek sumber.

Jika status replikasi objek`PENDING`, Amazon S3 belum menyelesaikan replikasi. Jika status replikasi objek adalah `FAILED`, periksa konfigurasi replikasi yang ditetapkan pada bucket sumber.

Selain itu, untuk menerima informasi tentang kegagalan selama replikasi, Anda dapat mengatur replikasi Notifikasi Peristiwa Amazon S3 agar menerima peristiwa kegagalan. Untuk informasi lebih lanjut, lihat Menerima peristiwa kegagalan replikasi dengan Notifikasi Peristiwa Amazon S3.

- Untuk memeriksa status replikasi objek, Anda dapat memanggil operasi `HeadObject` API. Operasi `HeadObject` API mengembalikan`PENDING`,`COMPLETED`, atau status `FAILED` replikasi objek. Dalam menanggapi panggilan `HeadObject` API, status replikasi dikembalikan di `x-amz-replication-status` header.

  > **Note**
  >
  > Untuk menjalankan `HeadObject`, Anda harus memiliki akses baca ke objek yang Anda minta. Permintaan `HEAD` memiliki opsi yang sama dengan permintaan `GET`, tanpa melakukan operasi `GET`. Misalnya, untuk menjalankan `HeadObject` permintaan dengan menggunakan AWS Command Line Interface (AWS CLI), Anda dapat menjalankan perintah berikut. Ganti *`user input placeholders`* dengan informasi Anda sendiri.
  >
  > ```
  > aws s3api head-object --bucket amzn-s3-demo-source-bucket --key index.html
  > ```

- Jika `HeadObject` mengembalikan objek dengan status `FAILED` replikasi, Anda dapat menggunakan Replikasi Batch S3 untuk mereplikasi objek yang gagal tersebut. Untuk informasi selengkapnya, lihat the section called “Mereplikasi objek yang ada”. Atau, Anda dapat mengunggah ulang objek yang gagal ke bucket sumber, yang akan memulai replikasi untuk objek baru.
- Dalam konfigurasi replikasi pada bucket sumber, verifikasi hal berikut:
  - Amazon Resource Name (ARN) dari bucket tujuan sudah benar.
  - Prefiks nama kunci sudah benar. Misalnya, jika Anda mengatur konfigurasi agar mereplikasi objek dengan prefiks `Tax`, maka hanya objek dengan nama kunci seperti `Tax/document1` atau `Tax/document2` yang direplikasi. Objek dengan nama kunci `document3` tidak direplikasi.
  - Status dari aturan replikasi adalah `Enabled`.
- Verifikasi bahwa pembuatan versi belum ditangguhkan pada bucket apa pun dalam konfigurasi replikasi. Bucket sumber dan tujuan harus memiliki penentuan versi yang diaktifkan.

- Jika aturan replikasi disetel ke Ubah kepemilikan objek ke pemilik bucket tujuan, maka peran AWS Identity and Access Management (IAM) yang digunakan untuk replikasi harus memiliki izin. `s3:ObjectOwnerOverrideToBucketOwner` Izin ini diberikan pada sumber daya (dalam hal ini, bucket tujuan). Misalnya, pernyataan `Resource` berikut menunjukkan cara memberikan izin ini di bucket tujuan:

```
{
  "Effect":"Allow",
  "Action":[
    "s3:ObjectOwnerOverrideToBucketOwner"
  ],
  "Resource":"arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
}
```

- Jika bucket tujuan dimiliki oleh akun lain, pemilik bucket tujuan juga harus memberikan izin `s3:ObjectOwnerOverrideToBucketOwner` kepada pemilik bucket sumber melalui kebijakan bucket tujuan. Untuk menggunakan kebijakan bucket contoh berikut, ganti *`user input placeholders`* dengan informasi Anda sendiri:

```
{
  "Version": "2012-10-17",
  "Id": "Policy1644945280205",
  "Statement": [
    {
      "Sid": "Stmt1644945277847",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789101:role/s3-replication-role"
      },
      "Action": [
        "s3:ReplicateObject",
        "s3:ReplicateTags",
        "s3:ObjectOwnerOverrideToBucketOwner"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
    }
  ]
}
```

> **Note**
>
> Jika setelan kepemilikan objek bucket tujuan menyertakan Diberlakukan oleh pemilik bucket, maka Anda tidak perlu memperbarui setelan ke Ubah kepemilikan objek ke pemilik bucket tujuan dalam aturan replikasi. Perubahan kepemilikan objek akan terjadi secara default. Untuk informasi selengkapnya tentang mengubah kepemilikan replika, lihat Mengubah pemilik replika.

- Jika Anda menyetel konfigurasi replikasi dalam skenario lintas akun, di mana bucket sumber dan tujuan dimiliki oleh yang berbeda Akun AWS, bucket tujuan tidak dapat dikonfigurasi sebagai bucket Requester Pays. Untuk informasi selengkapnya, lihat Menggunakan Requester Pays ember tujuan umum untuk transfer dan penggunaan penyimpanan.
- Jika objek sumber bucket dienkripsi dengan menggunakan enkripsi sisi server dengan AWS Key Management Service (AWS KMS) keys (SSE-KMS), maka aturan replikasi harus dikonfigurasi untuk menyertakan objek -enkripsi. AWS KMS Pastikan untuk memilih Replikasi objek yang dienkripsi AWS KMS di bawah pengaturan Enkripsi Anda di konsol Amazon S3. Kemudian, pilih AWS KMS kunci untuk mengenkripsi objek tujuan.

> **Note**
>
> Jika bucket tujuan berada di akun yang berbeda, tentukan kunci yang dikelola AWS KMS pelanggan yang dimiliki oleh akun tujuan. Jangan gunakan kunci terkelola Amazon S3 default (`aws/s3`). Menggunakan kunci default mengenkripsi objek dengan kunci terkelola Amazon S3 yang dimiliki oleh akun sumber, mencegah objek dibagikan dengan akun lain. Akibatnya, akun tujuan tidak akan dapat mengakses objek di bucket tujuan.

Untuk menggunakan AWS KMS kunci milik akun tujuan untuk mengenkripsi objek tujuan, akun tujuan harus memberikan `kms:GenerateDataKey` dan `kms:Encrypt` izin untuk peran replikasi dalam kebijakan kunci KMS. Untuk menggunakan contoh pernyataan berikut ini dalam kebijakan kunci KMS Anda, ganti *`user input placeholders`* dengan informasi Anda sendiri:

```
{
    "Sid": "AllowS3ReplicationSourceRoleToUseTheKey",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::123456789101:role/s3-replication-role"
```

```
    },
    "Action": ["kms:GenerateDataKey", "kms:Encrypt"],
    "Resource": "*"
}
```

Jika Anda menggunakan tanda bintang (*) untuk pernyataan `Resource` dalam kebijakan kunci AWS KMS , kebijakan akan memberikan izin untuk menggunakan kunci KMS hanya pada peran replikasi. Kebijakan ini tidak mengizinkan peran replikasi untuk meningkatkan izinnya.

Secara default, kebijakan kunci KMS memberikan izin penuh kepada pengguna root ke kunci tersebut. Izin ini dapat didelegasikan ke pengguna lain di akun yang sama. Kecuali terdapat pernyataan `Deny` dalam kebijakan kunci KMS sumber, menggunakan kebijakan IAM untuk memberikan izin peran replikasi ke kunci KMS sumber sudah cukup.

> **Note**
>
> Kebijakan utama KMS yang membatasi akses ke rentang CIDR tertentu, titik akhir virtual private cloud (VPC), atau titik akses S3 dapat menyebabkan replikasi gagal.

Jika kunci KMS sumber atau tujuan memberikan izin berdasarkan konteks enkripsi, konfirmasikan bahwa Kunci Bucket Amazon S3 diaktifkan untuk bucket. Jika bucket mengaktifkan Kunci Bucket S3, konteks enkripsi harus berupa sumber daya tingkat bucket, seperti ini:

```
"kms:EncryptionContext:arn:aws:arn": [
     "arn:aws:s3:::amzn-s3-demo-source-bucket"
     ]
"kms:EncryptionContext:arn:aws:arn": [
     "arn:aws:s3:::amzn-s3-demo-destination-bucket"
     ]
```

Selain izin yang diberikan oleh kebijakan kunci KMS, akun sumber harus menambahkan izin minimum berikut ke kebijakan IAM peran replikasi:

```
{
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
    ],
```

```
    "Resource": [
        "Source-KMS-Key-ARN"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "kms:GenerateDataKey",
        "kms:Encrypt"
    ],
    "Resource": [
        "Destination-KMS-Key-ARN"
    ]
}
```

Untuk informasi selengkapnya tentang cara mereplikasi objek yang dienkripsi AWS KMS, lihat Mereplikasi objek terenkripsi.

- Jika bucket tujuan dimiliki oleh yang lain Akun AWS, verifikasi bahwa pemilik bucket memiliki kebijakan bucket pada bucket tujuan yang memungkinkan pemilik bucket sumber untuk mereplikasi objek. Sebagai contoh, lihat Mengkonfigurasi replikasi untuk bucket di akun yang berbeda.
- Untuk menggunakan Object Lock dengan replikasi, Anda harus memberikan dua izin tambahan pada bucket S3 sumber dalam peran AWS Identity and Access Management (IAM) yang Anda gunakan untuk mengatur replikasi. Dua izin tambahan tersebut adalah `s3:GetObjectRetention` dan`s3:GetObjectLegalHold`. Jika peran tersebut memiliki pernyataan izin `s3:Get*`, pernyataan itu memenuhi persyaratan. Untuk informasi selengkapnya, lihat the section called "Menggunakan Kunci Objek dengan Replikasi S3".
- Jika objek Anda masih tidak bereplikasi setelah Anda memvalidasi izin, periksa pernyataan `Deny` eksplisit di lokasi berikut:
  - Pernyataan `Deny` dalam kebijakan bucket sumber atau tujuan. Replikasi gagal jika kebijakan bucket menghalangi akses ke peran replikasi untuk tindakan-tindakan berikut:

    Bucket sumber:

```
            "s3:GetReplicationConfiguration",
            "s3:ListBucket",
            "s3:GetObjectVersionForReplication",
            "s3:GetObjectVersionAcl",
            "s3:GetObjectVersionTagging"
```

Bucket tujuan:

```
            "s3:ReplicateObject",
            "s3:ReplicateDelete",
            "s3:ReplicateTags"
```

  - Pernyataan atau batasan izin `Deny` yang dilampirkan pada peran IAM dapat menyebabkan replikasi gagal.
  - `Deny`pernyataan dalam kebijakan kontrol AWS Organizations layanan (SCPs) yang dilampirkan ke akun sumber atau tujuan dapat menyebabkan replikasi gagal.
  - `Deny`pernyataan dalam kebijakan kontrol AWS Organizations sumber daya (RCPs) yang dilampirkan pada bucket sumber atau tujuan dapat menyebabkan replikasi gagal.
- Jika replika objek tidak muncul di bucket tujuan, masalah berikut mungkin dapat mencegah replikasi:
  - Amazon S3 tidak mereplikasi objek dalam bucket sumber yang merupakan replika yang dibuat oleh konfigurasi replikasi lain. Misalnya, jika Anda mengatur konfigurasi replikasi dari bucket A ke bucket B ke bucket C, Amazon S3 tidak mereplikasi replika objek di bucket B ke bucket C.
  - Pemilik bucket sumber dapat memberikan Akun AWS izin lain untuk mengunggah objek. Secara default, pemilik bucket sumber tidak memiliki izin untuk objek yang dibuat oleh akun lain. Konfigurasi replikasi hanya mereplikasi objek yang izin aksesnya dimiliki pemilik bucket sumber. Untuk menghindari masalah ini, pemilik bucket sumber dapat memberikan Akun AWS izin lain untuk membuat objek secara kondisional, yang memerlukan izin akses eksplisit pada objek tersebut. Untuk contoh kebijakan, lihat Berikan izin lintas akun untuk unggah objek sekaligus memastikan bahwa pemilik bucket memiliki kendali penuh.
- Misalkan bahwa dalam konfigurasi replikasi, Anda menambahkan aturan untuk mereplikasi subset objek yang memiliki tag tertentu. Dalam kasus ini, Anda harus menetapkan kunci dan nilai tag spesifik pada saat objek dibuat agar Amazon S3 mereplikasi objek. Jika Anda membuat objek terlebih dahulu lalu menambahkan tag ke objek yang sudah ada, Amazon S3 tidak akan mereplikasi objek.
- Gunakan Pemberitahuan Acara Amazon S3 untuk memberi tahu Anda tentang kejadian saat objek tidak mereplikasi ke tujuannya. Wilayah AWS Pemberitahuan Acara Amazon S3 tersedia melalui Amazon Simple Queue Service (Amazon SQS), Amazon Simple Notification Service (Amazon SNS), atau. AWS Lambda Untuk informasi selengkapnya, lihat Menerima peristiwa kegagalan replikasi dengan Notifikasi Peristiwa Amazon S3.

Anda juga dapat melihat alasan kegagalan replikasi dengan menggunakan Notifikasi Peristiwa Amazon S3. Untuk meninjau daftar alasan kegagalan, lihat Alasan kegagalan replikasi Amazon S3.

Kesalahan Replikasi Batch

Untuk memecahkan masalah objek yang tidak bereplikasi ke bucket tujuan, periksa berbagai jenis izin untuk bucket, peran replikasi, dan peran IAM yang digunakan untuk membuat pekerjaan Replikasi Batch. Selain itu, pastikan untuk memeriksa pengaturan Blokir Akses Publik dan pengaturan Kepemilikan Objek S3 untuk bucket Anda.

Untuk tips pemecahan masalah tambahan untuk bekerja dengan Operasi Batch, lihat. the section called “Pemecahan Masalah Operasi Batch”

Jika Anda telah menyiapkan replikasi dan objek tidak bereplikasi, lihat Mengapa objek Amazon S3 saya tidak bereplikasi saat saya mengatur replikasi di antara bucket saya? di pusat AWS re:Post pengetahuan.

Saat menggunakan Replikasi Batch, Anda mungkin akan menjumpai salah satu kesalahan berikut:

- Pembuatan manifes tidak menemukan kunci yang cocok dengan kriteria filter.

  Kesalahan ini terjadi karena salah satu alasan berikut:

  - Saat objek di bucket sumber disimpan di kelas penyimpanan S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive.

    Untuk menggunakan Replikasi Batch pada objek ini, pertama-tama kembalikan ke kelas penyimpanan Standar S3 dengan menggunakan operasi Restore (`S3InitiateRestoreObjectOperation`) dalam pekerjaan Operasi Batch. Untuk informasi selengkapnya, lihat Memulihkan objek yang diarsipkan dan Memulihkan objek (Operasi Batch). Setelah Anda memulihkan objek, Anda dapat mereplikasi mereka dengan menggunakan pekerjaan Replikasi Batch.
  - Jika kriteria filter yang diberikan tidak cocok dengan objek yang valid di bucket sumber.

    Verifikasi dan perbaiki kriteria filter. Misalnya, dalam aturan Replikasi Batch, kriteria filter mencari semua objek di bucket sumber dengan awalan`Tax/`. Jika nama awalan dimasukkan secara tidak akurat, dengan garis miring di awal dan akhir, `/Tax/` bukan hanya di akhir, maka tidak ada objek S3 yang ditemukan. Untuk mengatasi kesalahan, perbaiki awalan, dalam hal ini, dari `/Tax/` ke `Tax/` dalam aturan replikasi.

- Status operasi batch gagal karena alasan: Laporan pekerjaan tidak dapat ditulis ke bucket laporan Anda.

  Kesalahan ini terjadi jika peran IAM yang digunakan untuk pekerjaan Operasi Batch tidak dapat menempatkan laporan penyelesaian ke lokasi yang ditentukan saat Anda membuat pekerjaan. Untuk mengatasi kesalahan ini, periksa apakah peran IAM memiliki `s3:PutObject` izin untuk bucket tempat Anda ingin menyimpan laporan penyelesaian Operasi Batch. Sebaiknya kirimkan laporan ke ember yang berbeda dari ember sumber.

  Untuk tips tambahan tentang mengatasi kesalahan ini, lihatthe section called “Laporan pekerjaan tidak terkirim saat ada masalah izin atau mode retensi diaktifkan”.

- Operasi Batch selesai dengan kegagalan dan Total gagal bukan 0.

  Kesalahan ini terjadi jika terdapat masalah izin objek yang tidak memadai dengan pekerjaan Replikasi Batch yang sedang berjalan. Jika Anda menggunakan aturan replikasi untuk pekerjaan Replikasi Batch, pastikan bahwa peran IAM yang digunakan untuk replikasi memiliki izin yang tepat untuk mengakses objek baik dari keranjang sumber atau tujuan. Anda juga dapat memeriksa Laporan penyelesaian Replikasi Batch untuk meninjau alasan kegagalan replikasi Amazon S3 tertentu.

- Pekerjaan batch berhasil berjalan tetapi jumlah objek yang diharapkan di bucket tujuan tidak sama.

  Kesalahan ini terjadi ketika ada ketidakcocokan antara objek yang tercantum dalam manifes yang disediakan dalam tugas Replikasi Batch dan filter yang Anda pilih saat membuat pekerjaan. Anda mungkin juga menerima pesan ini jika objek di bucket sumber Anda tidak cocok dengan aturan replikasi apa pun dan tidak disertakan dalam manifes yang dihasilkan.

Kegagalan Operasi Batch terjadi setelah menambahkan aturan replikasi baru ke konfigurasi replikasi yang ada

Operasi Batch mencoba melakukan replikasi objek yang ada untuk setiap aturan dalam konfigurasi replikasi bucket sumber. Jika ada masalah dengan salah satu aturan replikasi yang ada, kegagalan dapat terjadi.

Laporan penyelesaian pekerjaan Operasi Batch menjelaskan alasan kegagalan pekerjaan. Untuk daftar kesalahan umum, lihat Penyebab kegagalan replikasi Amazon S3.

Memantau replikasi dengan metrik, pemberitahuan acara, dan status

Anda dapat memantau konfigurasi replikasi langsung dan pekerjaan Replikasi Batch S3 Anda melalui mekanisme berikut:

- Metrik Replikasi S3 — Saat Anda mengaktifkan metrik Replikasi S3, CloudWatch Amazon memancarkan metrik yang dapat Anda gunakan untuk melacak byte yang tertunda, operasi tertunda, dan latensi replikasi pada tingkat aturan replikasi. Anda dapat melihat metrik Replikasi S3 melalui konsol Amazon S3 dan konsol Amazon. CloudWatch Di konsol Amazon S3, Anda dapat melihat metrik ini di tab Metrik bucket sumber. Untuk informasi selengkapnya tentang metrik Replikasi S3, lihat. the section called “Menggunakan metrik Replikasi S3”
- Metrik Lensa Penyimpanan S3 — Selain metrik Replikasi S3, Anda dapat menggunakan metrik Perlindungan Data terkait replikasi yang disediakan oleh dasbor Lensa Penyimpanan S3. Misalnya, jika Anda menggunakan metrik gratis di Lensa Penyimpanan S3, Anda dapat melihat metrik seperti jumlah total byte yang direplikasi dari bucket sumber atau jumlah objek yang direplikasi dari bucket sumber.

  Untuk mengaudit posisi replikasi Anda secara keseluruhan, Anda dapat mengaktifkan metrik lanjutan di Lensa Penyimpanan S3. Dengan metrik lanjutan di S3 Storage Lens, Anda dapat melihat berapa banyak aturan replikasi yang Anda miliki dari berbagai jenis, termasuk jumlah aturan replikasi dengan tujuan replikasi yang tidak valid.

  Untuk informasi selengkapnya tentang bekerja dengan metrik replikasi di Lensa Penyimpanan S3, lihat. the section called “Melihat metrik replikasi di dasbor Lensa Penyimpanan S3”
- Pemberitahuan Peristiwa S3 — Pemberitahuan Acara S3 dapat memberi tahu Anda di tingkat objek jika objek tidak mereplikasi ke tujuannya Wilayah AWS atau ketika objek tidak direplikasi dalam ambang batas tertentu. Pemberitahuan Acara S3 menyediakan jenis peristiwa replikasi berikut:`s3:Replication:OperationFailedReplication`,, `s3:Replication:OperationMissedThresholds3:Replication:OperationReplicatedAfter` dan. `s3:Replication:OperationNotTracked`

  Acara Amazon S3 tersedia melalui Amazon Simple Queue Service (Amazon SQS), Amazon Simple Notification Service (Amazon SNS), atau. AWS Lambda Untuk informasi selengkapnya, lihat the section called “Menerima peristiwa kegagalan replikasi”.
- Nilai status replikasi - Anda juga dapat mengambil status replikasi objek Anda. Status replikasi dapat membantu Anda menentukan keadaan saat ini dari objek yang sedang direplikasi. Status

replikasi dari objek sumber akan kembali sebagai `PENDING`, `COMPLETED`, atau `FAILED`. Status replikasi dari replika akan mengembalikan `REPLICA`.

Anda juga dapat menggunakan nilai status replikasi saat membuat pekerjaan Replikasi Batch S3. Misalnya, Anda dapat menggunakan nilai status ini untuk mereplikasi objek yang belum pernah direplikasi atau replikasi gagal.

Untuk informasi selengkapnya tentang mengambil status replikasi objek Anda, lihat. the section called “Mendapatkan status replikasi” Untuk informasi selengkapnya tentang penggunaan nilai-nilai ini dengan Replikasi Batch, lihatthe section called “Filter untuk tugas Replikasi Batch”.

Topik

- Menggunakan metrik Replikasi S3
- Melihat metrik replikasi di dasbor Lensa Penyimpanan S3
- Menerima peristiwa kegagalan replikasi dengan Notifikasi Peristiwa Amazon S3
- Mendapatkan informasi status replikasi

Menggunakan metrik Replikasi S3

Metrik Replikasi S3 memberikan metrik terperinci untuk aturan replikasi dalam konfigurasi replikasi Anda. Dengan metrik replikasi, Anda dapat memantau minute-by-minute kemajuan dengan melacak byte yang tertunda, operasi tertunda, operasi yang gagal replikasi, dan latensi replikasi.

> **Note**
>
> - Metrik Replikasi S3 ditagih dengan tarif yang sama dengan metrik kustom Amazon. CloudWatch Untuk informasi lebih lanjut, lihat harga Amazon CloudWatch.
> - Jika Anda menggunakan Kontrol Waktu Replikasi S3, Amazon CloudWatch mulai melaporkan metrik replikasi 15 menit setelah Anda mengaktifkan S3 RTC pada aturan replikasi masing-masing.

Metrik Replikasi S3 diaktifkan secara otomatis ketika Anda mengaktifkan Kontrol Waktu Replikasi S3 (S3 RTC). Anda juga dapat mengaktifkan metrik Replikasi S3 secara independen dari S3 RTC saat membuat atau mengedit aturan. S3 RTC mencakup fitur lain seperti perjanjian tingkat layanan (SLA)

dan pemberitahuan atas ambang batas yang terlewat. Untuk informasi selengkapnya, lihat Memenuhi persyaratan kepatuhan dengan Kontrol Waktu Replikasi S3.

Saat metrik Replikasi S3 diaktifkan, Amazon S3 menerbitkan metrik berikut ke Amazon. CloudWatch CloudWatchmetrik disampaikan atas dasar upaya terbaik.

Nama metrik	Deskripsi metrik	Objek apa yang diterapkan metrik ini?	Di Wilayah mana metrik ini diterbitkan?	Apakah metrik ini masih diterbitkan jika bucket tujuan dihapus?	Apakah metrik ini masih diterbitkan jika replikasi tidak terjadi?
Bytes Menunggu Replikasi	Jumlah total byte objek yang menunggu replikasi untuk aturan replikasi yang diberikan.	Metrik ini hanya berlaku untuk objek baru yang direplikasi dengan S3 Cross-Region Replication (S3 CRR) atau S3 Same-Region Replication (S3 SRR).	Metrik ini diterbitkan di Region of the destination bucket.	Tidak	Ya
Latensi Replikasi	Jumlah maksimum detik dimana bucket tujuan replikasi berada di belakang bucket sumber	Metrik ini hanya berlaku untuk objek baru yang direplikasi dengan S3 CRR atau S3 SRR.	Metrik ini diterbitkan di Region of the destination bucket.	Tidak	Ya

Nama metrik	Deskripsi metrik	Objek apa yang diterapkan metrik ini?	Di Wilayah mana metrik ini diterbitkan?	Apakah metrik ini masih diterbitkan jika bucket tujuan dihapus?	Apakah metrik ini masih diterbitkan jika replikasi tidak terjadi?
	untuk aturan replikasi tertentu.				
Operasi Menunggu Replikasi	Jumlah operasi yang menunggu replikasi untuk aturan replikasi tertentu. Metrik ini melacak operasi yang terkait dengan objek, menghapus penanda, tag, daftar kontrol akses (ACLs), dan Kunci Objek S3.	Metrik ini hanya berlaku untuk objek baru yang direplikasi dengan S3 CRR atau S3 SRR.	Metrik ini diterbitkan di Region of the destination bucket.	Tidak	Ya

Nama metrik	Deskripsi metrik	Objek apa yang diterapkan metrik ini?	Di Wilayah mana metrik ini diterbitkan?	Apakah metrik ini masih diterbitkan jika bucket tujuan dihapus?	Apakah metrik ini masih diterbitkan jika replikasi tidak terjadi?
Operasi Gagal Replikasi	Jumlah operasi yang gagal replikasi untuk aturan replikasi tertentu. Metrik ini melacak operasi yang terkait dengan objek, menghapus penanda, tag, daftar kontrol akses (ACLs), dan Kunci Objek. Operasi Replikasi Gagal melacak kegagalan Replikasi S3 yang dikumpulkan pada	Metrik ini berlaku baik untuk objek baru yang direplikasi dengan S3 CRR atau S3 SRR dan juga untuk objek yang ada yang direplikasi dengan S3 Batch Replication. **Note** Jika tugas Replikasi Batch S3 gagal dijalankan sama sekali, metrik	Metrik ini diterbitkan di Region of the source bucket.	Ya	Tidak

Nama metrik	Deskripsi metrik	Objek apa yang diterapkan metrik ini?	Di Wilayah mana metrik ini diterbitkan?	Apakah metrik ini masih diterbitkan jika bucket tujuan dihapus?	Apakah metrik ini masih diterbitkan jika replikasi tidak terjadi?
	interval per menit. Untuk mengidentifikasi objek tertentu yang gagal mereplikasi dan alasan kegagalannya, berlanggan peristiwa `OperationFailedReplication` di Notifikasi Peristiwa Amazon S3. Untuk informasi selengkapnya, lihat Menerima peristiwa kegagalan replikasi dengan Notifikasi	tidak akan dikirim ke Amazon CloudWatch Misalnya, tugas Anda tidak akan berjalan jika Anda tidak memiliki izin yang diperlukan untuk menjalankan tugas Replikasi Batch S3, atau			

Nama metrik	Deskripsi metrik	Objek apa yang diterapkan metrik ini?	Di Wilayah mana metrik ini diterbitkan?	Apakah metrik ini masih diterbitkan jika bucket tujuan dihapus?	Apakah metrik ini masih diterbitkan jika replikasi tidak terjadi?
	Peristiwa Amazon S3.	jika tag atau awalan dalam konfigurasi replikasi tidak cocok.			

Untuk informasi tentang bekerja dengan metrik ini CloudWatch, lihatthe section called “Metrik replikasi S3 di CloudWatch”.

Mengaktifkan metrik Replikasi S3

Anda dapat mulai menggunakan metrik Replikasi S3 dengan aturan replikasi baru atau yang sudah ada. Untuk petunjuk lengkap tentang pembuatan aturan replikasi, lihat Mengkonfigurasi replikasi untuk bucket di akun yang sama. Anda dapat memilih untuk menerapkan aturan replikasi Anda ke seluruh bucket S3, atau ke objek Amazon S3 dengan awalan atau tag tertentu.

Topik ini memberikan petunjuk untuk mengaktifkan metrik Replikasi S3 dalam konfigurasi replikasi Anda saat bucket sumber dan tujuan dimiliki oleh Akun AWS yang sama atau berbeda.

Untuk mengaktifkan metrik replikasi menggunakan AWS Command Line Interface (AWS CLI), Anda harus menambahkan konfigurasi replikasi ke bucket sumber dengan diaktifkan. `Metrics` Dalam konfigurasi contoh ini, objek di bawah awalan *`Tax`* direplikasi ke bucket tujuan *`amzn-s3-demo-bucket`*, dan metrik dibuat untuk objek tersebut.

```
{
    "Rules": [
        {
            "Status": "Enabled",
            "Filter": {
                "Prefix": "Tax"
            },
            "Destination": {
                "Bucket": "arn:aws:s3:::amzn-s3-demo-bucket",
                "Metrics": {
                    "Status": "Enabled"
                }
            },
            "Priority": 1
        }
    ],
    "Role": "IAM-Role-ARN"
}
```

Melihat metrik replikasi

Anda dapat melihat metrik Replikasi S3 di tab Metrik bucket tujuan umum sumber di konsol Amazon S3. CloudWatch Metrik Amazon ini juga tersedia di CloudWatch konsol Amazon. Saat Anda mengaktifkan metrik Replikasi S3, Amazon CloudWatch memancarkan metrik yang dapat Anda gunakan untuk melacak byte yang tertunda, operasi tertunda, dan latensi replikasi pada tingkat aturan replikasi.

Metrik Replikasi S3 diaktifkan secara otomatis saat Anda mengaktifkan replikasi dengan Kontrol Waktu Replikasi S3 (S3 RTC) dengan menggunakan konsol Amazon S3 atau Amazon S3 REST API. Anda juga dapat mengaktifkan metrik Replikasi S3 secara independen dari S3 RTC saat membuat atau mengedit aturan.

Jika Anda menggunakan Kontrol Waktu Replikasi S3, Amazon CloudWatch mulai melaporkan metrik replikasi 15 menit setelah Anda mengaktifkan S3 RTC pada aturan replikasi masing-masing. Untuk informasi selengkapnya, lihat Menggunakan metrik Replikasi S3.

Metrik replikasi melacak aturan konfigurasi IDs replikasi. ID aturan replikasi dapat dikhususkan untuk awalan, tag, atau kombinasi keduanya.

Untuk informasi selengkapnya tentang CloudWatch metrik untuk Amazon S3, lihat. Memantau metrik dengan Amazon CloudWatch

Prasyarat

Buat aturan replikasi yang mengaktifkan metrik Replikasi S3. Untuk informasi selengkapnya, lihat the section called “Mengaktifkan metrik replikasi”.

Untuk melihat metrik Replikasi S3 melalui tab Metrik bucket sumber

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Dalam daftar bucket, pilih nama bucket sumber yang berisi objek yang Anda inginkan untuk metrik replikasi.
4. Pilih tab Metrik.
5. Di bawah Metrik replikasi, pilih aturan replikasi yang ingin Anda lihat metrik.
6. Pilih Tampilkan bagan.

   Amazon S3 menampilkan latensi Replikasi, replikasi tertunda Bytes, replikasi tertunda operasi, dan bagan replikasi gagal operasi untuk aturan yang Anda pilih.

Melihat metrik replikasi di dasbor Lensa Penyimpanan S3

Selain metrik Replikasi S3, Anda dapat menggunakan metrik Perlindungan Data terkait replikasi yang disediakan oleh S3 Storage Lens. Lensa Penyimpanan S3 adalah fitur analisis penyimpanan cloud yang dapat Anda gunakan untuk mendapatkan visibilitas seluruh organisasi ke dalam penggunaan dan aktivitas penyimpanan objek. Untuk informasi selengkapnya, lihat Menggunakan Lensa Penyimpanan S3 untuk melindungi data Anda.

Lensa Penyimpanan S3 memiliki dua tingkatan metrik: metrik gratis, serta metrik dan rekomendasi lanjutan, yang dapat Anda tingkatkan dengan biaya tambahan. Dengan metrik dan rekomendasi lanjutan, Anda dapat mengakses metrik dan fitur tambahan untuk mendapatkan wawasan tentang penyimpanan Anda. Untuk informasi tentang harga Lensa Penyimpanan S3, lihat harga Amazon S3.

Jika Anda menggunakan metrik gratis di Lensa Penyimpanan S3, Anda dapat melihat metrik seperti jumlah total byte yang direplikasi dari bucket sumber atau jumlah objek yang direplikasi dari bucket sumber.

Untuk mengaudit posisi replikasi Anda secara keseluruhan, Anda dapat mengaktifkan metrik lanjutan di Lensa Penyimpanan S3. Dengan metrik lanjutan di S3 Storage Lens, Anda dapat melihat berapa

banyak aturan replikasi yang Anda miliki dari berbagai jenis, termasuk jumlah aturan replikasi dengan tujuan replikasi yang tidak valid.

Untuk daftar lengkap metrik Lensa Penyimpanan S3, termasuk metrik replikasi yang ada di setiap tingkatan, lihat glosarium metrik Lensa Penyimpanan S3.

Prasyarat

Buat konfigurasi replikasi langsung atau pekerjaan Replikasi Batch S3.

Untuk melihat metrik replikasi di Lensa Penyimpanan Amazon S3

1. Buat dasbor Lensa Penyimpanan S3. Untuk step-by-step instruksi, lihatthe section called "Menggunakan konsol S3".
2. (Opsional) Selama pengaturan dasbor, jika Anda ingin melihat semua metrik replikasi Lensa Penyimpanan S3, pilih Metrik dan rekomendasi lanjutan, lalu pilih Metrik perlindungan data lanjutan. Untuk daftar metrik lengkap, lihat glosarium metrik Lensa Penyimpanan S3.

   Jika Anda mengaktifkan metrik dan rekomendasi lanjutan, Anda dapat memperoleh wawasan lebih lanjut tentang konfigurasi replikasi Anda. Misalnya, Anda dapat menggunakan metrik jumlah aturan replikasi Lensa Penyimpanan S3 untuk mendapatkan informasi terperinci tentang bucket yang dikonfigurasi untuk replikasi. Informasi ini mencakup aturan replikasi di dalam dan di seluruh bucket dan Wilayah. Untuk informasi selengkapnya, lihat the section called "Hitung jumlah total aturan replikasi untuk setiap bucket".
3. Setelah Anda membuat dasbor, buka dasbor, dan pilih tab Bucket.
4. Gulir ke bawah ke bagian Bucket. Di bawah Kategori metrik, pilih Perlindungan data. Kemudian hapus Ringkasan.
5. Untuk memfilter daftar Bucket agar hanya menampilkan metrik replikasi, pilih ikon preferensi ().

6. Hapus sakelar untuk semua metrik perlindungan data hingga hanya metrik replikasi yang tetap dipilih.
7. (Opsional) Di bawah Ukuran halaman, pilih jumlah bucket yang akan ditampilkan di daftar.
8. Pilih Lanjutkan.

Menerima peristiwa kegagalan replikasi dengan Notifikasi Peristiwa Amazon S3

Jika Anda telah mengaktifkan metrik Replikasi S3 pada konfigurasi replikasi, Anda dapat mengatur Pemberitahuan Acara Amazon S3 untuk memberi tahu Anda saat objek tidak mereplikasi ke tujuannya. Wilayah AWS Jika Anda telah mengaktifkan Kontrol Waktu Replikasi S3 (S3 RTC) pada konfigurasi replikasi Anda, Anda juga dapat diberi tahu ketika objek tidak bereplikasi dalam ambang batas S3 RTC 15 menit untuk replikasi.

Dengan menggunakan jenis `Replication` peristiwa berikut, Anda dapat memantau minute-by-minute kemajuan peristiwa replikasi dengan melacak byte yang tertunda, operasi tertunda, dan latensi replikasi. Untuk informasi selengkapnya tentang metrik Replikasi S3, lihat. Menggunakan metrik Replikasi S3

- Jenis peristiwa `s3:Replication:OperationFailedReplication` memberi tahu Anda saat objek yang memenuhi syarat untuk replikasi gagal mereplikasi.
- Jenis `s3:Replication:OperationMissedThreshold` acara memberi tahu Anda ketika objek yang memenuhi syarat untuk replikasi yang menggunakan S3 RTC melebihi ambang batas 15 menit untuk replikasi.
- Jenis `s3:Replication:OperationReplicatedAfterThreshold` acara memberi tahu Anda saat objek yang memenuhi syarat untuk replikasi yang menggunakan S3 RTC bereplikasi setelah ambang batas 15 menit.
- Jenis `s3:Replication:OperationNotTracked` acara memberi tahu Anda ketika objek yang memenuhi syarat untuk replikasi langsung (baik Same-Region Replication [SRR] atau Cross-Region Replication [CRR]) tidak lagi dilacak oleh metrik replikasi.

Untuk deskripsi lengkap dari semua jenis peristiwa replikasi yang didukung, lihat. the section called “Jenis event yang didukung untuk SQS, SNS, dan Lambda”

Untuk daftar kode kegagalan yang ditangkap oleh Pemberitahuan Acara S3, lihatPenyebab kegagalan replikasi Amazon S3.

Anda dapat menerima Notifikasi Peristiwa S3 melalui Amazon Simple Queue Service (Amazon SQS), Amazon Simple Notification Service (Amazon SNS), atau AWS Lambda. Untuk informasi selengkapnya, lihat the section called “Notifikasi Peristiwa Amazon S3”.

Untuk petunjuk tentang cara mengonfigurasi Notifikasi Peristiwa Amazon S3, lihat Mengaktifkan pemberitahuan peristiwa.

> **Note**
>
> Selain mengaktifkan pemberitahuan acara, pastikan Anda juga mengaktifkan metrik Replikasi S3. Untuk informasi selengkapnya, lihat the section called “Mengaktifkan metrik replikasi”.

Berikut ini adalah contoh pesan yang dikirimkan Amazon S3 untuk menerbitkan peristiwa `s3:Replication:OperationFailedReplication`. Untuk informasi selengkapnya, lihat the section called “Struktur pesan peristiwa”.


```
{
  "Records": [
    {
      "eventVersion": "2.2",
      "eventSource": "aws:s3",
      "awsRegion": "us-east-1",
      "eventTime": "2024-09-05T21:04:32.527Z",
      "eventName": "Replication:OperationFailedReplication",
      "userIdentity": {
        "principalId": "s3.amazonaws.com"
      },
      "requestParameters": {
        "sourceIPAddress": "s3.amazonaws.com"
      },
      "responseElements": {
        "x-amz-request-id": "123bf045-2b4b-4ca8-a211-c34a63c59426",
        "x-amz-id-2":
 "12VAWNDIHnwJsRhTccqQTeAPoXQmRt22KkewMV8G3XZihAuf9CLDdmkApgZzudaIe2KlLfDqGS0="
      },
      "s3": {
        "s3SchemaVersion": "1.0",
        "configurationId": "ReplicationEventName",
        "bucket": {
          "name": "amzn-s3-demo-bucket1",
          "ownerIdentity": {
            "principalId": "111122223333"
          },
          "arn": "arn:aws:s3:::amzn-s3-demo-bucket1"
        },
        "object": {
          "key": "replication-object-put-test.png",
          "size": 520080,
```

```
          "eTag": "e12345ca7e88a38428305d3ff7fcb99f",
          "versionId": "abcdeH0Xp66ep__QDjR76LK7Gc9X4wKO",
          "sequencer": "0066DA1CBF104C0D51"
        }
      },
      "replicationEventData": {
        "replicationRuleId": "notification-test-replication-rule",
        "destinationBucket": "arn:aws:s3:::amzn-s3-demo-bucket2",
        "s3Operation": "OBJECT_PUT",
        "requestTime": "2024-09-05T21:03:59.168Z",
        "failureReason": "AssumeRoleNotPermitted"
      }
    }
  ]
}
```

Penyebab kegagalan replikasi Amazon S3

Tabel berikut mencantumkan alasan kegagalan replikasi Amazon S3. Anda dapat melihat alasan ini dengan menerima `s3:Replication:OperationFailedReplication` acara dengan Pemberitahuan Acara Amazon S3 dan kemudian melihat nilainya. `failureReason`

Anda juga dapat melihat alasan kegagalan ini dalam laporan penyelesaian Replikasi Batch S3. Untuk informasi selengkapnya, lihat Laporan penyelesaian Replikasi Batch.

Penyebab kegagalan replikasi	Deskripsi
`AssumeRoleNotPermitted`	Amazon S3 tidak dapat mengasumsikan peran AWS Identity and Access Managemen t (IAM) yang ditentukan dalam konfigurasi replikasi atau dalam tugas Operasi Batch.
`DstBucketInvalidRegion`	Bucket tujuan tidak Wilayah AWS sama dengan yang ditentukan oleh pekerjaan Operasi Batch. Kesalahan ini khusus untuk Replikasi Batch.
`DstBucketNotFound`	Amazon S3 tidak dapat menemukan bucket tujuan yang ditentukan dalam konfigurasi replikasi.

Penyebab kegagalan replikasi	Deskripsi
`DstBucketObjectLockConfigMissing`	Untuk mereplikasi objek dari bucket sumber dengan dukungan Kunci Objek, bucket tujuan juga harus memiliki Kunci Objek yang diaktifkan. Kesalahan ini menunjukkan bahwa Kunci Objek mungkin tidak diaktifkan di bucket tujuan. Untuk informasi selengkapnya, lihat Pertimbangan Kunci Objek.
`DstBucketUnversioned`	Penentuan Versi tidak diaktifkan untuk bucket tujuan S3. Untuk mereplikasi objek dengan Replikasi S3, aktifkan Penentuan Versi untuk bucket tujuan.
`DstDelObjNotPermitted`	Amazon S3 tidak dapat mereplikasi penanda hapus ke bucket tujuan. Izin `s3:ReplicateDelete` mungkin hilang untuk bucket tujuan.
`DstKmsKeyInvalidState`	Kunci AWS Key Management Service (AWS KMS) untuk bucket tujuan tidak dalam status valid. Tinjau dan aktifkan AWS KMS kunci yang diperlukan. Untuk informasi selengkapnya tentang mengelola AWS KMS kunci, lihat Status AWS KMS kunci kunci di Panduan AWS Key Management Service Pengembang.
`DstKmsKeyNotFound`	AWS KMS Kunci yang dikonfigurasi untuk bucket tujuan dalam konfigurasi replikasi tidak ada.
`DstMultipartCompleteNotPermitted`	Amazon S3 tidak dapat menyelesaikan unggahan objek multibagian di bucket tujuan. Izin `s3:ReplicateObject` mungkin hilang untuk bucket tujuan.

Penyebab kegagalan replikasi	Deskripsi
`DstMultipartInitNotPermitted`	Amazon S3 tidak dapat memulai unggahan objek multibagian ke bucket tujuan. Izin `s3:ReplicateObject` mungkin hilang untuk bucket tujuan.
`DstMultipartUploadNotPermitted`	Amazon S3 tidak dapat mengunggah objek unggahan multibagian ke bucket tujuan. Izin `s3:ReplicateObject` mungkin hilang untuk bucket tujuan.
`DstObjectHardDeleted`	Replikasi Batch S3 tidak mendukung replikasi ulang objek yang dihapus dengan ID versi objek dari bucket tujuan. Kesalahan ini khusus untuk Replikasi Batch.
`DstPutAclNotPermitted`	Amazon S3 tidak dapat mereplikasi daftar kontrol akses objek (ACLs) ke bucket tujuan. Izin `s3:ReplicateObject` mungkin hilang untuk bucket tujuan.
`DstPutLegalHoldNotPermitted`	Amazon S3 tidak dapat menempatkan penahanan hukum Object Lock pada objek tujuan saat mereplikasi objek yang tidak dapat diubah. Izin `s3:PutObjectLegalHold` mungkin hilang untuk bucket tujuan. Untuk informasi selengkapnya, lihat Penahanan legal.

Penyebab kegagalan replikasi	Deskripsi
`DstPutObjectNotPermitted`	Amazon S3 tidak dapat mereplikasi objek ke bucket tujuan. Hal ini dapat terjadi ketika izin (`s3:ReplicateObject` atau `s3:ObjectOwnerOverrideToBucketOwner` izin) yang diperlukan hilang untuk bucket tujuan atau ketika kebijakan AWS KMS kunci tidak mengizinkan peran replikasi bucket sumber untuk menggunakan AWS KMS kunci (`kms:Decrypt` dan `kms:GenerateDataKey*` tindakan) di bucket tujuan.
`DstPutRetentionNotPermitted`	Amazon S3 tidak dapat menempatkan periode retensi pada objek tujuan saat mereplikasi objek yang tidak dapat diubah. Izin `s3:PutObjectRetention` mungkin hilang untuk bucket tujuan.
`DstPutTaggingNotPermitted`	Amazon S3 tidak dapat mereplikasi tag objek ke bucket tujuan. Izin `s3:ReplicateObject` mungkin hilang untuk bucket tujuan.
`DstVersionNotFound`	Amazon S3 tidak dapat menemukan versi objek yang diperlukan di bucket tujuan yang metadatanya perlu direplikasi.
`InitiateReplicationNotPermitted`	Amazon S3 tidak dapat memulai replikasi pada objek. Izin `s3:InitiateReplication` mungkin hilang untuk tugas Operasi Batch. Kesalahan ini khusus untuk Replikasi Batch.
`SrcBucketInvalidRegion`	Bucket sumber tidak Wilayah AWS sama dengan yang ditentukan oleh pekerjaan Operasi Batch. Kesalahan ini khusus untuk Replikasi Batch.

Penyebab kegagalan replikasi	Deskripsi
`SrcBucketNotFound`	Amazon S3 tidak dapat menemukan bucket sumber.
`SrcBucketReplicationConfigMissing`	Amazon S3 tidak dapat menemukan konfigurasi replikasi untuk bucket sumber.
`SrcGetAclNotPermitted`	Amazon S3 tidak dapat mengakses objek di bucket sumber untuk replikasi. Izin `s3:GetObjectVersionAcl` mungkin hilang untuk objek bucket sumber. Objek di bucket sumber harus dimiliki oleh pemilik bucket. Jika ACLs diaktifkan, verifikasi apakah Kepemilikan Objek disetel ke pilihan pemilik Bucket atau penulis Object. Jika Kepemilikan Objek ditetapkan ke Pemilik bucket pilihan, objek bucket sumber harus memiliki ACL `bucket-owner-full-control` agar pemilik bucket menjadi pemilik objek. Akun sumber dapat mengambil kepemilikan semua objek di bucket mereka dengan menyetel Kepemilikan Objek ke pemilik Bucket yang diberlakukan dan ACLs dinonaktifkan.
`SrcGetLegalHoldNotPermitted`	Amazon S3 tidak dapat mengakses informasi penahanan hukum Kunci Objek S3.
`SrcGetObjectNotPermitted`	Amazon S3 tidak dapat mengakses objek di bucket sumber untuk replikasi. Izin `s3:GetObjectVersionForReplication` mungkin hilang untuk bucket sumber.
`SrcGetRetentionNotPermitted`	Amazon S3 tidak dapat mengakses informasi periode retensi Kunci Objek S3.

Penyebab kegagalan replikasi	Deskripsi
`SrcGetTaggingNotPermitted`	Amazon S3 tidak dapat mengakses informasi tag objek dari bucket sumber. Izin `s3:GetObjectVersionTagging` mungkin hilang untuk bucket sumber.
`SrcHeadObjectNotPermitted`	Amazon S3 tidak dapat mengambil metadata objek dari bucket sumber. Izin `s3:GetObjectVersionForReplication` mungkin hilang untuk bucket sumber.
`SrcKeyNotFound`	Amazon S3 tidak dapat menemukan kunci objek sumber yang akan direplikasi. Objek sumber mungkin telah dihapus sebelum replikasi selesai.
`SrcKmsKeyInvalidState`	AWS KMS Kunci untuk bucket sumber tidak dalam status valid. Tinjau dan aktifkan AWS KMS kunci yang diperlukan. Untuk informasi selengkapnya tentang mengelola AWS KMS kunci, lihat Status AWS KMS kunci kunci di Panduan AWS Key Management Service Pengembang.
`SrcObjectNotEligible`	Beberapa objek tidak memenuhi syarat untuk replikasi. Ini mungkin karena kelas penyimpanan objek atau tag objek tidak cocok dengan konfigurasi replikasi.
`SrcObjectNotFound`	Objek sumber tidak ada.
`SrcReplicationNotPending`	Amazon S3 telah mereplikasi objek ini. Objek ini tidak lagi menunggu replikasi.

Penyebab kegagalan replikasi	Deskripsi
`SrcVersionNotFound`	Amazon S3 tidak dapat menemukan versi objek sumber yang akan direplikasi. Versi objek sumber mungkin telah dihapus sebelum replikasi selesai.

Topik terkait

Menyiapkan izin untuk replikasi langsung

Memecahkan masalah replikasi

Mendapatkan informasi status replikasi

Status replikasi dapat membantu Anda menentukan status saat ini dari objek yang sedang direplikasi. Status replikasi dari objek sumber akan kembali sebagai `PENDING`, `COMPLETED`, atau `FAILED`. Status replikasi dari replika akan mengembalikan `REPLICA`.

Anda juga dapat menggunakan nilai status replikasi saat membuat pekerjaan Replikasi Batch S3. Misalnya, Anda dapat menggunakan nilai status ini untuk mereplikasi objek yang belum pernah direplikasi atau replikasi gagal. Untuk informasi selengkapnya tentang penggunaan nilai-nilai ini dengan Replikasi Batch, lihatthe section called “Menggunakan informasi status replikasi dengan pekerjaan Batch Replication”.

Topik

- Gambaran status replikasi
- Status replikasi jika mereplikasi ke beberapa bucket tujuan
- Status replikasi jika sinkronisasi modifikasi replika Amazon S3 diaktifkan
- Menggunakan informasi status replikasi dengan pekerjaan Batch Replication
- Menemukan status replikasi

Gambaran status replikasi

Dalam replikasi, Anda memiliki bucket sumber tempat Anda mengonfigurasi replikasi dan satu atau beberapa bucket tujuan tempat Amazon S3 mereplikasi objek. Saat Anda meminta objek (dengan

menggunakan`GetObject`) atau metadata objek (dengan menggunakan`HeadObject`) dari bucket ini, Amazon S3 mengembalikan header dalam `x-amz-replication-status` respons:

- Saat Anda meminta objek dari bucket sumber, Amazon S3 mengembalikan header `x-amz-replication-status` jika objek di permintaan Anda memenuhi syarat untuk replikasi.

  Misalnya, Anda menetapkan awalan objek `TaxDocs` dalam konfigurasi replikasi Anda untuk memberi tahu Amazon S3 agar hanya mereplikasi objek dengan awalan nama kunci `TaxDocs`. Objek apa pun yang Anda unggah yang memiliki awalan nama kunci ini—misalnya, `TaxDocs/document1.pdf`—akan direplikasi. Untuk permintaan objek dengan awalan nama kunci ini, Amazon S3 mengembalikan header `x-amz-replication-status` dengan salah satu nilai berikut untuk status replikasi objek: `PENDING`, `COMPLETED`, atau `FAILED`.

  > **Note**
  >
  > Jika replikasi objek gagal setelah Anda mengunggah sebuah objek, Anda tidak dapat mencoba ulang replikasi. Anda harus mengunggah objek lagi, atau Anda harus menggunakan Replikasi Batch S3 untuk mereplikasi objek yang gagal. Untuk informasi selengkapnya tentang menggunakan Replikasi Batch, lihatthe section called “Mereplikasi objek yang ada”.
  > Objek bertransisi ke `FAILED` status untuk masalah seperti izin peran replikasi yang hilang, izin AWS Key Management Service (AWS KMS), atau izin bucket. Untuk kegagalan sementara, seperti jika bucket atau Region tidak tersedia, status replikasi tidak bertransisi ke`FAILED`, tetapi tetap ada. `PENDING` Setelah sumber daya kembali online, Amazon S3 melanjutkan mereplikasi objek tersebut.

- Saat Anda meminta objek dari bucket tujuan, jika objek dalam permintaan Anda adalah replika yang dibuat Amazon S3, Amazon S3 mengembalikan header `x-amz-replication-status` dengan nilai `REPLICA`.

> **Note**
>
> Sebelum menghapus objek dari bucket sumber yang mengaktifkan replikasi, periksa status replikasi objek untuk memastikan bahwa objek telah direplikasi.

Jika konfigurasi Siklus Hidup S3 diaktifkan di bucket sumber, Amazon S3 menangguhkan tindakan siklus hidup hingga menandai status objek sebagai salah satu atau. `COMPLETED` `FAILED`

Status replikasi jika mereplikasi ke beberapa bucket tujuan

Saat Anda mereplikasi objek ke beberapa bucket tujuan, header `x-amz-replication-status` bertindak secara berbeda. Header objek sumber mengembalikan nilai `COMPLETED` hanya ketika replikasi berhasil ke semua tujuan. Header tetap pada nilai `PENDING` sampai replikasi telah selesai untuk semua tujuan. Jika satu atau beberapa tujuan gagal replikasi, header mengembalikan `FAILED`.

Status replikasi jika sinkronisasi modifikasi replika Amazon S3 diaktifkan

Saat aturan replikasi Anda mengaktifkan sinkronisasi modifikasi replika Amazon S3, replika dapat melaporkan status selain `REPLICA`. Jika perubahan metadata sedang dalam proses replikasi, header `x-amz-replication-status` mengembalikan `PENDING`. Jika sinkronisasi modifikasi replika gagal mereplikasi metadata, header akan mengembalikan `FAILED`. Jika metadata direplikasi dengan benar, replika mengembalikan header `REPLICA`.

Menggunakan informasi status replikasi dengan pekerjaan Batch Replication

Saat membuat pekerjaan Replikasi Batch, Anda dapat secara opsional menentukan filter tambahan, seperti tanggal pembuatan objek dan status replikasi, untuk mengurangi cakupan pekerjaan.

Anda dapat memfilter objek yang akan mereplikasi berdasarkan nilai `ObjectReplicationStatuses`, dengan memberikan satu atau beberapa nilai berikut:

- `"NONE"`–Menunjukkan bahwa Amazon S3 belum pernah mencoba mereplikasi objek sebelumnya.
- `"FAILED"`— Menunjukkan bahwa Amazon S3 telah mencoba, tetapi gagal, untuk mereplikasi objek sebelumnya.
- `"COMPLETED"`–Menunjukkan bahwa Amazon S3 telah berhasil mereplikasi objek sebelumnya.
- `"REPLICA"`— Menunjukkan bahwa ini adalah objek replika yang telah direplikasi Amazon S3 dari sumber lain.

Untuk informasi selengkapnya tentang penggunaan nilai status replikasi ini dengan Replikasi Batch, lihat. the section called “Filter untuk tugas Replikasi Batch”

Menemukan status replikasi

Untuk mendapatkan status replikasi dari objek di dalam bucket, Anda dapat menggunakan alat Inventaris Amazon S3. Amazon S3 mengirim file CSV ke bucket tujuan yang Anda tentukan dalam konfigurasi persediaan. Anda juga dapat menggunakan Amazon Athena untuk memerika status replikasi dalam laporan persediaan. Untuk informasi selengkapnya tentang Inventaris Amazon S3, lihat Membuat katalog dan menganalisis data Anda dengan S3 Inventory.

Anda juga dapat menemukan status replikasi objek dengan menggunakan konsol Amazon S3,AWS CLI(), AWS Command Line Interface atau AWS SDK.

Menggunakan konsol S3

Di konsol Amazon S3, Anda dapat melihat status replikasi untuk objek di halaman detail objek.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Dalam daftar Bucket tujuan umum, pilih nama bucket sumber replikasi.
4. Di daftar Objek, pilih nama objek. Halaman detail objek akan muncul.
5. Pada tab Properties, gulir ke bawah ke bagian Ikhtisar manajemen objek. Di bawah Konfigurasi manajemen, lihat nilai di bawah Status replikasi.

Menggunakan AWS CLI

Gunakan `head-object` perintah AWS Command Line Interface (AWS CLI) untuk mengambil metadata objek, seperti yang ditunjukkan pada contoh berikut. Ganti *`amzn-s3-demo-source-bucket1`* dengan nama bucket sumber replikasi Anda, dan ganti yang lain *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3api head-object --bucket amzn-s3-demo-source-bucket1 --key object-key --version-id object-version-id
```

Perintah mengembalikan metadata objek, termasuk `ReplicationStatus` seperti yang ditunjukkan dalam contoh tanggapan berikut.

```
{
```

```
   "AcceptRanges":"bytes",
   "ContentType":"image/jpeg",
   "LastModified":"Mon, 23 Mar 2015 21:02:29 GMT",
   "ContentLength":3191,
   "ReplicationStatus":"COMPLETED",
   "VersionId":"jfnW.HIMOfYiD_9rGbSkmroXsFj3fqZ.",
   "ETag":"\"6805f2cfc46c0f04559748bb039d69ae\"",
   "Metadata":{

   }
}
```

Menggunakan AWS SDKs

Fragmen kode berikut mendapatkan status replikasi Anda dengan menggunakan AWS SDK untuk Java dan AWS SDK untuk .NET, masing-masing.

Java

```
GetObjectMetadataRequest metadataRequest = new GetObjectMetadataRequest(bucketName,
 key);
ObjectMetadata metadata = s3Client.getObjectMetadata(metadataRequest);

System.out.println("Replication Status : " +
 metadata.getRawMetadataValue(Headers.OBJECT_REPLICATION_STATUS));
```

.NET

```
GetObjectMetadataRequest getmetadataRequest = new GetObjectMetadataRequest
    {
         BucketName = sourceBucket,
         Key        = objectKey
    };

GetObjectMetadataResponse getmetadataResponse =
 client.GetObjectMetadata(getmetadataRequest);
Console.WriteLine("Object replication status: {0}",
 getmetadataResponse.ReplicationStatus);
```

Mengelola lalu lintas Multi-wilayah dengan Titik Akses Multi-Wilayah

Poin Akses Multi-Wilayah Amazon S3 menyediakan titik akhir global yang dapat digunakan aplikasi untuk memenuhi permintaan dari bucket S3 yang terletak di beberapa tempat. Wilayah AWS Anda dapat menggunakan Titik Akses Multi-Wilayah untuk membangun aplikasi multi-Wilayah dengan arsitektur yang sama dengan yang digunakan di satu Wilayah, dan kemudian menjalankan aplikasi tersebut di mana saja di seluruh dunia. Alih-alih mengirim permintaan melalui internet publik yang padat, Poin Akses Multi-Region menyediakan ketahanan jaringan bawaan dengan percepatan permintaan berbasis internet ke Amazon S3. Permintaan aplikasi yang dibuat ke titik akhir global Titik Akses Multi-Wilayah digunakan AWS Global Acceleratoruntuk secara otomatis merutekan jaringan AWS global ke bucket S3 kedekatan terdekat dengan status perutean aktif.

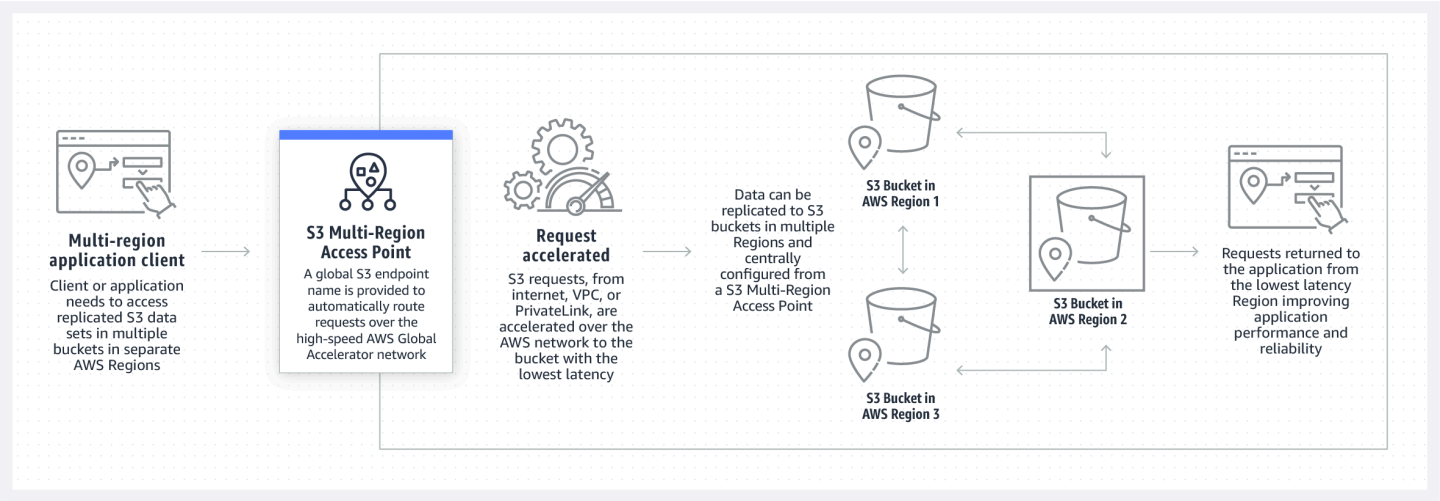
Jika terjadi gangguan lalu lintas Regional, Anda dapat menggunakan kontrol failover Titik Akses Multi-Wilayah untuk mengalihkan lalu lintas permintaan data S3 antara Wilayah AWS dan mengalihkan lalu lintas S3 dari gangguan dalam beberapa menit. Anda juga dapat menguji ketahanan aplikasi terhadap gangguan untuk melakukan failover aplikasi dan melakukan simulasi pemulihan bencana. Jika Anda perlu menghubungkan dan mempercepat permintaan ke S3 dari luar VPC, Anda dapat menyederhanakan aplikasi dan arsitektur jaringan dengan Amazon S3 Multi-Region Access Points. Permintaan Titik Akses Multi-Wilayah Anda akan dialihkan melalui jaringan AWS global dan kemudian kembali ke S3 di dalam Wilayah AWS, tanpa harus melintasi internet publik. Hasilnya, Anda dapat membangun lebih banyak aplikasi yang tersedia.

Selama pembuatan dan penyiapan Titik Akses Multi-Wilayah, Anda akan menentukan satu set Wilayah AWS tempat Anda ingin menyimpan data yang akan dilayani melalui Titik Akses Multi-Wilayah tersebut. Anda dapat menggunakan nama titik akhir Titik Akses Multi-Wilayah yang disediakan untuk menghubungkan klien Anda. Setelah membuat koneksi klien, Anda dapat memilih bucket yang ada atau baru yang ingin Anda rutekan antara permintaan Titik Akses Multi-Wilayah. Kemudian, gunakan aturan S3 Cross-Region Replication (CRR) untuk menyinkronkan data antar bucket di Wilayah tersebut.

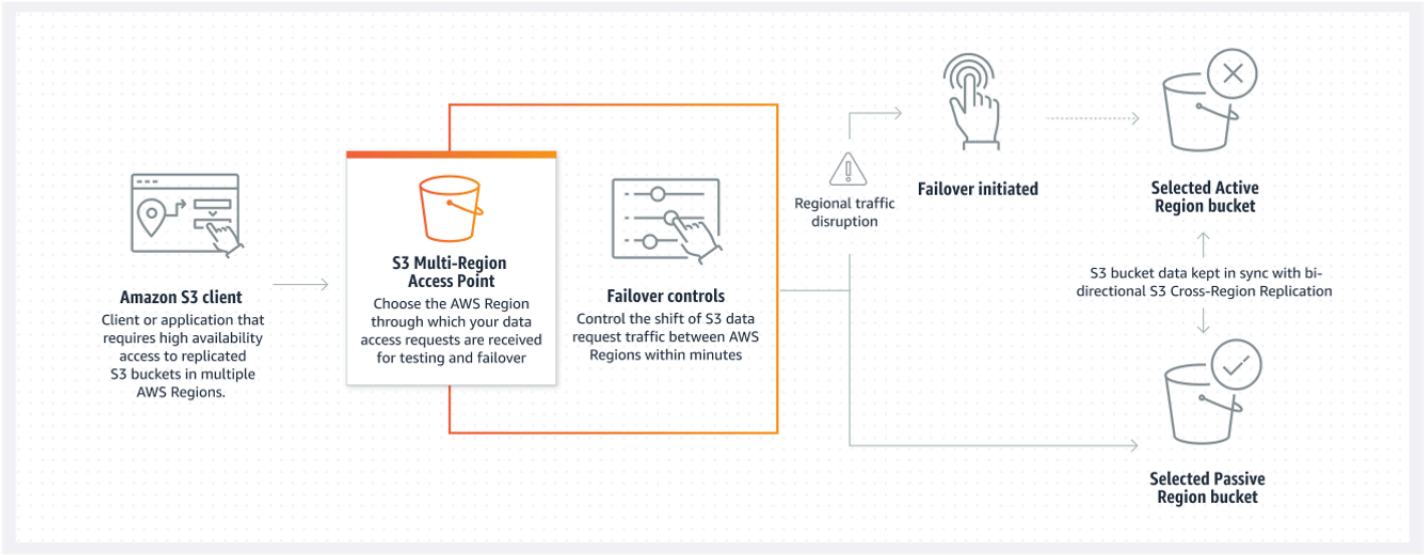
Setelah menyiapkan Titik Akses Multi-Wilayah. Anda kemudian dapat meminta atau menulis data melalui titik akhir global Titik Akses Multi-Wilayah. Amazon S3 secara otomatis menyajikan permintaan ke kumpulan data yang direplikasi dari Wilayah terdekat yang tersedia. Di dalamnya AWS Management Console, Anda juga dapat melihat topologi replikasi yang mendasari dan metrik replikasi yang terkait dengan permintaan Titik Akses Multi-Wilayah Anda. Ini memberi Anda cara yang lebih mudah untuk membangun, mengelola, dan memantau penyimpanan untuk aplikasi Multi-

wilayah. Atau, Anda dapat menggunakan Amazon CloudFront untuk mengotomatiskan pembuatan dan konfigurasi S3 Multi-Region Access Points.

Gambar berikut adalah representasi grafis dari Titik Akses Multi-Wilayah Amazon S3 dalam konfigurasi aktif-aktif. Grafik menunjukkan bagaimana permintaan Amazon S3 secara otomatis dialihkan ke bucket di Wilayah AWS aktif terdekat.



Gambar berikut adalah representasi grafis dari Titik Akses Multi-Wilayah Amazon S3 dalam konfigurasi aktif-pasif. Grafik tersebut menunjukkan bagaimana Anda dapat mengontrol lalu lintas akses data Amazon S3 untuk beralih antara Wilayah AWS aktif dan pasif.



Untuk mempelajari selengkapnya tentang cara menggunakan Titik Akses Multi-Wilayah, lihat Tutorial: Memulai dengan Titik Akses Multi-Wilayah Amazon S3.

Topik

- Membuat Titik Akses Multi-Wilayah
- Mengonfigurasi Titik Akses Multi-Wilayah untuk digunakan dengan AWS PrivateLink
- Membuat permintaan melalui Titik Akses Multi-Wilayah

Membuat Titik Akses Multi-Wilayah

Untuk membuat Titik Akses Multi-Wilayah di Amazon S3, lakukan hal berikut:

- Tentukan nama untuk Titik Akses Multi-Wilayah.
- Pilih satu bucket di masing-masing Wilayah AWS yang ingin Anda layani permintaan untuk Titik Akses Multi-Wilayah.
- Konfigurasikan pengaturan Blokir Akses Publik Amazon S3 untuk Titik Akses Multi-Wilayah.

Anda memberikan semua informasi ini dalam permintaan buat, yang diproses Amazon S3 secara asinkron. Amazon S3 menyediakan token yang dapat Anda gunakan untuk memantau status permintaan pembuatan asinkron.

Pastikan Anda menyelesaikan peringatan keamanan, kesalahan, peringatan umum, dan saran dari AWS Identity and Access Management Access Analyzer sebelum Anda menyimpan kebijakan Anda. Penganalisis Akses IAM menjalankan pemeriksaan kebijakan untuk memvalidasi kebijakan Anda terhadap tata bahasa kebijakan IAM dan praktik terbaik. Pemeriksaan ini menghasilkan temuan dan memberikan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang fungsional dan sesuai dengan praktik terbaik keamanan. Untuk mempelajari validasi kebijakan menggunakan IAM Access Analyzer lebih lanjut, lihat Validasi kebijakan IAM Access Analyzer di Panduan Pengguna IAM. Untuk melihat daftar peringatan, kesalahan, dan saran yang ditampilkan oleh Penganalisis Akses IAM, lihat referensi pemeriksaan kebijakan Penganalisis Akses IAM.

Saat Anda menggunakan API, permintaan untuk membuat Titik Akses Multi-Wilayah bersifat asinkron. Saat Anda mengirimkan permintaan untuk membuat Titik Akses Multi-Wilayah, Amazon S3 secara sinkron mengotorisasi permintaan tersebut. Kemudian segera memberikan token yang dapat Anda gunakan untuk melacak kemajuan permintaan pembuatan. Untuk informasi selengkapnya tentang pelacakan permintaan asinkron untuk membuat dan mengelola Titik Akses Multi-Wilayah, lihat Menggunakan Titik Akses Multi-Wilayah dengan operasi API yang didukung.

Setelah membuat Titik Akses Multi-Wilayah, Anda dapat membuat kebijakan kontrol akses untuk itu. Setiap Titik Akses Multi-Wilayah dapat memiliki kebijakan terkait. Kebijakan Titik Akses Multi-Wilayah

adalah kebijakan berbasis sumber daya yang dapat Anda gunakan untuk membatasi penggunaan Titik Akses Multi-Wilayah berdasarkan sumber daya, pengguna, atau kondisi lainnya.

> **Note**
>
> Agar aplikasi atau pengguna dapat mengakses objek melalui Titik Akses Multi-Wilayah, kedua kebijakan berikut ini harus mengizinkan permintaan tersebut:
>
> - Kebijakan akses untuk Titik Akses Multi-Wilayah
> - Kebijakan akses untuk bucket dasar yang berisi objek tersebut
>
> Jika kedua kebijakan tersebut berbeda, maka kebijakan yang lebih ketat yang diutamakan. Untuk menyederhanakan pengelolaan izin untuk Titik Akses Multi-Wilayah, Anda dapat mendelegasikan kontrol akses dari bucket ke Titik Akses Multi-Wilayah. Untuk informasi selengkapnya, lihat the section called “Contoh kebijakan Titik Akses Multi-Wilayah”.

Penggunaan bucket dengan Titik Akses Multi-Wilayah tidak akan mengubah perilaku bucket tersebut ketika diakses melalui nama bucket yang sudah ada atau melalui Amazon Resource Name (ARN). Semua operasi yang ada terhadap bucket akan terus bekerja seperti sebelumnya. Pembatasan yang Anda sertakan dalam kebijakan Titik Akses Multi-Wilayah hanya berlaku untuk permintaan yang dibuat melalui Titik Akses Multi-Wilayah.

Anda dapat memperbarui kebijakan untuk Titik Akses Multi-Wilayah setelah dibuat, tetapi tidak dapat menghapus kebijakan tersebut. Namun, Anda dapat memperbarui kebijakan Titik Akses Multi-Wilayah untuk menolak semua izin.

Topik

- Aturan penamaan Titik Akses Multi-Wilayah Amazon S3
- Aturan dalam memilih bucket untuk Titik Akses Multi-Wilayah Amazon S3
- Membuat Titik Akses Multi-Wilayah Amazon S3
- Memblokir akses publik dengan Titik Akses Multi-Wilayah Amazon S3
- Melihat detail konfigurasi Titik Akses Multi-Wilayah Amazon S3
- Menghapus Titik Akses Multi-Wilayah

Aturan penamaan Titik Akses Multi-Wilayah Amazon S3

Saat membuat Titik Akses Multi-Wilayah, Anda memberinya nama, yang merupakan string yang Anda pilih. Anda tidak dapat mengubah nama Titik Akses Multi-Region setelah dibuat. Nama harus unik dalam Akun AWS Anda, dan harus sesuai dengan persyaratan penamaan yang tercantum di Pembatasan dan batasan Titik Akses Multi-Wilayah. Untuk membantu Anda mengidentifikasi Titik Akses Multi-Wilayah, gunakan nama yang memiliki arti penting bagi Anda, organisasi, atau yang mencerminkan skenario.

Anda menggunakan nama ini saat menginvokasi operasi manajemen Titik Akses Multi-Wilayah, seperti `GetMultiRegionAccessPoint` dan `PutMultiRegionAccessPointPolicy`. Nama ini tidak digunakan untuk mengirim permintaan ke Titik Akses Multi-Wilayah, dan tidak perlu diekspos ke klien yang membuat permintaan dengan menggunakan Titik Akses Multi-Wilayah.

Saat membuat Titik Akses Multi-Wilayah, Amazon S3 secara otomatis memberikan nama alias untuk titik akses tersebut. Alias ini adalah string alfanumerik unik yang diakhiri dengan `.mrap`. Alias digunakan untuk membangun nama host dan Amazon Resource Name (ARN) untuk Titik Akses Multi-Wilayah. Nama yang sepenuhnya memenuhi syarat juga didasarkan pada alias untuk Titik Akses Multi-Wilayah.

Anda tidak dapat menentukan nama Titik Akses Multi-Wilayah dari nama aliasnya, sehingga Anda dapat mengungkapkan nama alias tanpa risiko mengungkap nama, tujuan, atau pemilik Titik Akses Multi-Wilayah. Amazon S3 memilih alias untuk setiap Titik Akses Multi-Wilayah baru, dan alias tidak dapat diubah. Untuk informasi selengkapnya tentang menangani Titik Akses Multi-Wilayah, lihat Membuat permintaan melalui Titik Akses Multi-Wilayah.

Alias Titik Akses Multi-Wilayah bersifat unik sepanjang waktu dan tidak didasarkan pada nama atau konfigurasi Titik Akses Multi-Wilayah. Jika Anda membuat Titik Akses Multi-Wilayah, lalu menghapusnya dan membuat yang lain dengan nama dan konfigurasi yang sama, Titik Akses Multi-Wilayah kedua akan memiliki alias yang berbeda dari yang pertama. Titik Akses Multi-Wilayah baru tidak akan pernah memiliki alias yang sama dengan Titik Akses Multi-Wilayah sebelumnya.

Aturan dalam memilih bucket untuk Titik Akses Multi-Wilayah Amazon S3

Setiap Titik Akses Multi-Wilayah dikaitkan dengan Wilayah tempat Anda ingin memenuhi permintaan. Titik Akses Multi-Wilayah harus dikaitkan dengan hanya satu bucket di masing-masing Wilayah tersebut. Anda menentukan nama setiap bucket dalam permintaan untuk membuat Titik Akses Multi-Wilayah. Bucket yang mendukung Titik Akses Multi-Region bisa sama dengan Akun AWS yang memiliki Titik Akses Multi-Region, atau bisa juga di tempat lain. Akun AWS

Satu bucket dapat digunakan oleh beberapa Titik Akses Multi-Wilayah.

Important

- Anda dapat menentukan bucket yang terkait dengan Titik Akses Multi-Wilayah hanya pada saat Anda membuatnya. Setelah dibuat, Anda tidak dapat menambahkan, memodifikasi, atau menghapus bucket dari konfigurasi Titik Akses Multi-Wilayah. Untuk mengubah bucket, Anda harus menghapus seluruh Titik Akses Multi-Wilayah dan membuat titik akses baru.
- Anda tidak dapat menghapus bucket yang merupakan bagian dari Titik Akses Multi-Wilayah. Jika Anda ingin menghapus bucket yang dilampirkan ke Titik Akses Multi-Wilayah, hapus Titik Akses Multi-Wilyah terlebih dahulu.
- Jika Anda menambahkan bucket yang dimiliki oleh akun lain ke Titik Akses Multi-Wilayah, pemilik bucket juga harus memperbarui kebijakan bucket untuk memberikan izin akses ke Titik Akses Multi-Wilayah. Jika tidak, Titik Akses Multi-Wilayah tidak akan dapat mengambil data dari bucket tersebut. Misalnya kebijakan yang menunjukkan cara memberikan akses tersebut, lihat contoh kebijakan Titik Akses Multi-Wilayah.
- Tidak semua Wilayah mendukung Titik Akses Multi-Wilayah. Untuk melihat daftar Wilayah yang didukung, lihat Pembatasan dan batasan Titik Akses Multi-Wilayah.

Anda dapat membuat aturan replikasi untuk menyinkronkan data antar bucket. Dengan aturan ini, Anda dapat menyalin data secara otomatis dari bucket sumber ke bucket tujuan. Menghubungkan bucket ke Titik Akses Multi-Wilayah tidak akan memengaruhi cara kerja replikasi. Untuk mengonfigurasi replikasi dengan Titik Akses Multi-Wilayah akan dijelaskan di bagian selanjutnya.

Important

Ketika Anda membuat permintaan ke Titik Akses Multi-Wilayah, Titik Akses Multi-Wilayah tidak mengetahui isi data bucket di dalamnya. Oleh karena itu, bucket yang mendapatkan permintaan mungkin tidak berisi data yang diminta. Untuk membuat set data yang konsisten di bucket Amazon S3 yang terkait dengan Titik Akses Multi-Wilayah, sebaiknya Anda mengonfigurasi Replikasi Lintas Wilayah (CRR) S3. Untuk informasi selengkapnya, lihat Mengkonfigurasi replikasi untuk digunakan dengan Titik Akses Multi-Wilayah.

Membuat Titik Akses Multi-Wilayah Amazon S3

Contoh berikut menunjukkan cara membuat Titik Akses Multi-Wilayah dengan menggunakan konsol Amazon S3.

Menggunakan konsol S3

Untuk membuat Titik Akses Multi-Wilayah

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Titik Akses Multi-Wilayah.
3. Pilih Buat Titik Akses Multi-Wilayah untuk mulai membuat Titik Akses Multi-Wilayah Anda.
4. Pada halaman Titik Akses Multi-Wilayah, berikan nama untuk Titik Akses Multi-Wilayah di bidang nama Titik Akses Multi-Wilayah.
5. Pilih bucket yang akan dikaitkan dengan Titik Akses Multi-Wilayah ini. Anda dapat memilih bucket yang ada di akun Anda, atau pilih bucket dari akun lain.

   > **Note**
   >
   > Anda harus menambahkan setidaknya satu bucket dari akun Anda atau akun lain. Selain itu, ketahuilah bahwa Titik Akses Multi-Wilayah hanya mendukung satu bucket per Wilayah AWS. Oleh karena itu, Anda tidak dapat menambahkan dua bucket dari Wilayah yang sama. Wilayah AWS yang dinonaktifkan secara default tidak didukung.

   - Untuk menambahkan bucket di akun Anda, pilih Tambahkan bucket. Daftar semua bucket di akun Anda akan ditampilkan. Anda dapat mencari bucket berdasarkan nama, atau mengurutkan nama bucket berdasarkan urutan abjad.
   - Untuk menambahkan bucket dari akun lain, pilih Tambahkan bucket dari akun lain. Pastikan Anda mengetahui nama bucket dan Akun AWS ID yang tepat karena Anda tidak dapat mencari atau menelusuri bucket di akun lain.

     > **Note**
     >
     > Anda harus memasukkan Akun AWS ID dan nama bucket yang valid. Bucket juga harus berada di Wilayah yang didukung, jika tidak Anda akan mengalami kesalahan saat mencoba membuat Titik Akses Multi-Wilayah. Untuk daftar Wilayah yang

mendukung Titik Akses Multi-Wilayah, lihat Pembatasan dan batasan Titik Akses Multi-Wilayah.

6. (Opsional) Jika Anda ingin menghapus bucket yang telah Anda tambahkan, pilih Hapus.

> **Note**
>
> Anda tidak dapat menambahkan atau menghapus bucket ke Titik Akses Multi-Wilayah ini setelah Anda selesai membuatnya.

7. Di Bawah Pengaturan Blokir Akses Publik untuk Titik Akses Multi-Wilayah, pilih pengaturan Blokir Akses Publik yang ingin Anda terapkan ke Titik Akses Multi-Wilayah. Secara default, semua pengaturan Blokir Akses Publik diaktifkan untuk Titik Akses Multi-Wilayah baru. Sebaiknya aktifkan semua pengaturan kecuali jika Anda tahu bahwa ada tujuan khusus untuk menonaktifkannya.

> **Note**
>
> Anda tidak dapat mengubah pengaturan Blokir Akses Publik untuk Titik Akses Multi-Wilayah setelah Titik Akses Multi-Wilayah dibuat. Oleh karena itu, jika Anda ingin memblokir akses publik, pastikan aplikasi Anda berfungsi dengan benar tanpa akses publik sebelum Anda membuat Titik Akses Multi-Wilayah.

8. Pilih Buat Titik Akses Multi-Wilayah.

> **Important**
>
> Saat Anda menambahkan bucket milik akun lain ke Titik Akses Multi-Wilayah, pemilik bucket juga harus memperbarui kebijakan bucket untuk memberikan izin akses ke Titik Akses Multi-Wilayah. Jika tidak, Titik Akses Multi-Wilayah tidak akan dapat mengambil data dari bucket tersebut. Misalnya kebijakan yang menunjukkan cara memberikan akses tersebut, lihat contoh kebijakan Titik Akses Multi-Wilayah.

Menggunakan AWS CLI

Anda dapat menggunakan AWS CLI untuk membuat Titik Akses Multi-Region. Saat Anda membuat Titik Akses Multi-Wilayah, Anda harus menyediakan semua bucket yang akan didukungnya. Anda tidak dapat menambahkan bucket ke Titik Akses Multi-Wilayah setelah dibuat.

Berikut ini contoh membuat Titik Akses Multi-Wilayah dengan dua bucket menggunakan perintah AWS CLI. Untuk menggunakan perintah contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3control create-multi-region-access-point --account-id 111122223333 --details '{
        "Name": "simple-multiregionaccesspoint-with-two-regions",
        "PublicAccessBlock": {
            "BlockPublicAcls": true,
            "IgnorePublicAcls": true,
            "BlockPublicPolicy": true,
            "RestrictPublicBuckets": true
        },
        "Regions": [
            { "Bucket": "amzn-s3-demo-bucket1" },
            { "Bucket": "amzn-s3-demo-bucket2" }
        ]
    }' --region us-west-2
```

Memblokir akses publik dengan Titik Akses Multi-Wilayah Amazon S3

Setiap Titik Akses Multi-Wilayah memiliki pengaturan berbeda untuk Blokir Akses Publik Amazon S3. Pengaturan ini beroperasi bersama dengan pengaturan Blokir Akses Publik untuk Akun AWS yang memiliki Titik Akses Multi-Wilayah dan bucket yang mendasarinya.

Ketika mengotorisasi permintaan, Amazon S3 akan menerapkan kombinasi yang paling ketat dari pengaturan ini. Jika pengaturan Blokir Akses Publik untuk salah satu sumber daya ini (akun pemilik Titik Akses Multi-Wilayah, bucket yang mendasari, atau akun pemilik bucket) memblokir akses untuk tindakan atau sumber daya yang diminta, maka Amazon S3 akan menolak permintaan tersebut.

Sebaiknya aktifkan semua pengaturan Blokir Akses Publik kecuali jika ada kebutuhan khusus untuk menonaktifkan salah satu dari pengaturan tersebut. Secara default, semua pengaturan Blokir Akses Publik diaktifkan untuk Titik Akses Multi-Wilayah. Jika Blokir Akses Publik diaktifkan, Titik Akses Multi-Wilayah tidak dapat menerima permintaan berbasis internet.

> **Important**
>
> Anda tidak dapat mengubah pengaturan Blokir Akses Publik untuk Titik Akses Multi-Wilayah setelah dibuat.

Untuk informasi selengkapnya tentang Blokir Akses Publik Amazon S3, lihat Melakukan blok akses publik ke penyimpanan Amazon S3 Anda.

Melihat detail konfigurasi Titik Akses Multi-Wilayah Amazon S3

Berikut ini adalah contoh yang menunjukkan cara melihat detail konfigurasi Titik Akses Multi-Wilayah dengan menggunakan konsol Amazon S3.

Menggunakan konsol S3

Untuk membuat Titik Akses Multi-Wilayah

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Titik Akses Multi-Wilayah.
3. Pilih nama Titik Akses Multi-Wilayah yang ingin Anda lihat detail konfigurasinya.

   - Tab Properti mencantumkan semua bucket yang terkait dengan Titik Akses Multi-Wilayah, tanggal pembuatan, Amazon Resource Name (ARN), dan alias. Kolom ID Akun AWS juga mencantumkan setiap bucket yang dimiliki oleh akun eksternal yang terkait dengan Titik Akses Multi-Wilayah Anda.
   - Tab Izin mencantumkan pengaturan Blokir Akses Publik yang diterapkan ke bucket yang terkait dengan Titik Akses Multi-Wilayah ini. Anda juga dapat melihat kebijakan Titik Akses Multi-Wilayah untuk Titik Akses Multi-Wilayah yang telah anda buat sebelumnya. Notifikasi Info di halaman Izin juga mencantumkan semua bucket (di akun Anda dan akun lain) untuk Titik Akses Multi-Wilayah ini yang mengaktifkan pengaturan Akses Publik diblokir.
   - Tab Replikasi dan failover menyediakan tampilan peta dari bucket yang terkait dengan Titik Akses Multi-Wilayah Anda dan Wilayah tempat bucket berada. Jika ada bucket dari akun lain yang tidak memiliki izin untuk menarik data, Wilayah akan ditandai dengan warna merah di peta Ringkasan replikasi, yang menunjukkan bahwa itu adalah Wilayah AWS dengan kesalahan mendapatkan status replikasi.

> **Note**
>
> Untuk mengambil informasi status replikasi dari bucket di akun eksternal, pemilik bucket harus memberi Anda izin `s3:GetBucketReplication` dalam kebijakan bucket mereka.

Tab ini juga menyediakan metrik replikasi, aturan replikasi, dan status failover untuk Wilayah yang digunakan dengan Titik Akses Multi-Wilayah Anda.

Menggunakan AWS CLI

Anda dapat menggunakan AWS CLI untuk melihat detail konfigurasi untuk Titik Akses Multi-Wilayah.

AWS CLI Contoh berikut mendapatkan konfigurasi Titik Akses Multi-Region Anda saat ini. Untuk menggunakan contoh perintah ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3control get-multi-region-access-point --account-id 111122223333 --name amzn-s3-demo-bucket
```

Menghapus Titik Akses Multi-Wilayah

Prosedur berikut menjelaskan cara menghapus Titik Akses Multi-Wilayah dengan menggunakan konsol Amazon S3.

Menghapus Titik Akses Multi-Wilayah tidak akan menghapus bucket yang terkait dengan Titik Akses Multi-Wilayah, tetapi hanya Titik Akses Multi-Wilayah itu sendiri.

Menggunakan konsol S3

Untuk menghapus Titik Akses Multi-Wilayah

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Titik Akses Multi-Wilayah.
3. Pilih tombol opsi di samping nama Titik Akses Multi-Wilayah Anda.

4. Pilih Hapus.
5. Di kotak dialog Hapus Titik Akses Multi-Wilayah, masukkan nama AWS bucket yang ingin Anda hapus.

> **Note**
>
> Pastikan untuk memasukkan nama bucket yang valid. Jika tidak, tombol Hapus akan dinonaktifkan.

6. Pilih Hapus untuk mengonfirmasi penghapusan Titik Akses Multi-Wilayah Anda.

Menggunakan AWS CLI

Anda dapat menggunakan AWS CLI untuk menghapus Titik Akses Multi-Region. Tindakan ini tidak menghapus bucket yang terkait dengan Titik Akses Multi-Wilayah, tetapi hanya Titik Akses Multi-Wilayah itu sendiri. Untuk menggunakan contoh perintah ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3control delete-multi-region-access-point --account-id 123456789012 --details
 Name=example-multi-region-access-point-name
```

Mengonfigurasi Titik Akses Multi-Wilayah untuk digunakan dengan AWS PrivateLink

Anda dapat menggunakan Titik Akses Multi-Wilayah untuk merutekan lalu lintas permintaan Amazon S3 di antara Wilayah AWS. Setiap titik akhir global Titik Akses Multi-Wilayah merutekan lalu lintas permintaan data Amazon S3 dari berbagai sumber tanpa memerlukan konfigurasi jaringan yang rumit menggunakan titik akhir terpisah. Sumber lalu lintas permintaan data ini meliputi:

- Lalu lintas yang berasal dari cloud privat virtual (VPC)
- Lalu lintas dari pusat data lokal yang bepergian AWS PrivateLink
- Lalu lintas dari internet publik

Jika Anda membuat AWS PrivateLink koneksi ke S3 Multi-Region Access Point, Anda dapat merutekan permintaan S3 ke AWS, atau di beberapa Wilayah AWS, melalui koneksi pribadi dengan menggunakan arsitektur dan konfigurasi jaringan sederhana. Saat Anda menggunakan AWS PrivateLink, Anda tidak perlu mengonfigurasi koneksi peering VPC.

Topik

- Mengonfigurasi Titik Akses Multi-Wilayah untuk digunakan dengan AWS PrivateLink
- Menghapus akses ke Titik Akses Multi-Wilayah dari titik akhir VPC

Mengonfigurasi Titik Akses Multi-Wilayah untuk digunakan dengan AWS PrivateLink

AWS PrivateLink memberi Anda konektivitas pribadi ke Amazon S3 menggunakan alamat IP pribadi di cloud pribadi virtual (VPC) Anda. Anda dapat menyediakan satu atau beberapa titik akhir antarmuka di dalam VPC agar terhubung ke Titik Akses Multi-Wilayah Amazon S3.

Anda dapat membuat titik akhir com.amazonaws.s3-global.accesspoint untuk Titik Akses Multi-Wilayah melalui,, atau. AWS Management Console AWS CLI AWS SDKs Untuk mempelajari selengkapnya tentang cara mengonfigurasi titik akhir antarmuka untuk Titik Akses Multi-Wilayah, lihat Titik akhir VPC Antarmuka di Panduan Pengguna VPC.

Untuk membuat permintaan ke Titik Akses Multi-Wilayah melalui titik akhir antarmuka, ikuti langkah-langkah berikut untuk mengonfigurasi VPC dan Titik Akses Multi-Wilayah.

Untuk mengkonfigurasi Titik Akses Multi-Region untuk digunakan AWS PrivateLink

1. Buat atau dapatkan titik akhir VPC yang sesuai yang dapat terhubung ke Titik Akses Multi-Wilayah. Untuk informasi selengkapnya tentang membuat titik akhir VPC, lihat Antarmuka titik akhir VPC di Panduan Pengguna VPC.

   > **Important**
   >
   > Pastikan untuk membuat titik akhir com.amazonaws.s3-global.accesspoint. Jenis titik akhir lainnya tidak dapat mengakses Titik Akses Multi-Wilayah.

   Setelah titik akhir VPC ini dibuat, semua permintaan Titik Akses Multi-Wilayah di VPC dirutekan melalui titik akhir ini jika Anda mengaktifkan DNS pribadi untuk titik akhir tersebut. Pengaturan ini diaktifkan secara default.
2. Jika kebijakan Titik Akses Multi-Wilayah tidak mendukung koneksi dari titik akhir VPC, Anda perlu memperbaruinya.
3. Verifikasi bahwa kebijakan bucket individual akan memungkinkan akses ke pengguna Titik Akses Multi-Wilayah.

Ingatlah bahwa Titik Akses Multi-Wilayah bekerja dengan merutekan permintaan ke bucket, bukan dengan memenuhi permintaan itu sendiri. Perlu diingat bahwa pembuat permintaan harus memiliki izin ke Titik Akses Multi-Wilayah dan diizinkan untuk mengakses masing-masing bucket di Titik Akses Multi-Wilayah. Jika tidak, permintaan tersebut mungkin akan dialihkan ke bucket yang tidak memiliki izin untuk memenuhi permintaan tersebut. Titik Akses Multi-Wilayah dan bucket yang terkait dapat dimiliki oleh akun yang sama atau akun lain AWS . Namun, VPCs dari akun yang berbeda dapat menggunakan Titik Akses Multi-Region jika izin dikonfigurasi dengan benar.

Karena itu, kebijakan titik akhir VPC harus mengizinkan akses ke Titik Akses Multi-Wilayah dan ke setiap bucket yang mendasarinya agar Anda dapat memenuhi permintaan. Misalnya, katakanlah Anda memiliki Titik Akses Multi-Wilayah dengan alias `mfzwi23gnjvgw.mrap`. Ini didukung oleh bucket `amzn-s3-demo-bucket1` dan `amzn-s3-demo-bucket2`, semuanya dimiliki oleh akun AWS `123456789012`. Dalam hal ini, kebijakan titik akhir VPC berikut ini akan mengizinkan permintaan `GetObject` dari VPC yang dibuat untuk `mfzwi23gnjvgw.mrap` agar dipenuhi oleh bucket pendukung.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Sid": "Read-buckets-and-MRAP-VPCE-policy",
        "Principal": "*",
        "Action": [
            "s3:GetObject"
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:s3:::amzn-s3-demo-bucket1/*",
            "arn:aws:s3:::amzn-s3-demo-bucket2/*",
            "arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap/object/*"
        ]
    }]
}
```

Seperti yang telah disebutkan sebelumnya, Anda juga harus memastikan bahwa kebijakan Titik Akses Multi-Wilayah dikonfigurasi untuk mendukung akses melalui titik akhir VPC. Anda tidak perlu menentukan titik akhir VPC yang meminta akses. Contoh kebijakan berikut akan memberikan akses ke setiap pemohon yang mencoba menggunakan Titik Akses Multi-Wilayah untuk permintaan `GetObject` tersebut.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Sid": "Open-read-MRAP-policy",
        "Effect": "Allow",
        "Principal": "*",
        "Action": [
            "s3:GetObject"
          ],
        "Resource": "arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap/object/*"
    }]
}
```

Dan tentu saja, masing-masing bucket akan membutuhkan kebijakan untuk memberikan akses dari permintaan yang dikirimkan melalui titik akhir VPC. Contoh kebijakan berikut memberikan akses baca ke setiap pengguna anonim, yang akan mencakup permintaan yang dibuat melalui titik akhir VPC.

```
{
    "Version":"2012-10-17",
   "Statement": [
   {
       "Sid": "Public-read",
       "Effect":"Allow",
       "Principal": "*",
       "Action": "s3:GetObject",
       "Resource": [
           "arn:aws:s3:::amzn-s3-demo-bucket1",
           "arn:aws:s3:::amzn-s3-demo-bucket2/*"]
    }]
}
```

Untuk informasi selengkapnya tentang mengedit kebijakan titik akhir VPC, lihat Mengontrol akses ke layanan dengan titik akhir VPC di Panduan Pengguna VPC.

Menghapus akses ke Titik Akses Multi-Wilayah dari titik akhir VPC

Jika Anda memiliki Titik Akses Multi-Wilayah dan ingin menghapus akses dari titik akhir antarmuka, Anda harus menentukan kebijakan akses baru untuk Titik Akses Multi-Wilayah yang mencegah akses untuk permintaan yang masuk melalui titik akhir VPC. Namun, jika bucket di Titik Akses Multi-Wilayah Anda mendukung permintaan melalui titik akhir VPC, bucket tersebut akan terus mendukung jenis

permintaan ini. Jika ingin menghentikan dukungan tersebut, Anda juga harus memperbarui kebijakan untuk bucket. Dengan menetapkan kebijakan akses baru ke Titik Akses Multi-Wilayah, maka akses akan dibatasi hanya ke Titik Akses Multi-Wilayah, bukan ke bucket yang mendasarinya.

> **Note**
>
> Anda tidak dapat menghapus kebijakan akses untuk Titik Akses Multi-Wilayah. Untuk menghapus akses ke Titik Akses Multi-Wilayah, Anda harus menetapkan kebijakan akses baru dengan akses yang dimodifikasi sesuai keinginan Anda.

Daripada memperbarui kebijakan akses untuk Titik Akses Multi-Wilayah, Anda dapat memperbarui kebijakan bucket untuk mencegah permintaan melalui titik akhir VPC. Dalam hal ini, pengguna masih dapat mengakses Titik Akses Multi-Wilayah melalui titik akhir VPC. Namun, jika permintaan Titik Akses Multi-Wilayah dirutekan ke bucket tempat kebijakan bucket mencegah akses, permintaan tersebut akan mendapatkan pesan kesalahan.

Membuat permintaan melalui Titik Akses Multi-Wilayah

Seperti sumber daya lainnya, Titik Akses Multi-Wilayah Amazon S3 memiliki Nama Sumber Daya Amazon (). ARNs Anda dapat menggunakannya ARNs untuk mengarahkan permintaan ke Titik Akses Multi-Wilayah dengan menggunakan AWS Command Line Interface (AWS CLI) AWS SDKs, atau Amazon S3 API. Anda juga dapat menggunakan ini ARNs untuk mengidentifikasi Titik Akses Multi-Wilayah dalam kebijakan kontrol akses. ARN Titik Akses Multi-Wilayah tidak menyertakan atau mengungkapkan nama Titik Akses Multi-Wilayah. Untuk informasi selengkapnya ARNs, lihat Amazon Resource Names (ARNs) di Referensi Umum AWS.

> **Note**
>
> Alias Titik Akses Multi-Wilayah dan ARN tidak dapat digunakan secara bergantian.

Titik Akses Multi-Region ARNs menggunakan format berikut:

`arn:aws:s3::`*`account-id`*`:accesspoint/`*`MultiRegionAccessPoint_alias`*

Berikut ini adalah beberapa contoh dari Multi-Region Access Point ARNs:

- `arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap`mewakili Titik Akses Multi-Region dengan alias`mfzwi23gnjvgw.mrap`, yang dimiliki oleh. Akun AWS `123456789012`

- `arn:aws:s3::123456789012:accesspoint/*` mewakili semua Titik Akses Multi-Wilayah di bawah akun `123456789012`. ARN ini cocok dengan semua Titik Akses Multi-Wilayah untuk akun `123456789012`, tetapi tidak cocok dengan Titik Akses Amazon S3 Regional mana pun karena ARN tidak menyertakan Wilayah AWS. Sebaliknya, ARN `arn:aws:s3:us-west-2:123456789012:accesspoint/*` cocok dengan semua Titik Akses Amazon S3 Regional di Wilayah `us-west-2` untuk `123456789012` akun, tetapi tidak cocok dengan Titik Akses Multi-Wilayah mana pun.

ARNs untuk objek yang diakses melalui Multi-Region Access Point menggunakan format berikut:

`arn:aws:s3::`*`account_id`*`:accesspoint/`*`MultiRegionAccessPoint_alias`*`//`*`key`*

Seperti halnya Titik Akses Multi-Wilayah ARNs, ARNs untuk objek yang diakses melalui Titik Akses Multi-Region tidak menyertakan file. Wilayah AWS Berikut ini adalah beberapa contoh.

- `arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap//-01` menunjukkan `-01`, yang diakses melalui Titik Akses Multi-Wilayah dengan alias `mfzwi23gnjvgw.mrap`, yang dimiliki oleh akun `123456789012`.
- `arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap//*` menunjukkan semua objek yang dapat diakses melalui Titik Akses Multi-Wilayah dengan alias `mfzwi23gnjvgw.mrap`, di akun `123456789012`.
- `arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap//-01/finance/*` menunjukkan semua objek yang dapat diakses di bawah `-01/finance/` untuk Titik Akses Multi-Wilayah dengan alias `mfzwi23gnjvgw.mrap`, di akun `123456789012`.

Nama host Titik Akses Multi-Wilayah

Anda dapat mengakses data di Amazon S3 melalui Titik Akses Multi-Wilayah dengan menggunakan nama host dari Titik Akses Multi-Wilayah. Permintaan dapat diarahkan ke nama host ini dari internet publik. Jika Anda telah mengonfigurasi satu atau lebih gateway internet untuk Titik Akses Multi-Wilayah, permintaan juga dapat diarahkan ke nama host ini dari cloud privat virtual (VPC). Untuk informasi selengkapnya tentang membuat titik akhir antarmuka VPC yang akan digunakan dengan Titik Akses Multi-Wilayah, lihat Mengonfigurasi Titik Akses Multi-Wilayah untuk digunakan dengan AWS PrivateLink.

Untuk membuat permintaan melalui Titik Akses Multi-Wilayah dari VPC dengan menggunakan titik akhir VPC, Anda dapat menggunakan AWS PrivateLink. Saat Anda membuat permintaan ke

Titik Akses Multi-Wilayah dengan menggunakan AWS PrivateLink, Anda tidak dapat langsung menggunakan nama Sistem Nama Domain Regional (DNS) spesifik titik akhir yang diakhiri dengan. *region*.vpce.amazonaws.com Nama host ini tidak akan memiliki sertifikat yang terkait dengannya, sehingga tidak dapat digunakan secara langsung. Anda masih dapat menggunakan nama SISTEM NAMA DOMAIN (DNS) dari titik akhir VPC sebagai target CNAME atau ALIAS. Atau, Anda dapat mengaktifkan SISTEM NAMA DOMAIN (DNS) pada titik akhir dan menggunakan nama Titik Akses Multi-Wilayah *MultiRegionAccessPoint_alias*.accesspoint.s3-global.amazonaws.com SISTEM NAMA DOMAIN (DNS) standar, seperti yang dijelaskan di bagian ini.

Saat Anda membuat permintaan ke API untuk operasi data Amazon S3 (misalnya, GetObject) melalui Titik Akses Multi-Wilayah, nama host untuk permintaan tersebut adalah sebagai berikut:

MultiRegionAccessPoint_alias.accesspoint.s3-global.amazonaws.com

Misalnya, untuk membuat permintaan GetObject melalui Titik Akses Multi-Wilayah dengan alias mfzwi23gnjvgw.mrap, buat permintaan ke nama host mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com. Bagian s3-global dari nama host menunjukkan bahwa nama host ini bukan untuk Wilayah tertentu.

Membuat permintaan melalui Titik Akses Multi-Wilayah mirip dengan membuat permintaan melalui titik akses satu-Wilayah. Namun demikian, Anda harus mengetahui perbedaan berikut ini:

- Titik Akses Multi-Wilayah ARNs tidak menyertakan file Wilayah AWS. ARN tersebut mengikuti format arn:aws:s3::*account-id*:accesspoint/*MultiRegionAccessPoint_alias*.
- Untuk permintaan yang dibuat melalui operasi API (permintaan ini tidak memerlukan penggunaan ARN), Titik Akses Multi-Wilayah menggunakan skema titik akhir yang berbeda. Skema ini *MultiRegionAccessPoint_alias*.accesspoint.s3-global.amazonaws.com—misalnya, mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com. Perhatikan perbedaannya dibandingkan dengan titik akses satu-Wilayah:
  - Nama host Titik Akses Multi-Wilayah menggunakan aliasnya, bukan nama Titik Akses Multi-Wilayah tersebut.
  - Nama host Titik Akses Multi-Wilayah tidak menyertakan ID pemilik. Akun AWS
  - Nama host Titik Akses Multi-Wilayah tidak menyertakan file Wilayah AWS.
  - Nama host Titik Akses Multi-Wilayah menyertakan s3-global.amazonaws.com dan bukan s3.amazonaws.com.

- Permintaan Titik Akses Multi-Wilayah harus ditandatangani dengan menggunakan Tanda Tangan Versi 4A (SigV4A). Saat Anda menggunakan AWS SDKs, SDK secara otomatis mengonversi SigV4 ke Sigv4a. Oleh karena itu, pastikan SDK AWS Anda mendukung SigV4A sebagai implementasi penandatanganan yang digunakan untuk menandatangani permintaan global Wilayah AWS . Untuk informasi selengkapnya tentang Sigv4a, lihat Menandatangani permintaan AWS API di. Referensi Umum AWS

Titik Akses Multi-Wilayah dan Amazon S3 Transfer Acceleration

Amazon S3 Transfer Acceleration adalah fitur yang memungkinkan transfer data dengan cepat ke bucket. Transfer Acceleration dikonfigurasi pada tingkat bucket individual. Untuk informasi lebih lanjut tentang Transfer Acceleration, lihat Mengonfigurasi transfer file yang cepat dan aman menggunakan Amazon S3 Transfer Acceleration.

Multi-Region Access Points menggunakan mekanisme transfer akselerasi serupa seperti Transfer Acceleration untuk mengirim objek besar melalui AWS jaringan. Karena itu, Anda tidak perlu menggunakan Transfer Acceleration saat mengirim permintaan melalui Titik Akses Multi-Wilayah. Peningkatan kinerja transfer ini secara otomatis dimasukkan ke dalam Titik Akses Multi-Wilayah.

Topik

- Izin
- Pembatasan dan batasan Titik Akses Multi-Wilayah
- Perutean permintaan Titik Akses Multi-Wilayah
- Kontrol failover Titik Akses Multi-Wilayah Amazon S3
- Mengkonfigurasi replikasi untuk digunakan dengan Titik Akses Multi-Wilayah
- Menggunakan Titik Akses Multi-Wilayah dengan operasi API yang didukung
- Memantau dan mencatat permintaan yang dilakukan melalui Titik Akses Multi-Wilayah ke sumber daya yang mendasarinya

Izin

Poin Akses Multi-Wilayah Amazon S3 dapat menyederhanakan akses data untuk bucket Amazon S3 dalam beberapa kali. Wilayah AWS Titik Akses Multi-Wilayah diberi nama titik akhir global yang dapat Anda gunakan untuk melakukan operasi objek akses data Amazon S3, seperti `GetObject` dan `PutObject`. Setiap Titik Akses Multi-Wilayah dapat memiliki izin dan kendali jaringan yang berbeda untuk setiap permintaan yang dibuat melalui titik akhir global.

Setiap Titik Akses Multi-Wilayah juga dapat memberlakukan kebijakan akses khusus yang bekerja bersama dengan kebijakan bucket yang melekat pada bucket yang mendasarinya. Agar permintaan lintas akun berhasil, kebijakan berikut harus mengizinkan operasi:

- Kebijakan Titik Akses Multi-Wilayah
- Kebijakan yang mendasari AWS Identity and Access Management (IAM)
- Kebijakan bucket yang mendasarinya (tempat permintaan dirutekan)

> **Note**
>
> Untuk permintaan akun yang sama, hanya kebijakan IAM yang mendasari, yang memberikan akses yang sesuai, yang diperlukan.

Anda dapat mengonfigurasi kebijakan Titik Akses Multi-Wilayah untuk menerima permintaan hanya dari pengguna IAM atau grup tertentu. Untuk contoh cara melakukannya, lihat Contoh 2 di the section called “Contoh kebijakan Titik Akses Multi-Wilayah”. Untuk membatasi akses data Amazon S3 ke jaringan privat, Anda dapat mengonfigurasi kebijakan Titik Akses Multi-Wilayah untuk menerima permintaan hanya dari cloud privat virtual (VPC).

Misalnya, Anda membuat `GetObject` permintaan melalui Titik Akses Multi-Wilayah dengan menggunakan pengguna yang dipanggil `AppDataReader` di AWS akun Anda. Untuk memastikan bahwa permintaan tidak akan ditolak, pengguna `AppDataReader` harus diberikan izin `s3:GetObject` oleh Titik Akses Multi-Wilayah dan oleh setiap bucket yang mendasari Titik Akses Multi-Wilayah. `AppDataReader` tidak akan dapat mengambil data dari bucket mana pun yang tidak memberikan izin ini.

> **Important**
>
> Mendelegasikan kontrol akses untuk bucket ke kebijakan Titik Akses Multi-Wilayah tidak mengubah perilaku bucket saat bucket diakses langsung melalui nama bucket atau Amazon Resource Name (ARN). Semua operasi yang dilakukan langsung terhadap bucket akan terus bekerja seperti sebelumnya. Pembatasan yang Anda sertakan dalam kebijakan Titik Akses Multi-Wilayah hanya berlaku untuk permintaan yang dibuat melalui Titik Akses Multi-Wilayah tersebut.

Mengelola akses publik ke Titik Akses Multi-Wilayah

Titik Akses Multi-Wilayah mendukung pengaturan Blokir Akses Publik independen untuk setiap Titik Akses Multi-Wilayah. Saat Anda membuat Titik Akses Multi-Wilayah, Anda dapat menentukan pengaturan Blokir Akses Publik yang berlaku untuk Titik Akses Multi-Wilayah tersebut.

> **Note**
>
> Setelan Blokir Akses Publik apa pun yang diaktifkan di bawah pengaturan Blokir Akses Publik untuk akun ini (di akun Anda sendiri) atau Blokir Pengaturan Publik untuk bucket eksternal tetap berlaku meskipun pengaturan Blokir Akses Publik independen untuk Titik Akses Multi-Wilayah Anda dinonaktifkan.

Untuk setiap permintaan yang dibuat melalui Titik Akses Multi-Wilayah, Amazon S3 mengevaluasi pengaturan Blokir Akses Publik untuk:

- Titik Akses Multi-Wilayah
- Bucket yang mendasarinya (termasuk bucket eksternal)
- Akun yang memiliki Titik Akses Multi-Wilayah
- Akun yang memiliki bucket yang mendasarinya (termasuk akun eksternal)

Jika salah satu pengaturan ini menunjukkan bahwa permintaan tersebut harus diblokir, Amazon S3 menolak permintaan tersebut. Untuk informasi selengkapnya tentang fitur Blokir Akses Publik Amazon S3, lihat Melakukan blok akses publik ke penyimpanan Amazon S3 Anda.

> **Important**
>
> Secara default, semua pengaturan Blokir Akses Publik diaktifkan untuk Titik Akses Multi-Wilayah. Anda harus secara eksplisit menonaktifkan pengaturan apa pun yang tidak ingin Anda terapkan ke Titik Akses Multi-Wilayah.
> Anda tidak dapat mengubah pengaturan Blokir Akses Publik untuk Titik Akses Multi-Wilayah setelah dibuat.

Melihat setelan Blokir Akses Publik untuk Titik Akses Multi-Wilayah

Untuk melihat pengaturan Blokir Akses Publik untuk Titik Akses Multi-Wilayah

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Titik Akses Multi-Wilayah.
3. Pilih nama Titik Akses Multi-Wilayah yang ingin Anda tinjau.
4. Pilih tab Izin.
5. Di bawah pengaturan Blokir Akses Publik untuk Titik Akses Multi-Wilayah ini, tinjau pengaturan Blokir Akses Publik untuk Titik Akses Multi-Wilayah Anda.

> **Note**
>
> Anda tidak dapat mengedit pengaturan Blokir Akses Publik setelah Titik Akses Multi-Wilayah dibuat. Oleh karena itu, jika Anda ingin memblokir akses publik, pastikan aplikasi Anda berfungsi dengan benar tanpa akses publik sebelum Anda membuat Titik Akses Multi-Wilayah.

Menggunakan kebijakan Titik Akses Multi-Wilayah

Contoh kebijakan Titik Akses Multi-Wilayah berikut memberikan akses pengguna IAM untuk membuat daftar dan mengunduh file dari Titik Akses Multi-Wilayah Anda. Untuk menggunakan kebijakan contoh ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
{
  "Version":"2012-10-17",
  "Statement":[
     {
        "Effect":"Allow",
        "Principal":{
           "AWS":"arn:aws:iam::123456789012:user/JohnDoe"
        },
        "Action":[
           "s3:ListBucket",
           "s3:GetObject"
        ],
        "Resource":[
```

```
            "arn:aws:s3::111122223333:accesspoint/MultiRegionAccessPoint_alias",
            "arn:aws:s3::111122223333:accesspoint/MultiRegionAccessPoint_alias/object/
*"
         ]
      }
   ]
}
```

Untuk mengaitkan kebijakan Titik Akses Multi-Wilayah Anda dengan Titik Akses Multi-Wilayah yang telah ditentukan menggunakan AWS Command Line Interface (AWS CLI), gunakan perintah `put-multi-region-access-point-policy` berikut. Untuk menggunakan contoh perintah ini, ganti *`user input placeholders`* dengan informasi Anda sendiri. Setiap Titik Akses Multi-Wilayah hanya dapat memiliki satu kebijakan, sehingga permintaan yang dibuat untuk tindakan `put-multi-region-access-point-policy` tersebut akan menggantikan kebijakan yang ada yang terkait dengan Titik Akses Multi-Wilayah yang telah ditentukan.

AWS CLI

```
aws s3control put-multi-region-access-point-policy
--account-id 111122223333
--details { "Name": "amzn-s3-demo-bucket-MultiRegionAccessPoint",
 "Policy": "{ \"Version\": \"2012-10-17\", \"Statement\": { \"Effect\":
 \"Allow\", \"Principal\": { \"AWS\": \"arn:aws:iam::111122223333:root
\" }, \"Action\": [\"s3:ListBucket\", \"s3:GetObject\"], \"Resource\":
 [ \"arn:aws:s3::111122223333:accesspoint/MultiRegionAccessPoint_alias",
 \"arn:aws:s3::111122223333:accesspoint/MultiRegionAccessPoint_alias/object/*
\" ] } }" }
```

Jika Anda ingin mengetahui hasil dari operasi sebelumnya, gunakan perintah berikut ini:

AWS CLI

```
aws s3control describe-multi-region-access-point-operation
--account-id 111122223333
--request-token-arn requestArn
```

Untuk mengambil kebijakan Titik Akses Multi-Wilayah, gunakan perintah berikut:

AWS CLI

```
aws s3control get-multi-region-access-point-policy
--account-id 111122223333
--name=amzn-s3-demo-bucket-MultiRegionAccessPoint
```

Mengedit kebijakan Titik Akses Multi-Wilayah

Kebijakan Titik Akses Multi-Wilayah (ditulis dalam JSON) menyediakan akses penyimpanan ke bucket Amazon S3 yang digunakan dengan Titik Akses Multi-Wilayah ini. Anda dapat mengizinkan atau menolak pengguna utama tertentu untuk melakukan berbagai tindakan pada Titik Akses Multi-Wilayah. Ketika permintaan dirutekan ke bucket melalui Titik Akses Multi-Wilayah, maka kebijakan akses untuk Titik Akses Multi-Wilayah dan bucket akan diberlakukan. Kebijakan akses yang lebih ketat selalu diutamakan.

> **Note**
>
> Jika bucket berisi objek yang dimiliki oleh akun lain, kebijakan Titik Akses Multi-Wilayah tidak berlaku untuk objek yang dimiliki oleh orang lain Akun AWS.

Setelah Anda menerapkan kebijakan Titik Akses Multi-Wilayah, kebijakan tersebut tidak dapat dihapus. Anda dapat mengedit kebijakan atau membuat kebijakan baru yang menimpa kebijakan yang sudah ada.

Untuk mengedit kebijakan Titik Akses Multi-Wilayah

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Titik Akses Multi-Wilayah.
3. Pilih nama Titik Akses Multi-Wilayah yang ingin Anda edit kebijakannya.
4. Pilih tab Izin.
5. Gulir ke bawah ke bagian kebijakan Titik Akses Multi-Wilayah. Pilih Edit untuk memperbarui kebijakan (di JSON).

6. Halaman kebijakan Edit Titik Akses Multi-Wilayah akan muncul. Anda dapat memasukkan kebijakan langsung ke bidang teks atau memilih Tambahkan pernyataan untuk memilih elemen kebijakan dari daftar dropdown.

> **Note**
>
> Konsol akan secara otomatis menampilkan Titik Akses Multi-Wilayah Amazon Resource Name (ARN), yang dapat Anda gunakan di dalam kebijakan. Misalnya kebijakan Titik Akses Multi-Wilayah, lihat the section called “Contoh kebijakan Titik Akses Multi-Wilayah”.

Contoh kebijakan Titik Akses Multi-Wilayah

Kebijakan sumber daya dukungan AWS Identity and Access Management (IAM) Amazon S3 Multi-Region Access Points (IAM). Anda dapat menggunakan kebijakan ini untuk mengontrol penggunaan Titik Akses Multi-Wilayah berdasarkan sumber daya, pengguna, atau ketentuan lainnya. Agar aplikasi atau pengguna dapat mengakses objek melalui Titik Akses Multi-Wilayah, baik Titik Akses Multi-Wilayah maupun bucket yang mendasarinya harus mengizinkan akses yang sama.

Untuk mengizinkan akses yang sama ke Titik Akses Multi-Wilayah dan bucket yang mendasarinya, lakukan salah satu hal berikut:

- (Disarankan) Untuk menyederhanakan kontrol akses saat menggunakan Titik Akses Multi-Wilayah Amazon S3, delegasikan kontrol akses untuk bucket Amazon S3 ke Titik Akses Multi-Wilayah. Untuk contoh cara melakukannya, lihat Contoh 1 di bagian ini.
- Tambahkan izin yang sama yang terdapat dalam kebijakan Titik Akses Multi-Wilayah ke kebijakan bucket yang mendasarinya.

> **Important**
>
> Mendelegasikan kontrol akses untuk bucket ke kebijakan Titik Akses Multi-Wilayah tidak mengubah perilaku bucket saat bucket diakses langsung melalui nama bucket atau Amazon Resource Name (ARN). Semua operasi yang dilakukan langsung terhadap bucket akan terus bekerja seperti sebelumnya. Pembatasan yang Anda sertakan dalam kebijakan Titik Akses Multi-Wilayah hanya berlaku untuk permintaan yang dibuat melalui Titik Akses Multi-Wilayah tersebut.

Example 1–Mendelegasikan akses ke Titik Akses Multi-Wilayah tertentu dalam kebijakan bucket Anda (untuk akun atau lintas akun yang sama)

Contoh kebijakan bucket berikut memberikan akses bucket penuh ke Titik Akses Multi-Wilayah tertentu. Ini berarti bahwa semua akses ke bucket ini dikontrol oleh kebijakan yang terdapat pada Titik Akses Multi-Wilayah. Kami menyarankan Anda untuk mengonfigurasi bucket dengan cara ini untuk semua kasus penggunaan yang tidak memerlukan akses langsung ke bucket. Anda dapat menggunakan struktur kebijakan bucket ini untuk Titik Akses Multi-Wilayah di akun yang sama atau di akun lain.

```
{
    "Version": "2012-10-17",
    "Statement" : [
    {
        "Effect": "Allow",
        "Principal" : { "AWS": "*" },
        "Action" : "*",
        "Resource" : [ "Bucket ARN", "Bucket ARN/*"],
        "Condition": {
            "StringEquals" : { "s3:DataAccessPointArn" : "MultiRegionAccessPoint_ARN" }
        }
    }]
}
```

> **Note**
>
> Jika ada beberapa Titik Akses Multi-Wilayah yang Anda berikan akses, pastikan untuk mencantumkan setiap Titik Akses Multi-Wilayah.

Example 2–Memberikan akses akun ke Titik Akses Multi-Wilayah dalam kebijakan Titik Akses Multi-Wilayah Anda

Kebijakan Titik Akses Multi-Wilayah berikut memungkinkan izin *123456789012* akun untuk mencantumkan dan membaca objek yang terdapat dalam Titik Akses Multi-Wilayah yang ditentukan oleh *MultiRegionAccessPoint_ARN*.

```
{
  "Version":"2012-10-17",
  "Statement": [
```

```
    {
       "Effect": "Allow",
       "Principal": {
          "AWS":"arn:aws:iam::123456789012:user/JohnDoe"
       },
       "Action":[
          "s3:ListBucket",
          "s3:GetObject"
       ],
       "Resource":[
          "MultiRegionAccessPoint_ARN",
          "MultiRegionAccessPoint_ARN/object/*"
       ]
     }
  ]
}
```

Example 3–Kebijakan Titik Akses Multi-Wilayah yang memungkinkan pembuatan daftar bucket

Kebijakan Titik Akses Multi-Wilayah berikut memungkinkan izin *123456789012* akun untuk mencantumkan objek yang terdapat dalam Titik Akses Multi-Wilayah yang ditentukan oleh *MultiRegionAccessPoint_ARN*.

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
       "Effect": "Allow",
       "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/JohnDoe"
        },
        "Action": "s3:ListBucket",
        "Resource": "MultiRegionAccessPoint_ARN"
       }
    ]
}
```

Pembatasan dan batasan Titik Akses Multi-Wilayah

Titik Akses Multi-Wilayah di Amazon S3 memiliki pembatasan dan batasan berikut.

- Nama Titik Akses Multi-Wilayah:

  - Harus unik dalam satu AWS akun
  - Harus dimulai dengan angka atau huruf kecil
  - Panjang nama harus berkisar antara 3 sampai 50 karakter
  - Tidak dapat diawali atau diakhiri dengan tanda hubung (`-`)
  - Tidak boleh berisi garis bawah (`_`), huruf besar, atau titik (`.`)
  - Tidak dapat diedit setelah dibuat
- Alias Titik Akses Multi-Wilayah dibuat oleh Amazon S3 dan tidak dapat diedit atau digunakan kembali.
- Anda tidak dapat mengakses data melalui Titik Akses Multi-Wilayah dengan menggunakan titik akhir gateway. Namun, Anda dapat mengakses data melalui Titik Akses Multi-Wilayah dengan menggunakan titik akhir antarmuka. Untuk menggunakannya AWS PrivateLink, Anda harus membuat titik akhir VPC. Untuk informasi selengkapnya, lihat Mengonfigurasi Titik Akses Multi-Wilayah untuk digunakan dengan AWS PrivateLink.
- Untuk menggunakan Titik Akses Multi-Wilayah dengan Amazon CloudFront, Anda harus mengonfigurasi Titik Akses Multi-Wilayah sebagai jenis `Custom Origin` distribusi. Untuk informasi selengkapnya tentang berbagai jenis asal, lihat Menggunakan berbagai asal dengan CloudFront distribusi. Untuk informasi selengkapnya tentang penggunaan Titik Akses Multi-Wilayah dengan Amazon CloudFront, lihat Membangun aplikasi berbasis kedekatan aktif aktif di beberapa Wilayah di Blog Penyimpanan.AWS
- Persyaratan minimum Titik Akses Multi-Wilayah:
  - Keamanan Lapisan Pengangkutan (TLS) v1.2

    > **Note**
    >
    > Transport Layer Security (TLS) v1.3 tidak didukung.

  - Tanda Tangan Versi 4 (SigV4A)

    Titik Akses Multi-Wilayah mendukung Tanda Tangan Versi 4A. Versi SigV4 ini memungkinkan permintaan ditandatangani untuk beberapa Wilayah AWS. Fitur ini berguna dalam operasi API yang dapat menghasilkan akses data dari beberapa Wilayah. Saat menggunakan AWS SDK, Anda memberikan kredensyal Anda, dan permintaan ke Titik Akses Multi-Wilayah akan menggunakan Signature Version 4A tanpa konfigurasi tambahan. Pastikan untuk memeriksa kompatibilitas SDK AWS Anda dengan algoritma SigV4A. Untuk informasi selengkapnya tentang Sigv4a, lihat Menandatangani permintaan AWS API di. Referensi Umum AWS

> **Note**
>
> Untuk menggunakan Sigv4a dengan kredensyal keamanan sementara—misalnya, saat menggunakan peran AWS Identity and Access Management (IAM) — Anda dapat meminta kredensyal sementara dari titik akhir Regional (). AWS Security Token Service AWS STS Jika Anda meminta kredensyal sementara dari AWS STS titik akhir global (`sts.amazonaws.com`), maka Anda harus terlebih dahulu mengatur kompatibilitas Wilayah token sesi agar titik akhir global valid di semua. Wilayah AWS Untuk informasi selengkapnya, lihat Mengelola AWS STSWilayah AWS dalam Panduan Pengguna IAM.

- Titik Akses Multi-Wilayah tidak mendukung permintaan anonim.
- Batasan Titik Akses Multi-Wilayah:
  - IPv6 tidak didukung.
  - Bucket Amazon S3 di Outposts tidak didukung.
  - Titik Akses Multi-Region mendukung operasi penyalinan menggunakan Titik Akses Multi-Wilayah hanya sebagai tujuan saat menggunakan ARN Titik Akses Multi-Wilayah.
  - Fitur Operasi Batch S3 tidak didukung.
- Beberapa AWS SDKs tidak didukung. Untuk mengonfirmasi mana AWS SDKs yang didukung untuk Titik Akses Multi-Wilayah, lihat Kompatibilitas dengan AWS SDKs.
- Kuota layanan untuk Titik Akses Multi-Wilayah adalah sebagai berikut:
  - Maksimal 100 Titik Akses Multi-Wilayah per akun.
  - Jumlah Wilayah maksimal untuk satu Titik Akses Multi-Wilayah adalah 17.
- Setelah membuat Titik Akses Multi-Wilayah, Anda tidak dapat menambahkan, memodifikasi, atau menghapus bucket dari konfigurasi Titik Akses Multi-Wilayah. Untuk mengubah bucket, Anda harus menghapus seluruh Titik Akses Multi-Wilayah dan membuat titik akses baru. Jika bucket lintas akun di Titik Akses Multi-Wilayah Anda dihapus, satu-satunya cara untuk menyambungkan kembali bucket ini adalah dengan membuat ulang bucket, menggunakan nama dan Wilayah yang sama di akun tersebut.
- Bucket yang mendasarinya (di akun yang sama) yang digunakan di Titik Akses Multi-Wilayah hanya dapat dihapus setelah Titik Akses Multi-Wilayah dihapus.
- Semua permintaan bidang kontrol untuk membuat atau mengelola Titik Akses Multi-Wilayah harus dirutekan ke Wilayah `US West (Oregon)`. Untuk permintaan bidang data Titik Akses Multi-Wilayah, tidak perlu menentukan Wilayah secara khusus. Untuk daftar operasi bidang kontrol, lihat Referensi API S3 di Referensi API Amazon Simple Storage Service.

- Untuk bidang kontrol failover Titik Akses Multi-Wilayah, permintaan harus diarahkan ke salah satu dari lima Wilayah yang didukung berikut ini:
  - US East (N. Virginia)
  - US West (Oregon)
  - Asia Pacific (Sydney)
  - Asia Pacific (Tokyo)
  - Europe (Ireland)
- Titik Akses Multi-Wilayah Anda hanya mendukung bucket dalam hal berikut: Wilayah AWS
  - US East (N. Virginia)
  - US East (Ohio)
  - US West (N. California)
  - US West (Oregon)
  - Asia Pacific (Mumbai)
  - Asia Pacific (Osaka)
  - Asia Pacific (Seoul)
  - Asia Pacific (Singapore)
  - Asia Pacific (Sydney)
  - Asia Pacific (Tokyo)
  - Canada (Central)
  - Europe (Frankfurt)
  - Europe (Ireland)
  - Europe (London)
  - Europe (Paris)
  - Europe (Stockholm)
  - South America (São Paulo)

Perutean permintaan Titik Akses Multi-Wilayah

Saat Anda mengajukan permintaan melalui Titik Akses Multi-Wilayah, Amazon S3 akan menentukan bucket yang terkait dengan Titik Akses Multi-Wilayah yang paling dekat dengan Anda. Amazon S3 kemudian mengarahkan permintaan ke bucket itu, terlepas dari AWS Wilayah tempat ia berada.

Setelah Titik Akses Multi-Wilayah merutekan permintaan ke bucket terdekat, Amazon S3 memproses permintaan tersebut seolah-olah Anda membuatnya langsung ke bucket tersebut. Titik Akses Multi-Wilayah tidak mengetahui konten data bucket Amazon S3. Oleh karena itu, bucket yang menerima permintaan mungkin tidak berisi data yang diminta. Untuk membuat set data yang konsisten di bucket Amazon S3 yang terkait dengan Titik Akses Multi-Wilayah, Anda dapat mengonfigurasi Replikasi Lintas Wilayah (CRR) S3. Maka setiap bucket dapat berhasil memenuhi permintaan tersebut.

Amazon S3 mengarahkan permintaan Titik Akses Multi-Wilayah sesuai dengan aturan berikut:

- Amazon S3 mengoptimalkan permintaan yang harus dipenuhi berdasarkan kedekatannya. Amazon S3 melihat bucket yang didukung oleh Titik Akses Multi-Wilayah dan meneruskan permintaan ke bucket yang memiliki jarak terdekat.
- Jika permintaan menentukan sumber daya yang ada (misalnya, `GetObject`), Amazon S3 tidak mempertimbangkan nama objek saat memenuhi permintaan. Ini berarti bahwa meskipun objek ada dalam satu bucket di Titik Akses Multi-Wilayah, permintaan Anda dapat dirutekan ke bucket yang tidak berisi objek tersebut. Situasi ini akan menghasilkan pesan kesalahan 404 yang diberikan kepada klien.

  Untuk mencegah kesalahan 404, sebaiknya konfigurasikan Replikasi Lintas Wilayah (CRR) S3 untuk bucket Anda. Replikasi membantu menyelesaikan masalah potensial ketika objek yang Anda inginkan berada di dalam bucket di Titik Akses Multi-Wilayah, tetapi tidak berada di dalam bucket tertentu yang menjadi tujuan permintaan Anda. Untuk informasi selengkapnya tentang konfigurasi replikasi, lihat Mengkonfigurasi replikasi untuk digunakan dengan Titik Akses Multi-Wilayah.

  Untuk memastikan bahwa permintaan Anda terpenuhi dengan menggunakan objek tertentu yang Anda inginkan, kami juga menyarankan Anda mengaktifkan versi bucket dan menyertakan versi IDs dalam permintaan Anda. Pendekatan ini membantu memastikan bahwa Anda mendapatkan versi yang tepat dari objek yang sedang dicari. Bucket dengan Penentuan Versi yang diaktifkan memungkinkan Anda memulihkan objek yang terlanjur ditimpa secara tidak disengaja. Untuk informasi selengkapnya, lihat Menggunakan Penentuan Versi S3 di bucket S3.
- Jika permintaannya adalah untuk membuat sumber daya (misalnya, `PutObject` atau `CreateMultipartUpload`), Amazon S3 akan memenuhi permintaan tersebut dengan menggunakan bucket jarak terdekat. Misalnya, sebuah perusahaan video yang ingin mendukung pengunggahan video dari mana pun di seluruh dunia. Saat pengguna membuat permintaan PUT ke Titik Akses Multi-Wilayah, objek tersebut dimasukkan ke dalam bucket dengan jarak terdekat. Agar video yang diunggah tersebut dapat diunduh oleh orang lain di seluruh dunia dengan latensi terendah, Anda dapat menggunakan CRR dengan replikasi bidireksional (dua arah). Menggunakan CRR dengan replikasi dua arah akan membuat konten semua bucket yang terkait dengan Titik

Akses Multi-Wilayah tetap tersinkronisasi. Untuk informasi selengkapnya tentang replikasi dengan Titik Akses Multi-Wilayah, lihat Mengkonfigurasi replikasi untuk digunakan dengan Titik Akses Multi-Wilayah.

Kontrol failover Titik Akses Multi-Wilayah Amazon S3

Dengan kontrol failover Titik Akses Multi-Wilayah Amazon S3, Anda dapat mempertahankan kelangsungan bisnis selama terjadinya gangguan lalu lintas Wilayah, sekaligus memberikan arsitektur multi-Wilayah pada aplikasi Anda untuk memenuhi kebutuhan kepatuhan dan redundansi. Jika lalu lintas Regional Anda terganggu, Anda dapat menggunakan kontrol failover Titik Akses Multi-Wilayah untuk memilih mana Wilayah AWS di belakang Titik Akses Multi-Wilayah Amazon S3 yang akan memproses permintaan akses data dan penyimpanan.

Untuk mendukung failover, Anda dapat mengatur Titik Akses Multi-Wilayah dalam konfigurasi aktif-pasif, dengan lalu lintas yang berjalan ke Wilayah aktif selama kondisi normal, dan Wilayah pasif dalam keadaan siaga untuk failover.

Misalnya, untuk melakukan failover ke pilihan Wilayah AWS Anda, Anda mengalihkan lalu lintas dari Wilayah utama (aktif) ke Wilayah sekunder (pasif) Anda. Dalam konfigurasi aktif-pasif seperti ini, satu bucket aktif dan menerima lalu lintas, sedangkan bucket lainnya pasif dan tidak menerima lalu lintas. Bucket pasif digunakan untuk pemulihan bencana. Saat Anda memulai failover, semua lalu lintas (seperti permintaan `GET` atau `PUT`) diarahkan ke bucket dalam status aktif (dalam satu Wilayah) dan menjauhi bucket dalam kondisi pasif (di Wilayah lain).

Jika Anda mengaktifkan Replikasi Lintas Wilayah (CRR) S3 dengan aturan replikasi dua arah, Anda bisa menjaga agar bucket tetap tersinkronisasi selama failover. Selain itu, jika CRR diaktifkan dalam konfigurasi aktif-aktif, Titik Akses Multi-Wilayah Amazon S3 juga dapat mengambil data dari lokasi bucket yang berdekatan, yang akan meningkatkan kinerja aplikasi.

Wilayah AWS dukungan

Dengan kontrol failover Titik Akses Multi-Wilayah Amazon S3, bucket S3 Anda dapat berada di salah satu dari 17 Wilayah di mana Titik Akses Multi-Wilayah didukung. Anda dapat memulai failover di dua Wilayah sekaligus.

> **Note**
>
> Meskipun failover hanya dimulai antara dua Wilayah pada satu waktu, Anda dapat memperbarui status perutean secara terpisah untuk beberapa Wilayah secara bersamaan di Titik Akses Multi-Wilayah Anda.

Topik berikut menunjukkan cara menggunakan dan mengelola kontrol failover Titik Akses Multi-Wilayah Amazon S3.

Topik

- Status perutean Titik Akses Multi-Wilayah Amazon S3
- Menggunakan kontrol failover Titik Akses Multi-Wilayah Amazon S3
- Kesalahan kontrol failover Titik Akses Multi-Wilayah Amazon S3

Status perutean Titik Akses Multi-Wilayah Amazon S3

Konfigurasi failover Titik Akses Multi-Wilayah Amazon S3 Anda menentukan status perutean Wilayah AWS yang digunakan dengan Titik Akses Multi-Wilayah. Anda dapat mengonfigurasi Titik Akses Multi-Wilayah Amazon S3 Anda dalam keadaan aktif-aktif atau aktif-pasif.

- Aktif-aktif–Dalam konfigurasi aktif-aktif, semua permintaan secara otomatis dikirim ke jarak terdekat Wilayah AWS di Titik Akses Multi-Wilayah Anda. Setelah Titik Akses Multi-Wilayah dikonfigurasikan dalam keadaan aktif-aktif, semua Wilayah dapat menerima lalu lintas. Jika terjadi gangguan lalu lintas dalam konfigurasi aktif-aktif, lalu lintas jaringan akan secara otomatis diarahkan ke salah satu Wilayah aktif.
- Aktif-pasif—Dalam konfigurasi aktif-pasif, Wilayah aktif di Titik Akses Multi-Wilayah Anda akan menerima lalu lintas dan yang pasif tidak. Jika Anda ingin menggunakan kontrol failover S3 untuk memulai failover dalam situasi bencana, siapkan Titik Akses Multi-Wilayah Anda dalam konfigurasi aktif-pasif saat Anda menguji dan melakukan perencanaan pemulihan bencana.

Menggunakan kontrol failover Titik Akses Multi-Wilayah Amazon S3

Bagian ini menjelaskan cara mengelola dan menggunakan kontrol failover Titik Akses Multi-Wilayah Amazon S3 Anda dengan menggunakan AWS Management Console.

Ada dua kontrol failover di bagian konfigurasi Failover pada halaman detail Titik Akses Multi-Wilayah Anda di AWS Management Console: Edit status perutean dan Failover. Anda dapat menggunakan kontrol ini sebagai berikut:

- Edit status perutean - Anda dapat mengedit status perutean hingga 17 secara manual Wilayah AWS dalam satu permintaan untuk Titik Akses Multi-Wilayah Anda dengan memilih Edit status perutean. Anda bisa menggunakan Edit status perutean untuk tujuan berikut:
  - Untuk mengatur atau mengedit status perutean dari satu atau beberapa Wilayah di Titik Akses Multi-Wilayah
  - Untuk membuat konfigurasi failover untuk Titik Akses Multi-Wilayah Anda dengan mengonfigurasi dua Wilayah agar berada dalam status aktif-pasif
  - Untuk memindahkan Wilayah Anda secara manual
  - Untuk mengalihkan lalu lintas antar Wilayah secara manual
- Failover—Ketika Anda memulai failover dengan memilih Failover, Anda hanya memperbarui status perutean dari dua Wilayah yang sudah dikonfigurasi agar dalam keadaan aktif-pasif. Selama failover yang telah Anda mulai dengan memilih Failover, status perutean antara dua Wilayah secara otomatis akan beralih.

Mengedit status perutean Wilayah di Titik Akses Multi-Wilayah

Anda dapat memperbarui status perutean hingga 17 secara manual Wilayah AWS dalam satu permintaan untuk Titik Akses Multi-Wilayah Anda dengan memilih Edit status perutean di bagian konfigurasi Failover di halaman detail Titik Akses Multi-Wilayah Anda. Namun, ketika Anda memulai failover dengan memilih Failover, Anda hanya memperbarui status perutean dari dua Wilayah yang sudah dikonfigurasi agar dalam keadaan aktif-pasif. Selama failover yang telah Anda mulai dengan memilih Failover, status perutean antara dua Wilayah secara otomatis akan beralih.

Anda dapat menggunakan Edit status perutean (seperti yang dijelaskan dalam prosedur berikut) untuk tujuan berikut:

- Untuk mengatur atau mengedit status perutean dari satu atau beberapa Wilayah di Titik Akses Multi-Wilayah
- Untuk membuat konfigurasi failover untuk Titik Akses Multi-Wilayah Anda dengan mengonfigurasi dua Wilayah agar berada dalam status aktif-pasif
- Untuk memindahkan Wilayah Anda secara manual
- Untuk mengalihkan lalu lintas antar Wilayah secara manual

Menggunakan konsol S3

Untuk memperbarui status perutean Wilayah di Titik Akses Multi-Wilayah

1. Masuk ke Konsol AWS Manajemen.
2. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
3. Di panel navigasi kiri, pilih Titik Akses Multi-Wilayah.
4. Pilih Titik Akses Multi-Wilayah yang akan Anda perbarui.
5. Pilih tab Replikasi dan failover.
6. Pilih satu atau beberapa Wilayah yang ingin Anda edit status peruteannya.

   > **Note**
   >
   > Untuk memulai failover, setidaknya satu Wilayah AWS harus ditetapkan sebagai Aktif dan satu Wilayah harus ditetapkan sebagai Pasif di Titik Akses Multi-Wilayah Anda.

7. Pilih Edit status perutean.
8. Di kotak dialog yang muncul, pilih Aktif atau Pasif untuk Status Perutean untuk setiap Wilayah.

   Keadaan aktif memungkinkan lalu lintas dialihkan ke Wilayah. Keadaan pasif menghentikan lalu lintas apa pun agar tidak mengarah ke Wilayah.

   Jika Anda membuat konfigurasi failover untuk Titik Akses Multi-Wilayah atau memulai failover, setidaknya satu Wilayah AWS harus ditetapkan sebagai Aktif dan satu Wilayah harus ditetapkan sebagai Pasif di Titik Akses Multi-Wilayah Anda.
9. Pilih Simpan status perutean. Dibutuhkan sekitar 2 menit untuk mengarahkan lalu lintas.

Setelah Anda mengirimkan status perutean Wilayah AWS untuk Titik Akses Multi-Wilayah Anda, Anda dapat memverifikasi perubahan status perutean Anda. Untuk memverifikasi perubahan ini, buka Amazon CloudWatch di https://console.aws.amazon.com/cloudwatch/untuk memantau pergeseran lalu lintas permintaan data Amazon S3 Anda (misalnya, `GET` dan `PUT` permintaan) antara Wilayah aktif dan pasif. Koneksi yang ada tidak akan terputus selama failover. Koneksi yang ada akan berlanjut hingga mencapai status berhasil atau gagal.

Menggunakan AWS CLI

> **Note**
>
> Anda dapat menjalankan perintah AWS CLI perutean Titik Akses Multi-Wilayah terhadap salah satu dari lima Wilayah ini:
>
> - `ap-southeast-2`
> - `ap-northeast-1`
> - `us-east-1`
> - `us-west-2`
> - `eu-west-1`

Contoh perintah berikut memperbarui konfigurasi rute Titik Akses Multi-Wilayah Anda saat ini. Untuk memperbarui status aktif atau pasif bucket, tetapkan nilai `TrafficDialPercentage` ke `100` untuk aktif dan ke `0` untuk pasif. Dalam contoh ini, *amzn-s3-demo-bucket1* diatur ke aktif, dan *amzn-s3-demo-bucket2* diatur ke pasif. Untuk menggunakan perintah contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3control submit-multi-region-access-point-routes
--region ap-southeast-2
--account-id 123456789012
--mrap MultiRegionAccessPoint_ARN
--route-updates Bucket=amzn-s3-demo-bucket1,TrafficDialPercentage=100
                Bucket=amzn-s3-demo-bucket2
,TrafficDialPercentage=0
```

Contoh perintah berikut mendapatkan konfigurasi perutean Titik Akses Multi-Wilayah Anda yang telah diperbarui. Untuk menggunakan contoh perintah ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3control get-multi-region-access-point-routes
--region eu-west-1
--account-id 123456789012
--mrap MultiRegionAccessPoint_ARN
```

Memulai failover

Saat Anda memulai failover dengan memilih Failover di bagian Konfigurasi failover di halaman detail Titik Akses Multi-Wilayah Anda, lalu lintas permintaan Amazon S3 secara otomatis akan dialihkan ke alternatif Wilayah AWS. Proses failover membutuhkan waktu 2 menit.

Anda dapat memulai failover di dua Wilayah AWS sekaligus (dari 17 Wilayah di mana Titik Akses Multi-Wilayah didukung). Peristiwa failover kemudian masuk AWS CloudTrail. Setelah failover selesai, Anda dapat memantau lalu lintas Amazon S3 dan pembaruan perutean lalu lintas apa pun ke Wilayah aktif baru di Amazon. CloudWatch

> **Important**
>
> Agar semua metadata dan objek tetap sinkron di seluruh bucket selama replikasi data, sebaiknya Anda membuat aturan replikasi dua arah dan mengaktifkan sinkronisasi modifikasi replika sebelum mengonfigurasi kontrol failover Anda.
> Aturan replikasi dua arah membantu memastikan bahwa ketika data ditulis ke bucket Amazon S3 yang digunakan untuk failover lalu lintas, data tersebut kemudian direplikasi kembali ke bucket sumber. Sinkronisasi modifikasi replika membantu memastikan bahwa metadata objek juga disinkronkan antar bucket selama replikasi dua arah.
> Untuk informasi selengkapnya tentang konfigurasi replikasi untuk mendukung failover, lihat the section called “Replikasi bucket”.

Untuk memulai failover antara bucket yang direplikasi

1. Masuk ke Konsol AWS Manajemen.
2. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
3. Di panel navigasi kiri, pilih Titik Akses Multi-Wilayah.
4. Pilih Titik Akses Multi-Wilayah yang ingin Anda gunakan untuk memulai failover.
5. Pilih tab Replikasi dan failover.
6. Gulir ke bawah ke bagian Konfigurasi failover dan pilih dua Wilayah AWS.

> **Note**
>
> Untuk memulai failover, setidaknya satu Wilayah AWS harus ditetapkan sebagai Aktif dan satu Wilayah harus ditetapkan sebagai Pasif di Titik Akses Multi-Wilayah Anda.

Keadaan aktif memungkinkan lalu lintas diarahkan ke suatu Wilayah. Keadaan pasif menghentikan lalu lintas apa pun agar tidak mengarah ke Wilayah.

7. Pilih Failover.
8. Di kotak dialog, pilih Failover lagi untuk memulai proses failover. Selama proses ini, status perutean kedua Wilayah secara otomatis dialihkan. Semua lalu lintas baru diarahkan ke Wilayah yang menjadi aktif, dan lalu lintas berhenti diarahkan ke Wilayah yang menjadi pasif. Dibutuhkan sekitar 2 menit untuk mengarahkan lalu lintas.

   Setelah memulai proses failover, Anda dapat memverifikasi perubahan lalu lintas. Untuk memverifikasi perubahan ini, buka Amazon CloudWatch di https://console.aws.amazon.com/cloudwatch/untuk memantau pergeseran lalu lintas permintaan data Amazon S3 Anda (misalnya, `GET` dan `PUT` permintaan) antara Wilayah aktif dan pasif. Koneksi yang ada tidak akan terputus selama failover. Koneksi yang ada akan berlanjut hingga mencapai status berhasil atau gagal.

Melihat kontrol perutean Titik Akses Multi-Wilayah Amazon S3 Anda

Menggunakan konsol S3

Untuk melihat kontrol perutean untuk Titik Akses Multi-Wilayah Amazon S3 Anda

1. Masuk ke Konsol AWS Manajemen.
2. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
3. Di panel navigasi kiri, pilih Titik Akses Multi-Wilayah.
4. Pilih Titik Akses Multi-Wilayah yang ingin Anda tinjau.
5. Pilih tab Replikasi dan failover. Halaman ini menampilkan detail konfigurasi perutean dan ringkasan untuk Titik Akses Multi-Wilayah, aturan replikasi terkait, dan metrik replikasi. Anda dapat melihat status perutean Wilayah di bagian Konfigurasi Failover.

Menggunakan AWS CLI

AWS CLI Perintah contoh berikut mendapatkan konfigurasi rute Titik Akses Multi-Region Anda saat ini untuk Wilayah yang ditentukan. Untuk menggunakan contoh perintah ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3control get-multi-region-access-point-routes
```

```
--region eu-west-1
--account-id 123456789012
--mrap MultiRegionAccessPoint_ARN
```

> **Note**
>
> Perintah ini hanya dapat dijalankan pada kelima Wilayah ini:
>
> - ap-southeast-2
> - ap-northeast-1
> - us-east-1
> - us-west-2
> - eu-west-1

Kesalahan kontrol failover Titik Akses Multi-Wilayah Amazon S3

Saat memperbarui konfigurasi failover untuk Titik Akses Multi-Wilayah, Anda mungkin akan mengalami salah satu kesalahan berikut:

- HTTP 400 Tidak Valid: Kesalahan ini dapat terjadi jika Anda memasukkan ARN Titik Akses Multi-Wilayah yang tidak valid saat memperbarui konfigurasi failover Anda. Anda dapat mengonfirmasi ARN Titik Akses Multi-Wilayah dengan meninjau kebijakan Titik Akses Multi-Wilayah. Untuk meninjau atau memperbarui kebijakan Titik Akses Multi-Wilayah, lihat Mengedit kebijakan Titik Akses Multi-Wilayah. Kesalahan ini juga dapat terjadi jika Anda menggunakan string kosong atau string acak saat memperbarui kontrol failover Titik Akses Multi-Wilayah Amazon S3. Pastikan untuk menggunakan format ARN Titik Akses Multi-Wilayah:

  `arn:aws:s3::account-id:accesspoint/MultiRegionAccessPoint_alias`
- HTTP 503 Lambat: Kesalahan ini terjadi jika Anda mengirim terlalu banyak permintaan dalam waktu singkat. Permintaan yang ditolak dapat menyebabkan kesalahan.
- HTTP 409 Konflik: Kesalahan ini terjadi ketika dua atau lebih permintaan pembaruan konfigurasi rute bersamaan menargetkan satu Titik Akses Multi-Wilayah. Permintaan pertama berhasil, tetapi permintaan lainnya gagal dengan kesalahan.
- Metode HTTP 405 Tidak Diizinkan: Kesalahan ini terjadi ketika Anda telah memilih Titik Akses Multi-Wilayah dengan hanya satu Wilayah AWS saat memulai failover. Anda harus memilih dua Wilayah sebelum dapat memulai failover. Jika tidak, kesalahan akan ditampilkan.

Mengkonfigurasi replikasi untuk digunakan dengan Titik Akses Multi-Wilayah

Saat Anda membuat permintaan ke titik akhir Titik Akses Multi-Wilayah, Amazon S3 secara otomatis merutekan permintaan ke bucket yang terdekat dengan Anda. Amazon S3 tidak mempertimbangkan isi permintaan saat membuat keputusan ini. Jika Anda membuat permintaan untuk `GET` objek, permintaan tersebut mungkin akan dirutekan ke bucket yang tidak memiliki salinan objek ini. Jika hal itu terjadi, Anda akan menerima kode status HTTP 404 (Tidak Ditemukan). Untuk informasi selengkapnya tentang perutean Titik Akses Multi-Wilayah, lihat the section called “Meminta perutean”.

Jika Anda ingin Titik Akses Multi-Wilayah dapat mengambil objek tanpa melihat bucket mana yang menerima permintaan, Anda harus mengonfigurasi Amazon S3 Replikasi Lintas Wilayah (CRR).

Misalnya, kita lihat Titik Akses Multi-Wilayah dengan tiga bucket:

- Sebuah bucket bernama *`amzn-s3-demo-bucket1`* di Wilayah `US West (Oregon)` yang berisi objek `my-image.jpg`
- Sebuah bucket bernama *`amzn-s3-demo-bucket2`* di Wilayah `Asia Pacific (Mumbai)` yang berisi objek `my-image.jpg`
- Sebuah bucket bernama *`amzn-s3-demo-bucket`* di Wilayah `Europe (Frankfurt)` yang tidak berisi objek `my-image.jpg`

Dalam kondisi ini, jika Anda membuat `GetObject` permintaan untuk objek `my-image.jpg`, keberhasilannya tergantung pada bucket yang menerima permintaan tersebut. Karena tidak mempertimbangkan konten permintaan, Amazon S3 ini dapat merutekan `GetObject` permintaan Anda ke bucket *`amzn-s3-demo-bucket`* jika bucket tersebut memberikan respons terdekat. Meskipun objek Anda berada dalam bucket di Titik Akses Multi-Wilayah, Anda akan mendapatkan kesalahan HTTP 404 Tidak Ditemukan karena bucket individual yang menerima permintaan Anda tidak memiliki objek tersebut.

Mengaktifkan Replikasi Lintas Wilayah (CRR) akan membantu mencegah hal ini terjadi. Dengan aturan replikasi yang tepat, objek `my-image.jpg` disalin ke bucket *`amzn-s3-demo-bucket`*. Oleh karena itu, jika Amazon S3 merutekan permintaan Anda ke bucket itu, Anda sekarang dapat mengambil objek tersebut.

Replikasi berfungsi seperti biasa dengan bucket yang ditetapkan ke Titik Akses Multi-Wilayah. Amazon S3 tidak melakukan penanganan replikasi khusus dengan bucket yang ada di Titik Akses

Multi-Wilayah. Untuk informasi selengkapnya tentang konfigurasi replikasi di bucket, lihat Menyiapkan ikhtisar replikasi langsung.

Rekomendasi untuk menggunakan replikasi dengan Titik Akses Multi-Wilayah

Untuk kinerja replikasi terbaik saat bekerja dengan Titik Akses Multi-Wilayah, kami merekomendasikan hal berikut:

- Konfigurasi Kontrol Waktu Replikasi S3 (S3 RTC). Untuk mereplikasi data di berbagai Wilayah dalam jangka waktu yang dapat diprediksi, Anda dapat menggunakan S3 RTC. S3 RTC mereplikasi 99,99 persen objek baru yang disimpan di Amazon S3 dalam 15 menit (didukung oleh perjanjian tingkat layanan). Untuk informasi selengkapnya, lihat the section called “Menggunakan Kontrol Waktu Replikasi S3”. Ada biaya tambahan untuk S3 RTC. Untuk informasi, lihat Harga Amazon S3.
- Gunakan replikasi dua arah (bidireksional) untuk mendukung agar bucket tetap sinkron ketika bucket diperbarui melalui Titik Akses Multi-Wilayah. Untuk informasi selengkapnya, lihat the section called “Buat aturan replikasi dua arah untuk Titik Akses Multi-Wilayah Anda”.
- Buat Titik Akses Multi-Wilayah lintas akun untuk mereplikasi data ke bucket secara terpisah Akun AWS. pendekatan ini memberikan pemisahan pada tingkat akun, sehingga data dapat diakses dan direplikasi di berbagai akun di Wilayah yang berbeda selain bucket sumber. Menyiapkan Titik Akses Multi-Wilayah lintas akun tidak dikenakan biaya tambahan. Jika Anda pemilik bucket tetapi tidak memiliki Titik Akses Multi-Wilayah, Anda hanya membayar untuk transfer data dan biaya permintaan. Pemilik Titik Akses Multi-Wilayah membayar biaya perutean data dan akselerasi internet. Untuk informasi selengkapnya, lihat Harga Amazon S3.
- Aktifkan sinkronisasi modifikasi replika untuk setiap aturan replikasi agar perubahan metadata pada objek Anda tetap sinkron. Untuk informasi selengkapnya, lihat Mengaktifkan sinkronisasi modifikasi replika.
- Aktifkan CloudWatch metrik Amazon untuk memantau peristiwa replikasi. CloudWatch biaya metrik berlaku. Untuk informasi lebih lanjut, lihat harga Amazon CloudWatch.

Topik

- Buat aturan replikasi satu arah untuk Titik Akses Multi-Wilayah Anda
- Buat aturan replikasi dua arah untuk Titik Akses Multi-Wilayah Anda
- Lihat aturan replikasi untuk Titik Akses Multi-Wilayah Anda

Buat aturan replikasi satu arah untuk Titik Akses Multi-Wilayah Anda

Aturan replikasi memungkinkan penyalinan objek secara otomatis dan asinkron di seluruh bucket. Aturan replikasi satu arah membantu memastikan bahwa data sepenuhnya direplikasi dari bucket sumber di bucket satu Wilayah AWS ke tujuan di Wilayah lain. Saat replikasi satu arah disiapkan, aturan replikasi dari bucket sumber (*amzn-s3-demo-bucket*) ke bucket tujuan (*amzn-s3-demo-bucket*) akan dibuat. Seperti halnya semua aturan replikasi, Anda dapat menerapkan aturan replikasi satu arah ke seluruh bucket Amazon S3 atau ke sebagian objek yang difilter dengan awalan atau tag objek.

> **Important**
>
> Sebaiknya gunakan replikasi satu arah jika pengguna Anda hanya akan menggunakan objek di bucket tujuan Anda. Jika pengguna Anda akan mengunggah atau memodifikasi objek di bucket tujuan, gunakan replikasi dua arah untuk menjaga agar semua bucket tetap sinkron. Selain itu, sebaiknya gunakan replikasi dua arah jika Anda berencana untuk menggunakan Titik Akses Multi-Wilayah untuk failover. Untuk mengatur replikasi dua arah, lihat the section called “Buat aturan replikasi dua arah untuk Titik Akses Multi-Wilayah Anda”.

Untuk membuat aturan replikasi satu arah untuk Titik Akses Multi-Wilayah Anda

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Titik Akses Multi-Wilayah.
3. Pilih nama Titik Akses Multi-Wilayah Anda.
4. Pilih tab Replikasi dan failover.
5. Gulir ke bawah ke bagian Aturan replikasi, lalu pilih Buat aturan replikasi. Pastikan Anda memiliki izin yang cukup untuk membuat aturan replikasi, jika tidak, maka Penentuan Versi akan dinonaktifkan.

> **Note**
>
> Anda dapat membuat aturan replikasi hanya untuk bucket di akun Anda sendiri. Untuk membuat aturan replikasi bucket eksternal, pemilik bucket harus membuat aturan replikasi untuk bucket tersebut.

6. Pada halaman Buat aturan replikasi, pilih templat Replikasi objek dari satu atau beberapa bucket sumber ke satu atau beberapa bucket tujuan.

   > **Important**
   >
   > Saat Anda membuat aturan replikasi dengan menggunakan templat ini, aturan tersebut menggantikan aturan replikasi yang ada yang sudah ditetapkan ke bucket.
   > Untuk menambah atau memodifikasi aturan replikasi yang ada tanpa menggantinya, buka tab Manajemen setiap bucket di konsol, lalu edit aturan di bagian Aturan replikasi. Anda juga dapat menambah atau memodifikasi aturan replikasi yang ada dengan menggunakan AWS CLI, SDKs, atau REST API. Untuk informasi selengkapnya, lihat Elemen file konfigurasi replikasi.

7. Di bagian Sumber dan tujuan, di bawah Bucket sumber, pilih satu atau beberapa bucket yang ingin Anda replikasi objeknya. Semua bucket (sumber dan tujuan) yang dipilih untuk replikasi harus mengaktifkan S3 Penentuan Versi, dan setiap bucket harus berada di Wilayah AWS yang berbeda. Untuk informasi selengkapnya tentang Penentuan Versi S3, lihat Menggunakan versi di bucket Amazon S3.

   Di bawah Bucket tujuan, pilih satu atau beberapa bucket yang ingin Anda gunakan sebagai tujuan replikasi objek.

8. Di bagian konfigurasi aturan replikasi, pilih apakah aturan replikasi akan Diaktifkan atau Dinonaktifkan saat dibuat.

   > **Note**
   >
   > Anda tidak dapat memasukkan nama di kotak Nama aturan replikasi. Nama aturan replikasi dihasilkan berdasarkan konfigurasi Anda saat membuat aturan replikasi.

9. Di bagian Cakupan, pilih cakupan yang sesuai untuk replikasi Anda.

   - Untuk mereplikasi seluruh bucket, pilih Terapkan ke semua objek di bucket.
   - Untuk mereplikasi subset objek dalam bucket, pilih Batasi cakupan aturan ini menggunakan satu atau beberapa filter.

     Anda dapat memfilter objek dengan menggunakan prefiks, tag objek, atau kombinasi keduanya.

- Untuk membatasi replikasi ke semua objek yang memiliki nama yang dimulai dengan string yang sama (misalnya `pictures`), masukkan prefiks ke dalam kotak Prefiks.

  Jika memasukkan prefiks yang merupakan nama folder, Anda harus menggunakan pembatas seperti / (garis miring) untuk menunjukkan tingkat hierarki (misalnya, `pictures/`). Untuk informasi selengkapnya tentang prefiks, lihat Mengatur objek menggunakan prefiks.

- Untuk mereplikasi semua objek yang memiliki satu atau beberapa tag objek, pilih Tambahkan tag dan masukkan pasangan nilai kunci ke dalam kotak. Untuk menambahkan tag lain, ulangi prosedur. Untuk informasi selengkapnya tentang tag objek, lihat Mengategorikan penyimpanan Anda menggunakan tag.

10. Gulir ke bawah ke bagian Opsi replikasi tambahan, dan pilih opsi replikasi yang ingin Anda terapkan.

> **Note**
>
> Sebaiknya Anda menerapkan opsi berikut:
>
> - Kontrol waktu replikasi (RTC)—Untuk mereplikasi data Anda di berbagai Wilayah dalam jangka waktu yang dapat diprediksi, Anda dapat menggunakan Kontrol Waktu Replikasi S3 (S3 RTC). S3 RTC mereplikasi 99,99 persen objek baru yang disimpan di Amazon S3 dalam 15 menit (didukung oleh perjanjian tingkat layanan). Untuk informasi selengkapnya, lihat the section called “Menggunakan Kontrol Waktu Replikasi S3”.
> - Metrik replikasi dan notifikasi — Aktifkan CloudWatch metrik Amazon untuk memantau peristiwa replikasi.
> - Replikasi penanda penghapusan–Penanda penghapusan yang dibuat oleh operasi penghapusan S3 akan direplikasi. Penanda penghapusan yang dibuat oleh aturan siklus hidup tidak direplikasi. Untuk informasi selengkapnya, lihat Mereplikasi penanda penghapusan di antara bucket.
>
> Ada biaya tambahan untuk S3 RTC dan metrik CloudWatch replikasi dan notifikasi. Untuk informasi selengkapnya, lihat Harga Amazon S3 dan harga Amazon CloudWatch .

11. Jika Anda menulis aturan replikasi baru untuk menggantikan yang sudah ada, pilih Saya menyadari bahwa dengan memilih Buat aturan replikasi, aturan replikasi yang sudah ada akan ditimpa.
12. Pilih Buat aturan replikasi untuk membuat dan menyimpan aturan replikasi satu arah baru Anda.

Buat aturan replikasi dua arah untuk Titik Akses Multi-Wilayah Anda

Aturan replikasi memungkinkan penyalinan objek secara otomatis dan asinkron di seluruh bucket. Aturan replikasi dua arah (juga dikenal sebagai aturan replikasi bidireksional) memastikan bahwa data sepenuhnya disinkronkan antara dua atau lebih bucket di Wilayah AWS yang berbeda. Saat replikasi dua arah disiapkan, aturan replikasi dari bucket sumber (DOC-EXAMPLE-BUCKET-1) ke bucket yang berisi replika (DOC-EXAMPLE-BUCKET-2) akan dibuat. Kemudian, aturan replikasi kedua dari bucket yang berisi replika (DOC-EXAMPLE-BUCKET-2) ke bucket sumber (DOC-EXAMPLE-BUCKET-1) akan dibuat.

Seperti halnya semua aturan replikasi, Anda dapat menerapkan aturan replikasi dua arah ke seluruh bucket Amazon S3 atau ke subset objek yang difilter oleh prefiks atau tag objek. Anda juga dapat menjaga perubahan metadata ke objek tetap sinkron dengan mengaktifkan sinkronisasi modifikasi replika untuk setiap aturan replikasi. Anda dapat mengaktifkan sinkronisasi modifikasi replika melalui konsol Amazon S3, AWS CLI API AWS SDKs REST Amazon S3, atau. AWS CloudFormation

Untuk memantau kemajuan replikasi objek dan metadata objek di Amazon CloudWatch, aktifkan metrik dan notifikasi Replikasi S3. Untuk informasi selengkapnya, lihat Memantau kemajuan dengan metrik replikasi dan notifikasi peristiwa Amazon S3.

Untuk membuat aturan replikasi dua arah untuk Titik Akses Multi-Wilayah Anda

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Titik Akses Multi-Wilayah.
3. Pilih nama Titik Akses Multi-Wilayah yang ingin Anda perbarui.
4. Pilih tab Replikasi dan failover.
5. Gulir ke bawah ke bagian Aturan replikasi, lalu pilih Buat aturan replikasi.
6. Pada halaman Buat aturan replikasi, pilih templat Replikasi objek di antara semua bucket yang ditentukan. Templat Replikasi objek di antara semua bucket yang ditentukan menyiapkan replikasi dua arah (dengan kemampuan failover) untuk bucket Anda.

> **Important**
>
> Saat Anda membuat aturan replikasi dengan menggunakan templat ini, aturan tersebut menggantikan aturan replikasi yang ada yang sudah ditetapkan ke bucket.
> Untuk menambah atau memodifikasi aturan replikasi yang ada tanpa menggantinya, buka tab Manajemen setiap bucket di konsol, lalu edit aturan di bagian Aturan replikasi. Anda juga dapat menambahkan atau memodifikasi aturan replikasi yang ada dengan menggunakan AWS CLI, AWS SDKs, atau Amazon S3 REST API. Untuk informasi selengkapnya, lihat Elemen file konfigurasi replikasi.

7. Di bagian Bucket, pilih setidaknya dua bucket yang ingin Anda replikasi objeknya. Semua bucket yang dipilih untuk replikasi harus mengaktifkan S3 Penentuan Versi, dan setiap bucket harus berada di Wilayah AWS yang berbeda. Untuk informasi selengkapnya tentang Penentuan Versi S3, lihat Menggunakan versi di bucket Amazon S3.

> **Note**
>
> Pastikan Anda memiliki izin baca dan replikasi yang diperlukan untuk membuat replikasi, jika tidak Anda akan menemukan kesalahan. Untuk informasi selengkapnya, lihat Membuat peran IAM.

8. Di bagian konfigurasi aturan replikasi, pilih apakah aturan replikasi akan Diaktifkan atau Dinonaktifkan saat dibuat.

> **Note**
>
> Anda tidak dapat memasukkan nama di kotak Nama aturan replikasi. Nama aturan replikasi dihasilkan berdasarkan konfigurasi Anda saat membuat aturan replikasi.

9. Di bagian Cakupan, pilih cakupan yang sesuai untuk replikasi Anda.

   - Untuk mereplikasi seluruh bucket, pilih Terapkan ke semua objek di bucket.
   - Untuk mereplikasi subset objek dalam bucket, pilih Batasi cakupan aturan ini menggunakan satu atau beberapa filter.

     Anda dapat memfilter objek dengan menggunakan prefiks, tag objek, atau kombinasi keduanya.

- Untuk membatasi replikasi ke semua objek yang memiliki nama yang dimulai dengan string yang sama (misalnya `pictures`), masukkan prefiks ke dalam kotak Prefiks.

  Jika memasukkan prefiks yang merupakan nama folder, Anda harus menggunakan / (garis miring) sebagai karakter terakhir (misalnya, `pictures/`).

- Untuk mereplikasi semua objek yang memiliki satu atau beberapa tag objek, pilih Tambahkan tag dan masukkan pasangan nilai kunci ke dalam kotak. Untuk menambahkan tag lain, ulangi prosedur. Untuk informasi selengkapnya tentang tag objek, lihat Mengategorikan penyimpanan Anda menggunakan tag.

10. Gulir ke bawah ke bagian Opsi replikasi tambahan, dan pilih opsi replikasi yang ingin Anda terapkan.

> **Note**
>
> Sebaiknya Anda menerapkan opsi berikut, terutama jika Anda bermaksud mengonfigurasi Titik Akses Multi-Wilayah untuk mendukung failover:
>
> - Kontrol waktu replikasi (RTC)—Untuk mereplikasi data Anda di berbagai Wilayah dalam jangka waktu yang dapat diprediksi, Anda dapat menggunakan Kontrol Waktu Replikasi S3 (S3 RTC). S3 RTC mereplikasi 99,99 persen objek baru yang disimpan di Amazon S3 dalam 15 menit (didukung oleh perjanjian tingkat layanan). Untuk informasi selengkapnya, lihat the section called “Menggunakan Kontrol Waktu Replikasi S3”.
> - Metrik replikasi dan notifikasi — Aktifkan CloudWatch metrik Amazon untuk memantau peristiwa replikasi.
> - Replikasi penanda penghapusan–Penanda penghapusan yang dibuat oleh operasi penghapusan S3 akan direplikasi. Penanda penghapusan yang dibuat oleh aturan siklus hidup tidak direplikasi. Untuk informasi selengkapnya, lihat Mereplikasi penanda penghapusan di antara bucket.
> - Sinkronisasi modifikasi replika–Aktifkan sinkronisasi modifikasi replika untuk setiap aturan replikasi agar perubahan metadata pada objek Anda tetap sinkron. Untuk informasi selengkapnya, lihat Mengaktifkan sinkronisasi modifikasi replika.
>
> Ada biaya tambahan untuk S3 RTC dan metrik CloudWatch replikasi dan notifikasi. Untuk informasi selengkapnya, lihat Harga Amazon S3 dan harga Amazon CloudWatch.

11. Jika Anda menulis aturan replikasi baru untuk menggantikan yang sudah ada, pilih Saya menyadari bahwa dengan memilih Buat aturan replikasi, aturan replikasi yang sudah ada akan ditimpa.
12. Pilih Buat aturan replikasi untuk membuat dan menyimpan aturan replikasi dua arah Anda yang baru.

Lihat aturan replikasi untuk Titik Akses Multi-Wilayah Anda

Dengan Titik Akses Multi-Wilayah, Anda dapat membuat aturan replikasi satu arah atau aturan replikasi dua arah (bidireksional). Untuk informasi tentang cara mengelola aturan replikasi, lihat Mengelola aturan replikasi menggunakan konsol Amazon S3.

Untuk melihat aturan replikasi untuk Titik Akses Multi-Wilayah

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Titik Akses Multi-Wilayah.
3. Pilih nama Titik Akses Multi-Wilayah Anda.
4. Pilih tab Replikasi dan failover.
5. Gulir ke bawah ke bagian Aturan replikasi. Bagian ini mencantumkan semua aturan replikasi yang telah dibuat untuk Titik Akses Multi-Wilayah Anda.

> **Note**
>
> Jika Anda telah menambahkan bucket dari akun lain ke Titik Akses Multi-Wilayah ini, Anda harus memiliki izin `s3:GetBucketReplication` dari pemilik bucket untuk melihat aturan replikasi bucket tersebut.

Menggunakan Titik Akses Multi-Wilayah dengan operasi API yang didukung

Amazon S3 menyediakan serangkaian operasi untuk mengelola Titik Akses Multi-Wilayah. Amazon S3 memproses beberapa operasi ini secara sinkron dan beberapa secara asinkron. Saat Anda menginvoke operasi asinkron, Amazon S3 terlebih dahulu akan mengotorisasi operasi yang diminta secara sinkron. Jika otorisasi berhasil, Amazon S3 akan memberikan token yang dapat Anda gunakan untuk melacak kemajuan dan hasil operasi yang diminta.

> **Note**
>
> Permintaan yang dibuat melalui konsol Amazon S3 selalu sinkron. Konsol menunggu hingga permintaan selesai sebelum memperbolehkan Anda mengirimkan permintaan lain.

Anda dapat melihat status saat ini dan hasil operasi asinkron dengan menggunakan konsol, atau Anda dapat menggunakannya `DescribeMultiRegionAccessPointOperation` di, AWS CLI AWS SDKs, atau REST API. Amazon S3 memberikan token pelacakan untuk merespons operasi asinkron. Anda menyertakan token pelacakan itu sebagai argumen untuk `DescribeMultiRegionAccessPointOperation`. Saat Anda menyertakan token pelacakan, Amazon S3 kemudian memberikan status saat ini dan hasil operasi yang ditentukan, termasuk kesalahan atau informasi sumber daya yang relevan. Amazon S3 melakukan operasi `DescribeMultiRegionAccessPointOperation` secara sinkron.

Semua permintaan bidang kontrol untuk membuat atau mengelola Titik Akses Multi-Wilayah harus dirutekan ke Wilayah `US West (Oregon)`. Untuk permintaan bidang data Titik Akses Multi-Wilayah, tidak perlu menentukan Wilayah secara khusus. Untuk bidang kontrol failover Titik Akses Multi-Wilayah, permintaan harus dirutekan ke salah satu dari lima Wilayah yang didukung. Untuk informasi selengkapnya tentang Wilayah yang didukung Titik Akses Multi-Wilayah, lihatPembatasan dan batasan Titik Akses Multi-Wilayah.

Selain itu, Anda harus memberikan `s3:ListAllMyBuckets` izin kepada pengguna, peran, atau entitas lain AWS Identity and Access Management (IAM) yang membuat permintaan untuk mengelola Titik Akses Multi-Wilayah.

Berikut ini adalah contoh cara menggunakan Titik Akses Multi-Wilayah dengan operasi yang kompatibel di Amazon S3.

Topik

- Kompatibilitas Titik Akses Multi-Wilayah dengan Layanan AWS dan AWS SDKs
- Kompatibilitas Titik Akses Multi-Wilayah dengan operasi S3
- Lihat konfigurasi perutean Titik Akses Multi-Wilayah
- Perbarui kebijakan bucket Amazon S3
- Memperbarui konfigurasi rute Titik Akses Multi-Wilayah
- Tambahkan objek ke bucket di Titik Akses Multi-Wilayah
- Mengambil objek dari Titik Akses Multi-Wilayah

- Buat daftar objek yang disimpan dalam bucket yang mendasari Titik Akses Multi-Wilayah Anda
- Gunakan URL yang telah ditetapkan sebelumnya dengan Titik Akses Multi-Wilayah
- Gunakan bucket yang dikonfigurasi dengan Pembayaran Pemohon dengan Titik Akses Multi-Wilayah

Kompatibilitas Titik Akses Multi-Wilayah dengan Layanan AWS dan AWS SDKs

Untuk menggunakan Titik Akses Multi-Wilayah dengan aplikasi yang memerlukan nama bucket Amazon S3, gunakan Nama Sumber Daya Amazon (ARN) dari Titik Akses Multi-Wilayah saat membuat permintaan menggunakan SDK. AWS Untuk memeriksa mana yang AWS SDKs kompatibel dengan Titik Akses Multi-Wilayah, lihat Kompatibilitas dengan AWS SDKs.

Kompatibilitas Titik Akses Multi-Wilayah dengan operasi S3

Anda dapat menggunakan operasi API bidang data Amazon S3 berikut untuk melakukan tindakan pada objek dalam bucket yang terkait dengan Titik Akses Multi-Wilayah Anda. Operasi S3 berikut dapat menerima Titik Akses Multi-Wilayah: ARNs

- AbortMultipartUpload
- CompleteMultipartUpload
- CreateMultipartUpload
- DeleteObject
- DeleteObjectTagging
- GetObject
- GetObjectAcl
- GetObjectLegalHold
- GetObjectRetention
- GetObjectTagging
- HeadObject
- ListMultipartUploads
- ListObjectsV2
- ListParts
- PutObject
- PutObjectAcl

- PutObjectLegalHold
- PutObjectRetention
- PutObjectTagging
- RestoreObject
- UploadPart

> **Note**
>
> Titik Akses Multi-Region mendukung operasi penyalinan menggunakan Titik Akses Multi-Wilayah hanya sebagai tujuan saat menggunakan ARN Titik Akses Multi-Wilayah.

Anda dapat menggunakan operasi bidang kontrol Amazon S3 berikut untuk membuat dan mengelola Titik Akses Multi-Wilayah:

- CreateMultiRegionAccessPoint
- DescribeMultiRegionAccessPointOperation
- GetMultiRegionAccessPoint
- GetMultiRegionAccessPointPolicy
- GetMultiRegionAccessPointPolicyStatus
- GetMultiRegionAccessPointRoutes
- ListMultiRegionAccessPoints
- PutMultiRegionAccessPointPolicy
- SubmitMultiRegionAccessPointRoutes

Lihat konfigurasi perutean Titik Akses Multi-Wilayah

AWS CLI

Contoh perintah berikut mengambil konfigurasi rute Titik Akses Multi-Wilayah sehingga Anda dapat melihat status perutean saat ini untuk bucket Anda. Untuk menggunakan contoh perintah ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3control get-multi-region-access-point-routes
--region eu-west-1
```

```
--account-id 111122223333
--mrap arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap
```

SDK for Java

Kode SDK for Java berikut mengambil konfigurasi rute Titik Akses Multi-Wilayah sehingga Anda dapat melihat status perutean saat ini untuk bucket Anda. Untuk menggunakan contoh sintaks ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
S3ControlClient s3ControlClient = S3ControlClient.builder()
    .region(Region.US_EAST_1)
    .credentialsProvider(credentialsProvider)
    .build();

GetMultiRegionAccessPointRoutesRequest request =
 GetMultiRegionAccessPointRoutesRequest.builder()
    .accountId("111122223333")
    .mrap("arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap")
    .build();

GetMultiRegionAccessPointRoutesResponse response =
 s3ControlClient.getMultiRegionAccessPointRoutes(request);
```

SDK for JavaScript

SDK berikut untuk JavaScript kode mengambil konfigurasi rute Titik Akses Multi-Wilayah sehingga Anda dapat melihat status perutean saat ini untuk bucket Anda. Untuk menggunakan contoh sintaks ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
const REGION = 'us-east-1'

const s3ControlClient = new S3ControlClient({
  region: REGION
})

export const run = async () => {
  try {
    const data = await s3ControlClient.send(
      new GetMultiRegionAccessPointRoutesCommand({
        AccountId: '111122223333',
        Mrap: 'arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap',
      })
    )
```

```
    console.log('Success', data)
    return data
  } catch (err) {
    console.log('Error', err)
  }
}

run()
```

SDK for Python

SDK berikut untuk kode Python mengambil konfigurasi rute Titik Akses Multi-Wilayah sehingga Anda dapat melihat status perutean saat ini untuk bucket Anda. Untuk menggunakan contoh sintaks ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
s3.get_multi_region_access_point_routes(
        AccountId=111122223333,
        Mrap=arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap)['Routes']
```

Perbarui kebijakan bucket Amazon S3

Untuk memberikan akses yang tepat, Anda juga harus memperbarui kebijakan bucket Amazon S3 yang mendasarinya. Contoh berikut mendelegasikan kontrol akses ke kebijakan Titik Akses Multi-Wilayah. Setelah Anda mendelegasikan kontrol akses ke kebijakan Titik Akses Multi-Wilayah, kebijakan bucket tidak lagi digunakan untuk kontrol akses saat permintaan dibuat melalui Titik Akses Multi-Wilayah.

Berikut adalah contoh kebijakan bucket yang mendelegasikan kontrol akses ke kebijakan Titik Akses Multi-Wilayah. Untuk menggunakan contoh kebijakan bucket ini, ganti *`user input placeholders`* dengan informasi Anda sendiri. Untuk menerapkan kebijakan ini melalui AWS CLI `put-bucket-policy` perintah, seperti yang ditunjukkan pada contoh berikutnya, simpan kebijakan dalam file, misalnya,`policy.json`.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Principal": { "AWS": "*" },
    "Effect": "Allow",
    "Action": ["s3:*"],
    "Resource": ["arn:aws:s3:::111122223333/*", "arn:aws:s3:::amzn-s3-demo-bucket"],
```

```
    "Condition": {
      "StringEquals": {
        "s3:DataAccessPointAccount": "444455556666"
      }
    }
  }
}
```

Contoh perintah `put-bucket-policy` berikut mengaitkan kebijakan bucket S3 yang diperbarui dengan bucket S3 Anda:

```
aws s3api put-bucket-policy
  --bucket amzn-s3-demo-bucket
  --policy file:///tmp/policy.json
```

Memperbarui konfigurasi rute Titik Akses Multi-Wilayah

Contoh perintah berikut memperbarui konfigurasi rute Titik Akses Multi-Wilayah. Perintah rute Titik Akses Multi-Wilayah dapat dijalankan terhadap lima Wilayah berikut:

- `ap-southeast-2`
- `ap-northeast-1`
- `us-east-1`
- `us-west-2`
- `eu-west-1`

Dalam konfigurasi perutean Titik Akses Multi-Wilayah, Anda dapat mengatur bucket ke status perutean aktif atau pasif. Bucket aktif menerima lalu lintas, sedangkan bucket pasif tidak. Anda dapat menyetel status perutean bucket dengan mengatur nilai `TrafficDialPercentage` bucket ke `100` untuk aktif atau `0` untuk pasif.

AWS CLI

Contoh perintah berikut ini memperbarui konfigurasi perutean Titik Akses Multi-Wilayah Anda. Dalam contoh ini, *amzn-s3-demo-bucket1* diatur ke status aktif dan *amzn-s3-demo-bucket2* diatur ke pasif. Untuk menggunakan contoh perintah ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3control submit-multi-region-access-point-routes
```

```
--region ap-southeast-2
--account-id 111122223333
--mrap arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap
--route-updates Bucket=amzn-s3-demo-bucket1,TrafficDialPercentage=100
                Bucket=amzn-s3-demo-bucket2,TrafficDialPercentage=0
```

SDK for Java

Kode SDK for Java berikut memperbarui konfigurasi perutean Titik Akses Multi-Wilayah Anda. Untuk menggunakan contoh sintaks ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
S3ControlClient s3ControlClient = S3ControlClient.builder()
    .region(Region.ap-southeast-2)
    .credentialsProvider(credentialsProvider)
    .build();

SubmitMultiRegionAccessPointRoutesRequest request =
 SubmitMultiRegionAccessPointRoutesRequest.builder()
    .accountId("111122223333")
    .mrap("arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap")
    .routeUpdates(
        MultiRegionAccessPointRoute.builder()
            .region("eu-west-1")
            .trafficDialPercentage(100)
            .build(),
        MultiRegionAccessPointRoute.builder()
            .region("ca-central-1")
            .bucket("111122223333")
            .trafficDialPercentage(0)
            .build()
    )
    .build();

SubmitMultiRegionAccessPointRoutesResponse response =
 s3ControlClient.submitMultiRegionAccessPointRoutes(request);
```

SDK for JavaScript

SDK berikut untuk JavaScript kode memperbarui konfigurasi rute Titik Akses Multi-Wilayah Anda. Untuk menggunakan contoh sintaks ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
const REGION = 'ap-southeast-2'

const s3ControlClient = new S3ControlClient({
  region: REGION
})

export const run = async () => {
  try {
    const data = await s3ControlClient.send(
      new SubmitMultiRegionAccessPointRoutesCommand({
        AccountId: '111122223333',
        Mrap: 'arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap',
        RouteUpdates: [
          {
            Region: 'eu-west-1',
            TrafficDialPercentage: 100,
          },
          {
            Region: 'ca-central-1',
            Bucket: 'amzn-s3-demo-bucket1',
            TrafficDialPercentage: 0,
          },
        ],
      })
    )
    console.log('Success', data)
    return data
  } catch (err) {
    console.log('Error', err)
  }
}

run()
```

SDK for Python

Kode SDK untuk Python berikut ini memperbarui konfigurasi rute Titik Akses Multi-Wilayah Anda. Untuk menggunakan contoh sintaks ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
s3.submit_multi_region_access_point_routes(
        AccountId=111122223333,
        Mrap=arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap,
```

```
        RouteUpdates= [{
            'Bucket': amzn-s3-demo-bucket,
            'Region': ap-southeast-2,
            'TrafficDialPercentage': 10
        }])
```

Tambahkan objek ke bucket di Titik Akses Multi-Wilayah

Untuk menambahkan objek ke bucket yang terkait dengan Titik Akses Multi-Wilayah, Anda dapat menggunakan PutObjectoperasi. Agar semua bucket di Titik Akses Multi-Wilayah tetap sinkron, aktifkan Replikasi Lintas Wilayah.

> **Note**
>
> Untuk menggunakan operasi ini, Anda harus memiliki izin `s3:PutObject` untuk Titik Akses Multi-Wilayah. Untuk informasi selengkapnya tentang persyaratan izin Titik Akses Multi-Wilayah, lihat Izin.

AWS CLI

Contoh permintaan bidang data berikut mengunggah *example.txt* ke Titik Akses Multi-Wilayah yang ditentukan. Untuk menggunakan contoh ini, ganti *user input placeholders*dengan informasi Anda sendiri.

```
aws s3api put-object --bucket
 arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap --key example.txt --
body example.txt
```

SDK for Java

```
S3Client s3Client = S3Client.builder()
        .build();

PutObjectRequest objectRequest = PutObjectRequest.builder()
        .bucket("arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap")
        .key("example.txt")
        .build();

s3Client.putObject(objectRequest, RequestBody.fromString("Hello S3!"));
```

SDK for JavaScript

```
const client = new S3Client({});

async function putObjectExample() {
    const command = new PutObjectCommand({
        Bucket: "arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap",
        Key: "example.txt",
        Body: "Hello S3!",
    });

    try {
        const response = await client.send(command);
        console.log(response);
    } catch (err) {
        console.error(err);
    }
}
```

SDK for Python

```
import boto3

client = boto3.client('s3')
client.put_object(
    Bucket='arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap',
    Key='example.txt',
    Body='Hello S3!'
)
```

Mengambil objek dari Titik Akses Multi-Wilayah

Untuk mengambil objek dari Multi-Region Access Point, Anda dapat menggunakan GetObjectoperasi.

> **Note**
>
> Untuk menggunakan operasi API ini, Anda harus memiliki izin `s3:GetObject` untuk Titik Akses Multi-Wiayah. Untuk informasi selengkapnya tentang persyaratan izin Titik Akses Multi-Wilayah, lihat Izin.

AWS CLI

Contoh permintaan bidang data berikut mengambil *example.txt* dari Titik Akses Multi-Wilayah yang ditentukan dan mengunduhnya sebagai *downloaded_example.txt*. Untuk menggunakan contoh ini, ganti *user input placeholders*dengan informasi Anda sendiri.

```
aws s3api get-object --bucket
 arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap --
key example.txt downloaded_example.txt
```

SDK for Java

```
S3Client s3 = S3Client
   .builder()
   .build();

GetObjectRequest getObjectRequest = GetObjectRequest.builder()
    .bucket("arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap")
    .key("example.txt")
    .build();

s3Client.getObject(getObjectRequest);
```

SDK for JavaScript

```
const client = new S3Client({})

async function getObjectExample() {
    const command = new GetObjectCommand({
        Bucket: "arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap",
        Key: "example.txt"
    });

    try {
        const response = await client.send(command);
        console.log(response);
    } catch (err) {
        console.error(err);
    }
}
```

SDK for Python

```
import boto3

client = boto3.client('s3')
client.get_object(
    Bucket='arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap',
    Key='example.txt'
)
```

Buat daftar objek yang disimpan dalam bucket yang mendasari Titik Akses Multi-Wilayah Anda

Untuk mengembalikan daftar objek yang disimpan dalam bucket yang mendasari Titik Akses Multi-Wilayah Anda, gunakan ListObjectsV2operasi. Dalam contoh perintah berikut, semua objek untuk Titik Akses Multi-Wilayah yang ditentukan dicantumkan dengan menggunakan ARN untuk Titik Akses Multi-Wilayah. Dalam hal ini, ARN Titik Akses Multi-Wilayah adalah:

`arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap`

> **Note**
>
> Untuk menggunakan operasi API ini, Anda harus memiliki izin `s3:ListBucket` untuk Titik Akses Multi-Wilayah dan bucket yang mendasarinya. Untuk informasi selengkapnya tentang persyaratan izin Titik Akses Multi-Wilayah, lihat Izin.

AWS CLI

Contoh permintaan bidang data berikut mencantumkan objek dalam bucket yang mendasari Titik Akses Multi-Wilayah yang ditentukan oleh ARN. Untuk menggunakan contoh ini, ganti *user input placeholders*dengan informasi Anda sendiri.

```
aws s3api list-objects-v2 --bucket
 arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap
```

SDK for Java

```
S3Client s3Client = S3Client.builder()
        .build();
```

```
String bucketName = "arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap";

ListObjectsV2Request listObjectsRequest = ListObjectsV2Request
    .builder()
    .bucket(bucketName)
    .build();

 s3Client.listObjectsV2(listObjectsRequest);
```

SDK for JavaScript

```
const client = new S3Client({});

async function listObjectsExample() {
    const command = new ListObjectsV2Command({
        Bucket: "arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap",
    });

    try {
        const response = await client.send(command);
        console.log(response);
    } catch (err) {
        console.error(err);
    }
}
```

SDK for Python

```
import boto3

client = boto3.client('s3')
client.list_objects_v2(
    Bucket='arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap'
)
```

Gunakan URL yang telah ditetapkan sebelumnya dengan Titik Akses Multi-Wilayah

Anda dapat menggunakan URL yang telah ditetapkan sebelumnya untuk membuat URL sehingga orang lain dapat mengakses bucket Amazon S3 Anda melalui Titik Akses Multi-Wilayah Amazon S3. Saat membuat URL yang telah ditetapkan sebelumnya, kaitkan dengan tindakan objek tertentu,

seperti unggah S3 (`PutObject`) atau unduhan S3 (`GetObject`). Anda dapat membagikan URL yang telah ditetapkan sebelumnya, dan siapa pun yang memiliki akses ke URL dapat melakukan tindakan yang disertakan dalam URL seolah-olah mereka adalah pengguna yang membuat URL sendiri.

Presigned URLs memiliki tanggal kedaluwarsa. Ketika masa berlaku habis, URL tidak akan berfungsi lagi.

Sebelum Anda menggunakan S3 Multi-Region Access Points dengan presigned URLs, periksa kompatibilitas AWS SDK dengan algoritma Sigv4a. Verifikasi bahwa versi SDK Anda mendukung Sigv4A sebagai implementasi penandatanganan yang digunakan untuk menandatangani permintaan global Wilayah AWS . Untuk informasi selengkapnya tentang penggunaan presigned URLs dengan Amazon S3, lihat Berbagi objek menggunakan presigned. URLs

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan Titik Akses Multi-Region dengan URLs presigned. Untuk menggunakan contoh ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

AWS CLI

```
aws s3 presign
 arn:aws:s3::123456789012:accesspoint/MultiRegionAccessPoint_alias/example-file.txt
```

SDK for Python

```
import logging
import boto3
from botocore.exceptions import ClientError

s3_client = boto3.client('s3',aws_access_key_id='xxx',aws_secret_access_key='xxx')
s3_client.generate_presigned_url(HttpMethod='PUT',ClientMethod="put_object",
 Params={'Bucket':'arn:aws:s3::123456789012:accesspoint/
abcdef0123456.mrap','Key':'example-file'})
```

SDK for Java

```
S3Presigner s3Presigner = S3Presigner.builder()
    .credentialsProvider(StsAssumeRoleCredentialsProvider.builder()
        .refreshRequest(assumeRole)
        .stsClient(stsClient)
```

```
        .build())
    .build();

GetObjectRequest getObjectRequest = GetObjectRequest.builder()
    .bucket("arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap")
    .key("example-file")
    .build();

GetObjectPresignRequest preSignedReq = GetObjectPresignRequest.builder()
    .getObjectRequest(getObjectRequest)
    .signatureDuration(Duration.ofMinutes(10))
    .build();

PresignedGetObjectRequest presignedGetObjectRequest =
 s3Presigner.presignGetObject(preSignedReq);
```

> **Note**
>
> Untuk menggunakan Sigv4a dengan kredensi keamanan sementara—misalnya, saat menggunakan peran IAM—pastikan Anda meminta kredensi sementara dari titik akhir Regional di (), bukan titik akhir global. AWS Security Token Service AWS STS Jika Anda menggunakan endpoint global for AWS STS (`sts.amazonaws.com`), AWS STS akan menghasilkan kredensi sementara dari titik akhir global, yang tidak didukung oleh Sig4A. Akibatnya, akan terjadi kesalahan. Untuk mengatasi masalah ini, gunakan salah satu titik akhir Regional yang terdaftar untuk AWS STS.

Gunakan bucket yang dikonfigurasi dengan Pembayaran Pemohon dengan Titik Akses Multi-Wilayah

Jika bucket S3 yang terkait dengan Titik Akses Multi-Wilayah Anda dikonfigurasi untuk menggunakan Pembayaran Pemohon, pemohon akan membayar permintaan bucket, unduhan, dan biaya terkait Titik Akses Multi-Wilayah. Untuk informasi selengkapnya, lihat Harga Amazon S3.

Berikut adalah contoh permintaan bidang data ke Titik Akses Multi-Wilayah yang terhubung ke bucket Pembayaran Pemohon.

AWS CLI

Untuk mengunduh objek dari Titik Akses Multi-Wilayah yang terhubung ke bucket Requester Pays, Anda harus menentukan `--request-payer requester` sebagai bagian dari

get-objectpermintaan. Anda juga harus menentukan nama file di dalam bucket dan lokasi penyimpanan file yang diunduh.

```
aws s3api get-object --bucket MultiRegionAccessPoint_ARN --request-payer requester
 --key example-file-in-bucket.txt example-location-of-downloaded-file.txt
```

SDK for Java

Untuk mengunduh objek dari Titik Akses Multi-Wilayah yang terhubung ke bucket Pembayaran Pemohon, Anda harus menentukan `RequestPayer.REQUESTER` sebagai bagian dari permintaan `GetObject` Anda. Anda juga harus menentukan nama file dalam bucket, serta lokasi penyimpanannya.

```
GetObjectResponse getObjectResponse = s3Client.getObject(GetObjectRequest.builder()
    .key("example-file.txt")
    .bucket("arn:aws:s3::
123456789012:accesspoint/abcdef0123456.mrap")
    .requestPayer(RequestPayer.REQUESTER)
    .build()
).response();
```

Memantau dan mencatat permintaan yang dilakukan melalui Titik Akses Multi-Wilayah ke sumber daya yang mendasarinya

Permintaan log Amazon S3 yang dibuat melalui Titik Akses Multi-Wilayah dan permintaan yang dibuat ke operasi API yang mengelolanya, seperti dan `CreateMultiRegionAccessPoint` dan `GetMultiRegionAccessPointPolicy`. Permintaan yang dibuat ke Amazon S3 melalui Titik Akses Multi-Wilayah muncul di log akses server Amazon S3 dan AWS CloudTrail log dengan nama host Titik Akses Multi-Wilayah. Nama host titik akses memiliki format *`MRAP_alias`*`.accesspoint.s3-global.amazonaws.com`. Misalnya, katakanlah Anda memiliki bucket berikut dan konfigurasi Titik Akses Multi-Wilayah:

- Sebuah bucket bernama `my-bucket-usw2` di Wilayah `us-west-2` yang berisi objek `my-image.jpg`.
- Sebuah bucket bernama `my-bucket-aps1` di Wilayah `ap-south-1` yang berisi objek `my-image.jpg`.
- Sebuah bucket bernama `my-bucket-euc1` di Region `eu-central-1` yang tidak berisi objek bernama `my-image.jpg`.

- Titik Akses Multi-Wilayah bernama `my-mrap` dengan alias `mfzwi23gnjvgw.mrap` yang dikonfigurasi untuk memenuhi permintaan dari ketiga bucket.
- ID AWS akun Anda adalah`123456789012`.

Permintaan yang dibuat untuk mengambil `my-image.jpg` langsung melalui salah satu bucket muncul di log Anda dengan nama host *`bucket_name`*`.s3.`*`Region`*`.amazonaws.com`.

Jika Anda membuat permintaan melalui Titik Akses Multi-Wilayah, Amazon S3 akan terlebih dahulu menentukan bucket mana di Wilayah yang berbeda yang terdekat. Setelah menentukan bucket yang akan digunakan untuk memenuhi permintaan, Amazon S3 akan mengirimkan permintaan ke bucket tersebut dan mencatat operasi dengan menggunakan nama host Titik Akses Multi-Wilayah. Dalam contoh ini, jika Amazon S3 meneruskan permintaan ke `my-bucket-aps1`, log Anda akan mencerminkan permintaan `GET` yang berhasil untuk `my-image.jpg` dari`my-bucket-aps1`, menggunakan nama host `mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com`.

> **Important**
>
> Titik Akses Multi-Wilayah tidak mengetahui konten data dari bucket yang mendasarinya. Oleh karena itu, bucket yang mendapatkan permintaan mungkin tidak berisi data yang diminta. Misalnya, jika Amazon S3 menentukan bahwa bucket `my-bucket-euc1` adalah yang terdekat, log Anda akan mencerminkan permintaan `GET` yang gagal untuk `my-image.jpg` dari `my-bucket-euc1`, menggunakan nama host `mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com`. Jika permintaan dialihkan ke `my-bucket-usw2`, log Anda akan menunjukkan permintaan `GET` yang berhasil.

Untuk informasi selengkapnya tentang log akses server Amazon S3, lihat Pencatatan permintaan dengan pencatatan akses server. Untuk informasi lebih lanjut tentang AWS CloudTrail, lihat Apa itu AWS CloudTrail? dalam AWS CloudTrail User Guide.

Memantau dan mencatat permintaan yang dibuat untuk operasi API manajemen Titik Akses Multi-Wilayah

Amazon S3 menyediakan beberapa operasi API untuk mengelola Titik Akses Multi-Wilayah, seperti `CreateMultiRegionAccessPoint` dan `GetMultiRegionAccessPointPolicy`. Saat Anda membuat permintaan ke operasi API ini dengan menggunakan AWS Command Line Interface (AWS CLI),, atau Amazon S3 REST API AWS SDKs, Amazon S3 memproses permintaan ini secara asinkron. Asalkan Anda memiliki izin yang sesuai untuk permintaan tersebut,

Amazon S3 akan memberikan token untuk permintaan ini. Anda dapat menggunakan token ini `DescribeAsyncOperation` untuk membantu melihat status operasi asinkron yang sedang berlangsung. Amazon S3 memproses permintaan `DescribeAsyncOperation` secara sinkron. Untuk melihat status permintaan asinkron, Anda dapat menggunakan konsol Amazon S3, AWS CLI, SDKs atau REST API.

> **Note**
>
> Konsol hanya menampilkan status permintaan asinkron yang dibuat dalam 14 hari sebelumnya. Untuk melihat status permintaan lama, gunakan, AWS CLI SDKs, atau REST API.

Operasi manajemen asinkron dapat berada di salah satu dari beberapa status berikut:

NEW

Amazon S3 telah menerima permintaan dan sedang bersiap untuk melakukan operasi.

IN_PROGRESS

Amazon S3 saat ini sedang melakukan operasi.

SUCCESS

Operasi berhasil. Respons tersebut menyertakan informasi yang relevan, seperti alias Titik Akses Multi-Wilayah untuk permintaan `CreateMultiRegionAccessPoint`.

FAILED

Operasi gagal. Respons tersebut menyertakan pesan kesalahan yang menjelaskan alasan kegagalan permintaan.

Menggunakan AWS CloudTrail dengan Titik Akses Multi-Region

Anda dapat menggunakan AWS CloudTrail untuk melihat, mencari, mengunduh, mengarsipkan, menganalisis, dan menanggapi aktivitas akun di seluruh AWS infrastruktur Anda. Dengan Titik Akses Multi-Wilayah dan CloudTrail pencatatan, Anda dapat mengidentifikasi hal-hal berikut:

- Siapa atau apa yang mengambil tindakan
- Sumber daya mana yang ditindaklanjuti
- Saat peristiwa terjadi

- Detail lainnya mengenai peristiwa

Anda dapat menggunakan informasi pencatatan ini untuk membantu Anda menganalisis dan merespons aktivitas yang terjadi melalui Titik Akses Multi-Wilayah.

Cara mengatur AWS CloudTrail untuk Titik Akses Multi-Wilayah

Untuk mengaktifkan CloudTrail pencatatan untuk operasi apa pun yang terkait dengan pembuatan atau pemeliharaan Titik Akses Multi-Wilayah, Anda harus mengonfigurasi CloudTrail pencatatan untuk merekam peristiwa di Wilayah AS Barat (Oregon). Anda harus mengatur konfigurasi pencatatan dengan cara ini tanpa melihat di Wilayah mana Anda berada saat membuat permintaan, atau Wilayah mana yang didukung oleh Titik Akses Multi-Wilayah. Semua permintaan untuk membuat atau mengelola Titik Akses Multi-Wilayah dirutekan melalui Wilayah AS Barat (Oregon). Sebaiknya Anda menambahkan Wilayah ini ke jalur yang sudah ada atau membuat jalur baru yang berisi Wilayah ini dan semua Wilayah yang terkait dengan Titik Akses Multi-Wilayah.

Amazon S3 mencatat semua permintaan yang dibuat melalui Titik Akses Multi-Wilayah dan permintaan yang dibuat ke operasi API yang mengelola titik akses, seperti `CreateMultiRegionAccessPoint` dan `GetMultiRegionAccessPointPolicy`. Saat Anda mencatat permintaan ini melalui Titik Akses Multi-Wilayah, permintaan tersebut muncul di AWS CloudTrail log Anda dengan nama host dari Titik Akses Multi-Wilayah. Misalnya, jika Anda membuat permintaan ke bucket melalui Titik Akses Multi-Region dengan alias`mfzwi23gnjvgw.mrap`, entri dalam CloudTrail log akan memiliki nama host. `mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com`

Titik Akses Multi-Wilayah merutekan permintaan ke ember terdekatKarena perilaku ini, saat Anda melihat CloudTrail log untuk Titik Akses Multi-Wilayah, Anda akan melihat permintaan dibuat ke bucket yang mendasarinya. Beberapa dari permintaan tersebut mungkin merupakan permintaan langsung ke bucket yang tidak dirutekan melalui Titik Akses Multi-Wilayah. Ingat selalu hal ini saat meninjau lalu lintas. Saat bucket berada di Titik Akses Multi-Wilayah, permintaan masih dapat dibuat ke bucket tersebut secara langsung tanpa melalui Titik Akses Multi-Wilayah.

Dalam proses pembuatan dan pengelolaan Titik Akses Multi-Wilayah, terdapat peristiwa asinkron yang terlibat di dalamnya. Permintaan asinkron tidak memiliki peristiwa penyelesaian di log. CloudTrail Untuk informasi selengkapnya tentang permintaan asinkron, lihat Memantau dan mencatat permintaan yang dibuat untuk operasi API manajemen Titik Akses Multi-Wilayah.

Untuk informasi lebih lanjut tentang AWS CloudTrail, lihat Apa itu AWS CloudTrail? dalam AWS CloudTrail User Guide.

Mempertahankan beberapa versi objek dengan S3 Versioning

Penentuan Versi di Amazon S3 adalah cara menyimpan beberapa varian objek dalam bucket yang sama. Anda dapat menggunakan fitur Penentuan Versi S3 untuk menyimpan, mengambil, dan memulihkan setiap versi dari setiap objek yang disimpan dalam bucket Anda. Dengan Penentuan Versi, Anda dapat lebih mudah memulihkan dari tindakan pengguna yang tidak diinginkan dan kegagalan aplikasi. Setelah Penentuan Versi diaktifkan untuk bucket, jika Amazon S3 menerima beberapa permintaan tulis untuk objek yang sama secara bersamaan, itu akan menyimpan semua objek.

Bucket dengan Penentuan Versi yang diaktifkan memungkinkan Anda memulihkan objek dari penghapusan atau penimpaan yang tidak disengaja. Misalnya, jika Anda menghapus objek, Amazon S3 menyisipkan penanda hapus alih-alih menghapus objek secara permanen. Penanda hapus menjadi versi objek saat ini. Jika Anda menimpa sebuah objek, akan muncul versi objek baru dalam bucket. Anda dapat selalu memulihkan versi sebelumnya. Untuk informasi selengkapnya, lihat Menghapus versi objek dari bucket dengan dukungan Penentuan Versi.

Secara default, Penentuan Versi S3 dinonaktifkan pada bucket, dan Anda harus mengaktifkannya secara eksplisit. Untuk informasi selengkapnya, lihat Mengaktifkan Penentuan Versi pada bucket.

> **Note**
>
> - SOAP API tidak mendukung Penentuan Versi S3. Dukungan SOAP melalui HTTP dihilangkan, tetapi masih tersedia melalui HTTPS. Fitur Amazon S3 baru tidak mendukung SOAP.
> - Tarif normal Amazon S3 berlaku untuk setiap versi objek yang disimpan dan ditransfer. Setiap versi objek adalah seluruh objek; bukan hanya sebuah diff dari versi sebelumnya. Dengan demikian, jika memiliki tiga versi objek yang disimpan, Anda akan dikenakan biaya untuk tiga objek.

Bucket tanpa versi, dengan dukungan Penentuan Versi, dan dengan Penentuan Versi ditangguhkan

Bucket bisa berada di salah satu dari tiga status:

- Tanpa versi (default)

- Dengan dukungan Penentuan Versi
- Dengan Penentuan Versi ditangguhkan

Anda mengaktifkan dan menangguhkan Penentuan Versi di tingkat bucket. Setelah Anda mengaktifkan Penentuan Versi sebuah bucket, itu tidak akan pernah dapat kembali ke status tanpa versi. Namun, Anda dapat menangguhkan Penentuan Versi pada bucket tersebut.

Status Penentuan Versi berlaku untuk semua (tidak pernah sebagian) objek dalam bucket tersebut. Saat Anda mengaktifkan Penentuan Versi di bucket, semua objek baru akan mendapatkan Penentuan Versi dan diberi ID versi unik. Objek yang sudah ada di bucket pada saat Penentuan Versi diaktifkan akan selalu mendapatkan Penentuan Versi dan diberi ID versi unik saat diubah oleh permintaan di masa mendatang. Perhatikan hal-hal berikut:

- Objek yang disimpan di bucket sebelum Anda mengatur status Penentuan Versi memiliki ID versi `null`. Saat Anda mengaktifkan Penentuan Versi, objek yang ada di bucket tidak berubah. Perubahannya adalah cara Amazon S3 menangani objek di masa mendatang. Untuk informasi selengkapnya, lihat Bekerja dengan objek di dalam bucket dengan dukungan Penentuan Versi.
- Pemilik bucket (atau pengguna dengan izin yang sesuai) dapat menangguhkan Penentuan Versi untuk menghentikan pengumpulan versi objek. Saat Anda menangguhkan Penentuan Versi, objek yang ada di bucket tidak berubah. Perubahannya adalah cara Amazon S3 menangani objek di masa mendatang. Untuk informasi selengkapnya, lihat Bekerja dengan objek dalam bucket dengan Penentuan Versi ditangguhkan.

Menggunakan Penentuan Versi S3 dengan Siklus Hidup S3

Untuk menyesuaikan pendekatan retensi data Anda dan mengontrol biaya penyimpanan, gunakan Penentuan Versi objek dengan Siklus Hidup S3. Untuk informasi selengkapnya, lihat Mengelola siklus hidup objek. Untuk informasi tentang membuat konfigurasi Siklus Hidup S3 menggunakan AWS Management Console,, AWS CLI, atau REST AWS SDKs API, lihat. Menyetel konfigurasi Siklus Hidup S3 pada bucket

> **Important**
>
> Jika Anda memiliki konfigurasi siklus hidup kedaluwarsa objek di bucket tanpa versi dan ingin mempertahankan perilaku penghapusan permanen yang sama saat mengaktifkan Penentuan Versi, Anda harus menambahkan konfigurasi kedaluwarsa yang lama. Konfigurasi siklus hidup kedaluwarsa noncurrent mengelola penghapusan versi objek noncurrent di bucket

berkemampuan versi. (Bucket berkemampuan versi mempertahankan satu versi objek saat ini, dan nol atau lebih versi objek yang tidak terkini.) Untuk informasi selengkapnya, lihat Menyetel konfigurasi Siklus Hidup S3 pada bucket.

Untuk informasi tentang bekerja dengan Penentuan Versi S3, lihat topik-topik berikut.

Topik

- Cara kerja Penentuan Versi S3
- Mengaktifkan Penentuan Versi pada bucket
- Mengonfigurasi penghapusan MFA
- Bekerja dengan objek di dalam bucket dengan dukungan Penentuan Versi
- Bekerja dengan objek dalam bucket dengan Penentuan Versi ditangguhkan
- Pemecahan masalah pembuatan versi

Cara kerja Penentuan Versi S3

Gunakan Penentuan Versi S3 untuk menyimpan beberapa versi dari sebuah objek di dalam satu bucket sehingga Anda dapat memulihkan objek yang terhapus atau ditimpa secara tidak sengaja. Misalnya, jika Anda menerapkan Penentuan Versi S3 ke bucket, perubahan berikut akan terjadi:

- Jika Anda menghapus objek, alih-alih menghapus objek secara permanen, Amazon S3 menyisipkan penanda hapus yang menjadi versi objek saat ini. Anda dapat selalu memulihkan versi sebelumnya. Untuk informasi selengkapnya, lihat Menghapus versi objek dari bucket dengan dukungan Penentuan Versi.
- Jika Anda menimpa sebuah objek, Amazon S3 akan menambahkan versi objek baru dalam bucket. Versi sebelumnya tetap ada di bucket dan menjadi versi lama. Anda dapat selalu memulihkan versi sebelumnya.

> **Note**
>
> Tarif normal Amazon S3 berlaku untuk setiap versi objek yang disimpan dan ditransfer. Setiap versi objek adalah seluruh objek; bukan hanya sebuah diff dari versi sebelumnya. Dengan demikian, jika memiliki tiga versi objek yang disimpan, Anda akan dikenakan biaya untuk tiga objek.

Setiap bucket S3 yang Anda buat memiliki subsumber daya Penentuan Versi yang terkait dengannya. (Untuk informasi selengkapnya, lihat opsi konfigurasi bucket tujuan umum.) Secara default, bucket Anda tanpa versi, dan subsumber daya Penentuan Versi menyimpan konfigurasi Penentuan Versi yang kosong, sebagai berikut.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
</VersioningConfiguration>
```

Untuk mengaktifkan Penentuan Versi, Anda dapat mengirimkan permintaan ke Amazon S3 dengan konfigurasi Penentuan Versi yang menyertakan status `Enabled`.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>Enabled</Status>
</VersioningConfiguration>
```

Untuk menangguhkan Penentuan Versi, Anda menetapkan nilai status ke `Suspended`.

> **Note**
>
> Saat Anda mengaktifkan Penentuan Versi pada bucket untuk pertama kalinya, mungkin diperlukan beberaoa saat agar perubahan dapat dilakukan sepenuhnya. Saat perubahan ini menyebar, Anda mungkin mengalami `HTTP 404 NoSuchKey` kesalahan intermiten untuk permintaan ke objek yang dibuat atau diperbarui setelah mengaktifkan versi. Kami menyarankan Anda menunggu selama 15 menit setelah mengaktifkan Penentuan Versi sebelum mengeluarkan operasi tulis (`PUT` atau `DELETE`) pada objek di dalam bucket.

Pemilik bucket dan semua pengguna resmi AWS Identity and Access Management (IAM) dapat mengaktifkan pembuatan versi. Pemilik ember adalah Akun AWS yang menciptakan ember. Untuk informasi selengkapnya tentang izin, lihat Identity and Access Management untuk Amazon S3.

Untuk informasi selengkapnya tentang mengaktifkan dan menonaktifkan Pembuatan Versi S3 dengan menggunakan AWS Management Console, AWS Command Line Interface (AWS CLI), atau REST API, lihat. the section called “Mengaktifkan Penentuan Versi pada bucket”

Topik

- Versi IDs
- Alur kerja Penentuan Versi

Versi IDs

Jika Anda mengaktifkan Penentuan Versi untuk bucket, Amazon S3 secara otomatis menghasilkan ID versi unik untuk objek yang sedang disimpan. Misalnya, dalam satu ember Anda dapat memiliki dua objek dengan kunci yang sama (nama objek) tetapi versi yang berbeda IDs, seperti `photo.gif` (versi 111111) dan `photo.gif` (versi 121212).
Diagram yang menggambarkan bucket berkemampuan versi yang memiliki dua objek dengan kunci yang sama tetapi versi yang berbeda. IDs

Setiap objek memiliki ID versi, terlepas dari apakah Penentuan Versi S3 diaktifkan atau tidak. Jika Penentuan Versi S3 tidak diaktifkan, Amazon S3 menetapkan nilai ID versi ke `null`. Jika Penentuan Versi S3 diaktifkan, Amazon S3 akan menetapkan nilai ID versi untuk objek tersebut. Nilai ini membedakan objek dengan versi lain dari kunci yang sama.

Saat Anda mengaktifkan Penentuan Versi S3 pada bucket yang sudah ada, objek yang sudah disimpan dalam bucket tidak akan berubah. Versi IDs (`null`), konten, dan izinnya tetap sama. Setelah Anda mengaktifkan Penentuan Versi S3, setiap objek yang ditambahkan ke bucket akan mendapatkan sebuah ID versi, yang membedakannya dengan versi lain dari kunci yang sama.

Hanya Amazon S3 yang menghasilkan versi IDs, dan mereka tidak dapat diedit. Versi IDs Unicode, UTF-8 dikodekan, URL-ready, string buram yang panjangnya tidak lebih dari 1.024 byte. Berikut ini adalah contohnya:

```
3sL4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY+MTRCxf3vjVBH40Nr8X8gdRQBpUMLUo
```

> **Note**
>
> Untuk kesederhanaan, contoh lain dalam topik ini menggunakan jauh lebih pendek IDs.

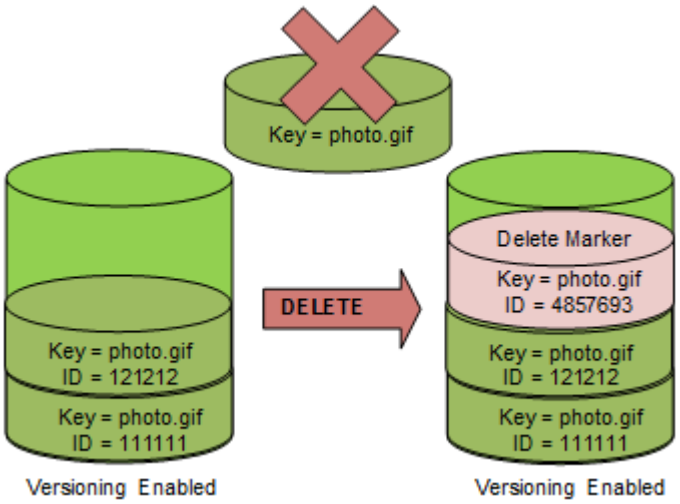
Alur kerja Penentuan Versi

Saat Anda `PUT` sebuah objek dalam sebuah bucket dengan dukungan Penentuan Versi, versi nonterkini tidak ditimpa. Seperti yang ditunjukkan pada gambar berikut, ketika versi `photo.gif` baru `PUT` ke dalam bucket yang sudah berisi objek dengan nama yang sama, perilaku berikut terjadi:

- Objek asli (ID = 111111) tetap berada di bucket.
- Amazon S3 menghasilkan ID versi baru (121212), dan menambahkan versi objek yang lebih baru ini ke bucket.

Dengan fungsi ini, Anda dapat mengambil versi objek sebelumnya jika objek telah ditimpa atau dihapus secara tidak sengaja.

Saat Anda DELETE sebuah objek, semua versi tetap berada dalam bucket, dan Amazon S3 menyisipkan penanda hapus, seperti yang ditunjukkan pada gambar berikut.



Delete marker menjadi versi objek saat ini. Secara default, permintaan GET akan mengambil versi yang paling terakhir disimpan. Melakukan permintaan `GET Object` ketika versi saat ini merupakan penanda hapus akan menampilkan pesan kesalahan `404 Not Found`, seperti yang ditunjukkan pada gambar berikut.

Akan tetapi, Anda dapat GET versi objek nonterkini dengan menentukan ID versinya. Di gambar berikut, Anda akan GET versi objek tertentu, 111111. Amazon S3 akan menampilkan versi objek tersebut meskipun bukan versi saat ini.

Untuk informasi selengkapnya, lihat Mengambil versi objek dari bucket dengan dukungan Penentuan Versi.

Anda dapat menghapus objek secara permanen dengan menentukan versi yang ingin Anda hapus. Hanya pemilik bucket Amazon S3 atau pengguna IAM resmi yang dapat menghapus versi secara permanen. Jika operasi DELETE Anda menentukan `versionId`, versi objek tersebut akan dihapus secara permanen, dan Amazon S3 tidak menyisipkan penanda hapus.

Anda dapat menambahkan lebih banyak keamanan dengan mengonfigurasi bucket untuk mengaktifkan penghapusan autentikasi multi-faktor (MFA). Saat Anda mengaktifkan penghapusan MFA untuk sebuah bucket, pemilik bucket harus menyertakan dua bentuk autentikasi dalam setiap permintaan untuk menghapus sebuah versi atau mengubah status Penentuan Versi bucket. Untuk informasi selengkapnya, lihat Mengonfigurasi penghapusan MFA.

Kapan versi baru dibuat untuk suatu objek?

Versi baru objek dibuat hanya ketika Anda PUT obyek baru. Ingatlah bahwa tindakan tertentu, seperti `CopyObject`, bekerja dengan menerapkan operasi PUT.

Beberapa tindakan yang mengubah objek saat ini tidak membuat versi baru karena tindakan tersebut tidak PUT objek baru. Ini termasuk tindakan seperti mengubah tag pada objek.

> **Important**
>
> Jika Anda menyadari adanya peningkatan signifikan dalam jumlah HTTP 503 (Layanan Tidak Tersedia) yang diterima oleh objek permintaan PUT atau `DELETE` Amazon S3 ke sebuah bucket dengan Penentuan Versi S3 yang diaktifkan, mungkin Anda memiliki satu atau beberapa objek dalam bucket yang memiliki jutaan versi. Untuk informasi selengkapnya, lihat bagian Penentuan Versi S3 dari Pemecahan masalah pembuatan versi.

Mengaktifkan Penentuan Versi pada bucket

Gunakan Penentuan Versi S3 untuk menyimpan beberapa versi dari sebuah objek di dalam satu bucket. Bagian ini memberikan contoh cara mengaktifkan pembuatan versi pada bucket menggunakan konsol, REST API AWS SDKs, dan AWS Command Line Interface ()AWS CLI.

> **Note**
>
> Setelah mengaktifkan pembuatan versi pada bucket untuk pertama kalinya, mungkin diperlukan waktu hingga 15 menit agar perubahan menyebar sepenuhnya di seluruh sistem S3. Selama waktu ini, `GET` permintaan untuk objek yang dibuat atau diperbarui setelah mengaktifkan versi dapat mengakibatkan kesalahan. `HTTP 404 NoSuchKey` Sebaiknya tunggu 15 menit setelah mengaktifkan pembuatan versi sebelum melakukan operasi penulisan (`PUT`atau`DELETE`) apa pun pada objek di bucket. Masa tunggu ini membantu menghindari potensi masalah dengan visibilitas objek dan pelacakan versi.

Untuk informasi selengkapnya tentang Penentuan Versi S3, lihat Mempertahankan beberapa versi objek dengan S3 Versioning. Untuk informasi tentang cara bekerja dengan objek yang ada di bucket dengan dukungan Penentuan Versi, lihat Bekerja dengan objek di dalam bucket dengan dukungan Penentuan Versi.

Untuk mempelajari lebih lanjut cara menggunakan Penentuan Versi S3 untuk melindungi data, lihat Tutorial: Melindungi data di Amazon S3 dari penghapusan yang tidak disengaja atau bug aplikasi menggunakan Penentuan Versi S3, Kunci Objek S3, dan Replikasi S3.

Setiap bucket S3 yang Anda buat memiliki subsumber daya Penentuan Versi yang terkait dengannya. (Untuk informasi selengkapnya, lihat opsi konfigurasi bucket tujuan umum.) Secara default, bucket Anda tanpa versi, dan subsumber daya Penentuan Versi menyimpan konfigurasi Penentuan Versi yang kosong, sebagai berikut.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
</VersioningConfiguration>
```

Untuk mengaktifkan Penentuan Versi, Anda dapat mengirimkan permintaan ke Amazon S3 dengan konfigurasi Penentuan Versi yang menyertakan status.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>Enabled</Status>
</VersioningConfiguration>
```

Untuk menangguhkan Penentuan Versi, Anda menetapkan nilai status ke `Suspended`.

Pemilik bucket dan semua pengguna sah dapat mengaktifkan Penentuan Versi. Pemilik bucket adalah Akun AWS yang membuat bucket (akun root). Untuk informasi selengkapnya tentang izin, lihat Identity and Access Management untuk Amazon S3.

Bagian berikut memberikan detail lebih lanjut tentang mengaktifkan Versi S3 menggunakan konsol,, AWS CLI dan. AWS SDKs

Menggunakan konsol S3

Ikuti langkah-langkah ini untuk menggunakan fitur AWS Management Console untuk mengaktifkan pembuatan versi pada bucket S3.

Untuk mengaktifkan atau menonaktifkan pembuatan versi pada bucket tujuan umum S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Dalam daftar bucket, pilih nama bucket yang ingin Anda aktifkan pembuatan versi.
4. Pilih Properti.

5. Di bawah Penentuan Versi Bucket, pilih Edit.
6. Pilih Tangguhkan atau Aktifkan, lalu pilih Simpan perubahan.

> **Note**
>
> Anda dapat menggunakan otentikasi AWS multi-faktor (MFA) dengan pembuatan versi. Saat Anda menggunakan MFA dengan pembuatan versi, Anda harus memberikan kunci akses dan kode yang valid dari perangkat MFA akun untuk menghapus versi objek secara permanen atau menangguhkan atau mengaktifkan kembali pembuatan versi. Akun AWS Untuk menggunakan MFA dengan Penentuan Versi, Anda mengaktifkan `MFA Delete`. Namun, Anda tidak dapat mengaktifkan `MFA Delete` menggunakan AWS Management Console. Anda harus menggunakan AWS Command Line Interface (AWS CLI) atau API. Untuk informasi selengkapnya, lihat Mengonfigurasi penghapusan MFA.

Menggunakan AWS CLI

Contoh berikut memungkinkan pembuatan versi pada bucket tujuan umum S3.

```
aws s3api put-bucket-versioning --bucket amzn-s3-demo-bucket1 --versioning-
configuration Status=Enabled
```

Contoh berikut mengaktifkan penghapusan Penentuan Versi S3 dan autentikasi multi-faktor (MFA) pada bucket.

```
aws s3api put-bucket-versioning --bucket amzn-s3-demo-bucket1 --versioning-
configuration Status=Enabled,MFADelete=Enabled --mfa "SERIAL 123456"
```

> **Note**
>
> Menggunakan penghapusan MFA memerlukan perangkat autentikasi fisik atau virtual disetujui. Untuk informasi selengkapnya tentang menggunakan penghapusan MFA di Amazon S3, lihat Mengonfigurasi penghapusan MFA.

Untuk informasi selengkapnya tentang mengaktifkan versi menggunakan AWS CLI, lihat put-bucket-versioningdi Command Reference.AWS CLI

Menggunakan AWS SDKs

Contoh berikut memungkinkan pembuatan versi pada bucket dan kemudian mengambil status pembuatan versi menggunakan dan. AWS SDK untuk Java AWS SDK untuk .NET Untuk informasi tentang menggunakan yang lain AWS SDKs, lihat Pusat AWS Pengembang.

.NET

Untuk informasi tentang menyiapkan dan menjalankan contoh kode, lihat Memulai SDK for .NET di AWSAWS SDK for .NET Developer Guide.

```
using System;
using Amazon.S3;
using Amazon.S3.Model;

namespace s3.amazon.com.docsamples
{
    class BucketVersioningConfiguration
    {
        static string bucketName = "*** bucket name ***";

        public static void Main(string[] args)
        {
            using (var client = new AmazonS3Client(Amazon.RegionEndpoint.USEast1))
            {
                try
                {
                    EnableVersioningOnBucket(client);
                    string bucketVersioningStatus =
 RetrieveBucketVersioningConfiguration(client);
                }
                catch (AmazonS3Exception amazonS3Exception)
                {
                    if (amazonS3Exception.ErrorCode != null &&
                        (amazonS3Exception.ErrorCode.Equals("InvalidAccessKeyId")
                        ||
                        amazonS3Exception.ErrorCode.Equals("InvalidSecurity")))
                    {
                        Console.WriteLine("Check the provided AWS Credentials.");
                        Console.WriteLine(
```

```
                    "To sign up for service, go to http://aws.amazon.com/s3");
                }
                else
                {
                    Console.WriteLine(
                     "Error occurred. Message:'{0}' when listing objects",
                     amazonS3Exception.Message);
                }
            }
        }

        Console.WriteLine("Press any key to continue...");
        Console.ReadKey();
    }

    static void EnableVersioningOnBucket(IAmazonS3 client)
    {

            PutBucketVersioningRequest request = new PutBucketVersioningRequest
            {
                BucketName = bucketName,
                VersioningConfig = new S3BucketVersioningConfig
                {
                    Status = VersionStatus.Enabled
                }
            };

            PutBucketVersioningResponse response =
client.PutBucketVersioning(request);
    }


    static string RetrieveBucketVersioningConfiguration(IAmazonS3 client)
    {
            GetBucketVersioningRequest request = new GetBucketVersioningRequest
            {
                BucketName = bucketName
            };

            GetBucketVersioningResponse response =
client.GetBucketVersioning(request);
            return response.VersioningConfig.Status;
        }
   }
```

```
}
```

Java

Untuk petunjuk tentang cara membuat dan menguji sampel kerja, lihat Memulai di Panduan AWS SDK untuk Java Pengembang.

```
import java.io.IOException;

import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Region;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.model.AmazonS3Exception;
import com.amazonaws.services.s3.model.BucketVersioningConfiguration;
import com.amazonaws.services.s3.model.SetBucketVersioningConfigurationRequest;

public class BucketVersioningConfigurationExample {
    public static String bucketName = "*** bucket name ***";
    public static AmazonS3Client s3Client;

    public static void main(String[] args) throws IOException {
        s3Client = new AmazonS3Client(new ProfileCredentialsProvider());
        s3Client.setRegion(Region.getRegion(Regions.US_EAST_1));
        try {

            // 1. Enable versioning on the bucket.
         BucketVersioningConfiguration configuration =
           new BucketVersioningConfiguration().withStatus("Enabled");

   SetBucketVersioningConfigurationRequest setBucketVersioningConfigurationRequest
 =
     new SetBucketVersioningConfigurationRequest(bucketName,configuration);


 s3Client.setBucketVersioningConfiguration(setBucketVersioningConfigurationRequest);

   // 2. Get bucket versioning configuration information.
   BucketVersioningConfiguration conf =
 s3Client.getBucketVersioningConfiguration(bucketName);
    System.out.println("bucket versioning configuration status:    " +
 conf.getStatus());
```

```
        } catch (AmazonS3Exception amazonS3Exception) {
            System.out.format("An Amazon S3 error occurred. Exception: %s",
 amazonS3Exception.toString());
        } catch (Exception ex) {
            System.out.format("Exception: %s", ex.toString());
        }
    }
}
```

Python

Contoh kode Python berikut membuat bucket Amazon S3, mengaktifkannya untuk Penentuan Versi, dan mengonfigurasi siklus hidup yang membuat versi objek nonterkini kedaluwarsa setelah 7 hari.

```
def create_versioned_bucket(bucket_name, prefix):
    """
    Creates an Amazon S3 bucket, enables it for versioning, and configures a
 lifecycle
    that expires noncurrent object versions after 7 days.

    Adding a lifecycle configuration to a versioned bucket is a best practice.
    It helps prevent objects in the bucket from accumulating a large number of
    noncurrent versions, which can slow down request performance.

    Usage is shown in the usage_demo_single_object function at the end of this
 module.

    :param bucket_name: The name of the bucket to create.
    :param prefix: Identifies which objects are automatically expired under the
                   configured lifecycle rules.
    :return: The newly created bucket.
    """
    try:
        bucket = s3.create_bucket(
            Bucket=bucket_name,
            CreateBucketConfiguration={
                "LocationConstraint": s3.meta.client.meta.region_name
            },
        )
        logger.info("Created bucket %s.", bucket.name)
    except ClientError as error:
        if error.response["Error"]["Code"] == "BucketAlreadyOwnedByYou":
```

```
            logger.warning("Bucket %s already exists! Using it.", bucket_name)
            bucket = s3.Bucket(bucket_name)
        else:
            logger.exception("Couldn't create bucket %s.", bucket_name)
            raise

    try:
        bucket.Versioning().enable()
        logger.info("Enabled versioning on bucket %s.", bucket.name)
    except ClientError:
        logger.exception("Couldn't enable versioning on bucket %s.", bucket.name)
        raise

    try:
        expiration = 7
        bucket.LifecycleConfiguration().put(
            LifecycleConfiguration={
                "Rules": [
                    {
                        "Status": "Enabled",
                        "Prefix": prefix,
                        "NoncurrentVersionExpiration": {"NoncurrentDays":
expiration},
                    }
                ]
            }
        )
        logger.info(
            "Configured lifecycle to expire noncurrent versions after %s days "
            "on bucket %s.",
            expiration,
            bucket.name,
        )
    except ClientError as error:
        logger.warning(
            "Couldn't configure lifecycle on bucket %s because %s. "
            "Continuing anyway.",
            bucket.name,
            error,
        )

    return bucket
```

Mengonfigurasi penghapusan MFA

Saat bekerja dengan Penentuan Versi S3 di bucket Amazon S3, Anda dapat secara opsional menambahkan lapisan keamanan lainnya dengan mengonfigurasi bucket untuk mengaktifkan Penghapusan MFA (autentikasi multi-faktor). Saat melakukannya, pemilik bucket harus menyertakan dua bentuk autentikasi dalam setiap permintaan untuk menghapus sebuah versi atau mengubah status Penentuan Versi bucket.

Penghapusan MFA memerlukan autentikasi tambahan untuk salah satu dari operasi berikut:

- Mengubah status Penentuan Versi bucket Anda
- Menghapus versi objek secara permanen

Penghapusan MFA memerlukan dua bentuk autentikasi secara bersamaan:

- Kredensial keamanan Anda
- Gabungan nomor seri yang valid, spasi, dan kode enam digit yang ditampilkan di perangkat autentikasi yang disetujui

Penghapusan MFA memberikan keamanan tambahan jika, misalnya, kredensial keamanan Anda disusupi. Penghapusan MFA dapat membantu mencegah penghapusan bucket yang tidak disengaja dengan mengharuskan pengguna yang memulai tindakan penghapusan untuk membuktikan kepemilikan fisik perangkat MFA dengan kode MFA dan menambahkan lapisan gesek dan keamanan ekstra ke tindakan penghapusan.

Untuk mengidentifikasi bucket dengan penghapusan MFA yang aktif, Anda dapat menggunakan metrik Lensa Penyimpanan Amazon S3. Lensa Penyimpanan S3 adalah fitur analisis penyimpanan cloud yang dapat Anda gunakan untuk mendapatkan visibilitas seluruh organisasi ke dalam penggunaan dan aktivitas penyimpanan objek. Untuk informasi selengkapnya, lihat Menilai aktivitas penyimpanan dan penggunaan Anda dengan Lensa Penyimpanan S3. Untuk daftar lengkap metrik, lihat Glosarium metrik Lensa Penyimpanan S3.

Pemilik bucket, Akun AWS yang membuat bucket (akun root), dan semua pengguna yang berwenang dapat mengaktifkan pembuatan versi. Namun, hanya pemilik bucket (akun root) yang dapat

mengaktifkan penghapusan MFA. Untuk informasi selengkapnya, lihat Mengamankan Akses AWS Menggunakan MFA di Blog Keamanan AWS .

> **Note**
>
> Untuk menggunakan MFA dengan Penentuan Versi, Anda mengaktifkan `MFA Delete`. Namun, Anda tidak dapat mengaktifkan `MFA Delete` menggunakan AWS Management Console. Anda harus menggunakan AWS Command Line Interface (AWS CLI) atau API.
> Untuk contoh menggunakan penghapusan MFA dengan Penentuan Versi, lihat bagian contoh dalam topik Mengaktifkan Penentuan Versi pada bucket.
> Anda tidak dapat menggunakan penghapusan MFA dengan konfigurasi siklus hidup. Untuk informasi selengkapnya tentang konfigurasi siklus hidup dan caranya berinteraksi dengan konfigurasi lain, lihat Bagaimana Siklus Hidup S3 berinteraksi dengan konfigurasi bucket lainnya.

Untuk mengaktifkan atau menonaktifkan penghapusan MFA, Anda menggunakan API yang sama yang Anda gunakan untuk mengonfigurasi Penentuan Versi pada bucket. Amazon S3 menyimpan konfigurasi penghapusan MFA dalam subsumber daya Penentuan Versi yang menyimpan status Penentuan Versi bucket.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>VersioningState</Status>
  <MfaDelete>MfaDeleteState</MfaDelete>
</VersioningConfiguration>
```

Untuk menggunakan penghapusan MFA, Anda dapat menggunakan perangkat keras atau perangkat MFA virtual untuk menghasilkan kode autentikasi. Contoh berikut menunjukkan kode autentikasi yang dihasilkan yang ditampilkan pada perangkat keras.

Penghapusan MFA dan akses API yang dilindungi MFA adalah fitur yang dimaksudkan untuk memberikan perlindungan untuk skenario yang berbeda. Anda mengonfigurasi penghapusan MFA di bucket untuk membantu memastikan bahwa data di bucket Anda tidak dapat terhapus secara tidak sengaja. Akses API yang dilindungi MFA digunakan untuk memberlakukan faktor autentikasi lain (kode MFA) ketika mengakses sumber daya Amazon S3 yang sensitif. Anda dapat meminta operasi

apa pun terhadap sumber daya Amazon S3 ini dilakukan dengan kredensial sementara yang dibuat menggunakan MFA. Sebagai contoh, lihat Membutuhkan MFA.

Untuk informasi selengkapnya tentang cara membeli dan mengaktifkan perangkat autentikasi, lihat Autentikasi multi-faktor.

Untuk mengaktifkan Penentuan Versi S3 dan mengkonfigurasi penghapusan MFA

Menggunakan AWS CLI

Nomor seri adalah nomor yang secara unik mengidentifikasi perangkat MFA. Untuk perangkat MFA fisik, ini adalah nomor seri unik yang disediakan dengan perangkat. Untuk perangkat MFA virtual, nomor seri adalah perangkat ARN.

Contoh berikut mengaktifkan penghapusan Penentuan Versi S3 dan autentikasi multi-faktor (MFA) pada bucket.

```
aws s3api put-bucket-versioning --bucket amzn-s3-demo-bucket1 --versioning-
configuration Status=Enabled,MFADelete=Enabled --mfa "SERIAL 123456"
```

Penggunaan API REST

Untuk informasi selengkapnya tentang menentukan penghapusan MFA menggunakan Amazon S3 REST API, lihat Referensi API Layanan Penyimpanan Sederhana PutBucketVersioningAmazon.

Bekerja dengan objek di dalam bucket dengan dukungan Penentuan Versi

Objek yang disimpan di bucket Amazon S3 sebelum Anda mengatur status Penentuan Versi memiliki ID versi `null`. Saat Anda mengaktifkan Penentuan Versi, objek yang ada di bucket tidak berubah. Perubahannya adalah cara Amazon S3 menangani objek di masa mendatang.

Peralihan versi objek

Anda dapat menentukan aturan konfigurasi siklus hidup untuk objek yang memiliki siklus hidup yang ditetapkan dengan baik untuk transisi versi objek ke kelas penyimpanan S3 Glacier Flexible Retrieval pada waktu tertentu dalam masa aktif objek. Untuk informasi selengkapnya, lihat Mengelola siklus hidup objek.

Topik dalam bagian ini menjelaskan berbagai operasi objek dalam bucket dengan dukungan Penentuan Versi. Untuk informasi selengkapnya tentang penentuan versi, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.

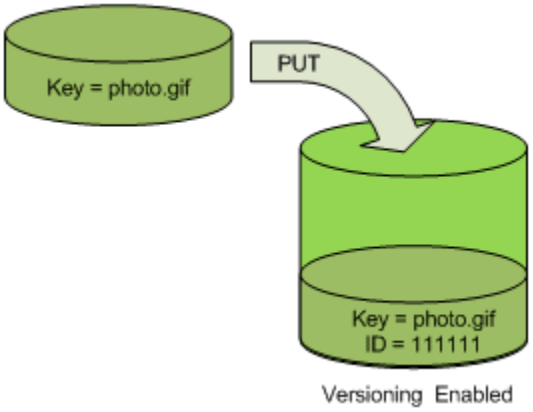
Topik

- Menambahkan objek ke bucket dengan dukungan Penentuan Versi
- Mencantumkan objek dalam bucket dengan dukungan Penentuan Versi
- Mengambil versi objek dari bucket dengan dukungan Penentuan Versi
- Menghapus versi objek dari bucket dengan dukungan Penentuan Versi
- Mengonfigurasi izin objek berversi

Menambahkan objek ke bucket dengan dukungan Penentuan Versi

Setelah Anda mengaktifkan Penentuan Versi di bucket, Amazon S3 secara otomatis menambahkan ID versi unik ke setiap objek yang disimpan (menggunakan `PUT`, `POST`, atau `CopyObject`) di bucket.

Gambar berikut menunjukkan bahwa Amazon S3 menambahkan ID versi unik ke objek saat ditambahkan ke bucket dengan dukungan Penentuan Versi.



> **Note**
>
> Nilai ID versi yang ditetapkan Amazon S3 adalah URL aman (dapat disertakan sebagai bagian dari URI).

Untuk informasi selengkapnya tentang penentuan versi, lihat Mempertahankan beberapa versi objek dengan S3 Versioning. Anda dapat menambahkan versi objek ke bucket berkemampuan versi menggunakan konsol AWS SDKs, dan REST API.

Menggunakan konsol

Untuk petunjuk, lihat Mengunggah Objek.

Menggunakan AWS SDKs

Untuk contoh mengunggah objek menggunakan AWS SDKs untuk Java, .NET, dan PHP, lihatMengunggah Objek. Contoh untuk mengunggah objek dalam bucket tanpa Penentuan Versi dan dengan Penentuan Versi yang diaktifkan adalah sama, meskipun dalam kasus bucket dengan dukungan Penentuan Versi, Amazon S3 menetapkan nomor versi. Jika tidak, nomor versi akan null.

Untuk informasi tentang menggunakan yang lain AWS SDKs, lihat Pusat AWS Pengembang.

Penggunaan API REST

Untuk menambahkan objek ke bucket dengan dukungan Penentuan Versi

1. Aktifkan Penentuan Versi di bucket menggunakan permintaan `PutBucketVersioning`.

   Untuk informasi selengkapnya, lihat PutBucketVersioning dalam Referensi API Amazon Simple Storage Service.
2. Kirim permintaan `PUT`, `POST`, atau `CopyObject` untuk menyimpan objek dalam bucket.

Saat Anda menambahkan objek ke bucket dengan dukungan Penentuan Versi, Amazon S3 mengembalikan ID versi objek di header respons `x-amz-version-id`, seperti yang ditunjukkan dalam contoh berikut.

```
x-amz-version-id: 3/L4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY
```

Mencantumkan objek dalam bucket dengan dukungan Penentuan Versi

Bagian ini memberikan contoh daftar versi objek dari bucket dengan dukungan Penentuan Versi. Amazon S3 menyimpan informasi versi objek dalam subsumber daya versi yang dikaitkan dengan bucket. Untuk informasi selengkapnya, lihat opsi konfigurasi bucket tujuan umum. Untuk mencantumkan objek dalam bucket dengan dukungan Penentuan Versi, Anda memerlukan izin `ListBucketVersions`.

Menggunakan konsol S3

Ikuti langkah-langkah ini untuk menggunakan konsol Amazon S3 guna melihat berbagai versi objek.

Untuk melihat beberapa versi objek

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di dalam daftar Bucket, pilih nama bucket yang berisi objek.
3. Untuk melihat daftar versi objek di bucket, pilih tombol Tampilkan versi.

   Untuk setiap versi objek, konsol menunjukkan ID versi unik, tanggal dan waktu versi objek dibuat, dan properti lainnya. (Objek yang disimpan di bucket sebelum Anda mengatur status Penentuan Versi memiliki ID versi null.)

   Untuk mencantumkan objek tanpa versi, pilih tombol Buat daftar versi.

Anda juga dapat melihat, mengunduh, dan menghapus versi objek di panel ikhtisar objek di konsol. Untuk informasi selengkapnya, lihat Melihat properti objek di konsol Amazon S3.

> **Note**
>
> Untuk mengakses versi objek yang lebih lama dari 300 versi, Anda harus menggunakan AWS CLI atau URL objek.

> **Important**
>
> Anda dapat membatalkan penghapusan objek hanya jika objek tersebut dihapus sebagai versi terbaru (saat ini). Anda tidak dapat membatalkan penghapusan objek versi sebelumnya. Untuk informasi selengkapnya, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.

Menggunakan AWS SDKs

Contoh dalam bagian ini menunjukkan cara mengambil daftar objek dari bucket dengan dukungan Penentuan Versi. Setiap permintaan akan menampilkan hingga 1.000 versi, kecuali jika Anda menentukan nomor yang lebih rendah. Jika bucket berisi lebih banyak versi dari batas ini, Anda

mengirimkan serangkaian permintaan untuk mengambil daftar semua versi. Proses mengembalikan hasil dalam "halaman" disebut penomoran halaman.

Untuk menunjukkan cara kerja penomoran halaman, contoh membatasi masing-masing respons pada dua versi objek. Setelah mengambil halaman pertama dari hasil, masing-masing contoh memeriksa untuk menentukan apakah daftar versi terpotong. Jika iya, contoh ini terus mengambil halaman hingga semua versi diambil.

> **Note**
>
> Contoh berikut juga bekerja dengan bucket tanpa Penentuan Versi aktif, atau untuk objek yang tidak memiliki versi tersendiri. Dalam kasus tersebut, Amazon S3 mengembalikan daftar objek dengan ID versi `null`.

Untuk informasi tentang menggunakan yang lain AWS SDKs, lihat Pusat AWS Pengembang.

Java

Untuk petunjuk cara membuat dan menguji sampel yang berfungsi, lihat Memulai di Panduan AWS SDK untuk Java Pengembang.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ListVersionsRequest;
import com.amazonaws.services.s3.model.S3VersionSummary;
import com.amazonaws.services.s3.model.VersionListing;

public class ListKeysVersioningEnabledBucket {

    public static void main(String[] args) {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
```

```
                    .withRegion(clientRegion)
                    .build();

            // Retrieve the list of versions. If the bucket contains more versions
            // than the specified maximum number of results, Amazon S3 returns
            // one page of results per request.
            ListVersionsRequest request = new ListVersionsRequest()
                    .withBucketName(bucketName)
                    .withMaxResults(2);
            VersionListing versionListing = s3Client.listVersions(request);
            int numVersions = 0, numPages = 0;
            while (true) {
                numPages++;
                for (S3VersionSummary objectSummary :
 versionListing.getVersionSummaries()) {
                    System.out.printf("Retrieved object %s, version %s\n",
                            objectSummary.getKey(),
                            objectSummary.getVersionId());
                    numVersions++;
                }
                // Check whether there are more pages of versions to retrieve. If
                // there are, retrieve them. Otherwise, exit the loop.
                if (versionListing.isTruncated()) {
                    versionListing =
 s3Client.listNextBatchOfVersions(versionListing);
                } else {
                    break;
                }
            }
            System.out.println(numVersions + " object versions retrieved in " +
 numPages + " pages");
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

Untuk informasi tentang menyiapkan dan menjalankan contoh kode, lihat Memulai SDK for .NET di AWSAWS SDK for .NET Developer Guide.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class ListObjectsVersioningEnabledBucketTest
    {
        static string bucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main(string[] args)
        {
            s3Client = new AmazonS3Client(bucketRegion);
            GetObjectListWithAllVersionsAsync().Wait();
        }

        static async Task GetObjectListWithAllVersionsAsync()
        {
            try
            {
                ListVersionsRequest request = new ListVersionsRequest()
                {
                    BucketName = bucketName,
                    // You can optionally specify key name prefix in the request
                    // if you want list of object versions of a specific object.

                    // For this example we limit response to return list of 2
 versions.
                    MaxKeys = 2
                };
                do
                {
```

```
                    ListVersionsResponse response = await
 s3Client.ListVersionsAsync(request);
                    // Process response.
                    foreach (S3ObjectVersion entry in response.Versions)
                    {
                        Console.WriteLine("key = {0} size = {1}",
                            entry.Key, entry.Size);
                    }

                    // If response is truncated, set the marker to get the next
                    // set of keys.
                    if (response.IsTruncated)
                    {
                        request.KeyMarker = response.NextKeyMarker;
                        request.VersionIdMarker = response.NextVersionIdMarker;
                    }
                    else
                    {
                        request = null;
                    }
                } while (request != null);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
        }
    }
}
```

Penggunaan API REST

Example — Mencantumkan semua versi objek dalam bucket

Untuk mencantumkan semua versi dari objek dalam bucket, Anda menggunakan subsumber daya `versions` di permintaan `GET Bucket`. Amazon S3 dapat mengambil maksimum 1.000 objek, dan

setiap versi objek dihitung sepenuhnya sebagai objek. Oleh karena itu, jika bucket berisi dua kunci (misalnya, `photo.gif` dan `picture.jpg`), dan kunci pertama memiliki 990 versi dan kunci kedua memiliki 400 versi, satu permintaan akan mengambil seluruh 990 versi `photo.gif` dan hanya 10 versi terbaru dari `picture.jpg`.

Amazon S3 mengembalikan versi objek dalam urutan penyimpanannya, dengan versi yang disimpan paling akhir dikembalikan terlebih dahulu.

Di permintaan `GET Bucket`, sertakan subsumber daya `versions`.

```
GET /?versions HTTP/1.1
Host: bucketName.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 +0000
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Example — Mengambil semua versi kunci

Untuk mengambil subset versi objek, Anda menggunakan parameter permintaan untuk `GET Bucket`. Untuk informasi selengkapnya, lihat GET Bucket.

1. Atur parameter `prefix` ke kunci objek yang ingin Anda ambil.
2. Kirim permintaan `GET Bucket` menggunakan subsumber daya `versions` dan `prefix`.

   ```
   GET /?versions&prefix=objectName HTTP/1.1
   ```

Example — Mengambil objek menggunakan prefiks

Contoh berikut mengambil objek yang kuncinya adalah atau dimulai dengan `myObject`.

```
GET /?versions&prefix=myObject HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Anda dapat menggunakan parameter permintaan lain untuk mengambil subset semua versi objek. Untuk informasi selengkapnya, lihat GET Bucket dalam Referensi API Amazon Simple Storage Service.

Example — Mengambil daftar objek tambahan jika respons terpotong

Jika jumlah objek yang dapat dikembalikan dalam permintaan GET melebihi nilai `max-keys`, respons mengandung `<isTruncated>true</isTruncated>`, dan menyertakan kunci pertama (dalam `NextKeyMarker`) dan ID versi pertama (dalam `NextVersionIdMarker`) yang memenuhi permintaan, tetapi tidak dikembalikan. Anda menggunakan nilai yang dikembalikan sebagai posisi awal pada permintaan berikutnya untuk mengambil objek tambahan yang memenuhi permintaan GET.

Gunakan proses berikut untuk mengambil objek tambahan yang sesuai dengan permintaan GET `Bucket versions` dari bucket. Untuk informasi selengkapnya tentang `key-marker`, `version-id-marker`, `NextKeyMarker`, dan `NextVersionIdMarker`, lihat GET Bucket di Referensi API Amazon Simple Storage Service.

Berikut ini adalah tanggapan tambahan yang memenuhi permintaan GET asli:

- Tetapkan nilai dari `key-marker` ke kunci yang dikembalikan di `NextKeyMarker` dalam respons sebelumnya.
- Tetapkan nilai dari `version-id-marker` ke ID versi yang dikembalikan di `NextVersionIdMarker` dalam respons sebelumnya.
- Kirim permintaan GET `Bucket versions` menggunakan `key-marker` dan `version-id-marker`.

Example — Mengambil objek yang dimulai dengan kunci dan ID versi tertentu

```
GET /?versions&key-marker=myObject&version-id-marker=298459348571 HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Menggunakan AWS CLI

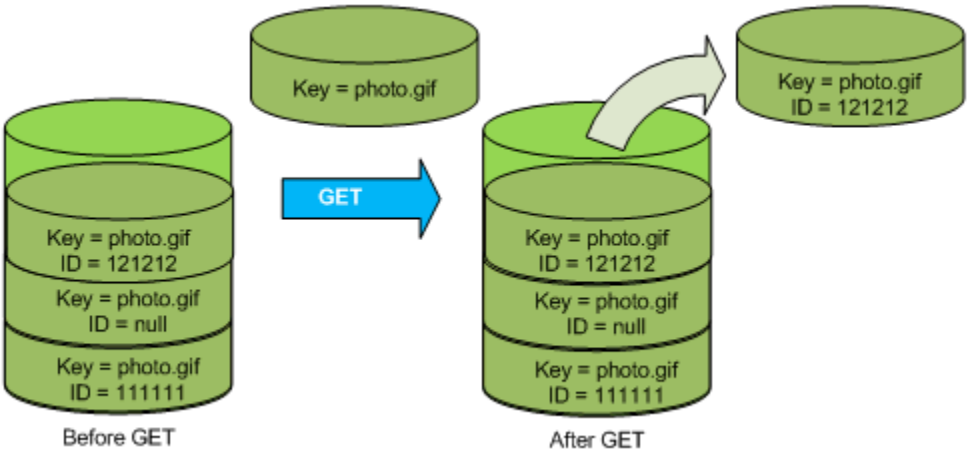
Perintah berikut mengembalikan metadata tentang semua versi objek dalam bucket.

```
aws s3api list-object-versions --bucket amzn-s3-demo-bucket1
```

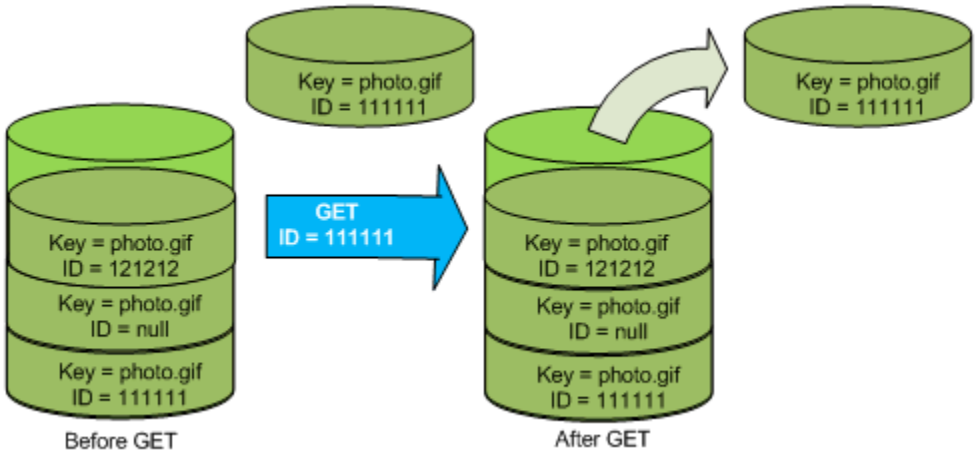
Untuk informasi selengkapnya tentang `list-object-versions`, lihat list-object-versions di Referensi Perintah AWS CLI .

Mengambil versi objek dari bucket dengan dukungan Penentuan Versi

Penentuan Versi di Amazon S3 adalah cara menyimpan beberapa varian objek dalam bucket yang sama. Permintaan `GET` sederhana mengambil versi saat ini dari sebuah objek. Gambar berikut menunjukkan bagaimana `GET` mengembalikan versi objek saat ini, `photo.gif`.



Untuk mengambil versi tertentu, Anda harus menentukan ID versinya. Gambar berikut menunjukkan bahwa permintaan `GET versionId` mengambil versi objek yang ditentukan (tidak harus versi saat ini).



Anda dapat mengambil versi objek di Amazon S3 menggunakan konsol AWS SDKs,, atau REST API.

> **Note**
>
> Untuk mengakses versi objek yang lebih lama dari 300 versi, Anda harus menggunakan AWS CLI atau URL objek.

Menggunakan konsol S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di dalam daftar Bucket, pilih nama bucket yang berisi objek.
3. Di daftar Objek, pilih nama objek.
4. Pilih Versi.

   Amazon S3 menampilkan semua versi untuk objek tersebut.
5. Pilih kotak centang di sebelah ID Versi untuk versi yang ingin Anda ambil.
6. Pilih Tindakan, pilih Unduh, dan simpan objek.

Anda juga dapat melihat, mengunduh, dan menghapus versi objek di panel ikhtisar objek. Untuk informasi selengkapnya, lihat Melihat properti objek di konsol Amazon S3.

> **Important**
>
> Anda dapat membatalkan penghapusan objek hanya jika objek tersebut dihapus sebagai versi terbaru (saat ini). Anda tidak dapat membatalkan penghapusan objek versi sebelumnya. Untuk informasi selengkapnya, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.

Menggunakan AWS SDKs

Contoh untuk pengunggahan objek dalam bucket tanpa versi dan dengan dukungan Penentuan Versi adalah sama. Namun, untuk bucket dengan dukungan Penentuan Versi, Amazon S3 menetapkan nomor versi. Jika tidak, nomor versi adalah null.

Untuk contoh mengunduh objek yang menggunakan AWS SDKs Java, .NET, dan PHP, lihat Mengunduh objek.

Untuk contoh mencantumkan versi objek yang digunakan AWS SDKs untuk.NET dan Rust, lihat Daftar versi objek dalam bucket Amazon S3.

Penggunaan API REST

Untuk mengambil versi objek tertentu

1. Atur `versionId` ke ID versi objek yang ingin Anda ambil.
2. Kirim permintaan `GET Object versionId`.

Example — Mengambil objek berversi

Permintaan berikut mengambil versi `L4kqtJlcpXroDTDmpUMLUo` dari `my-image.jpg`.

```
GET /my-image.jpg?versionId=L4kqtJlcpXroDTDmpUMLUo HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Anda hanya dapat mengambil metadata dari suatu objek (bukan konten). Untuk informasi, lihat the section called “Mengambil metadata versi”.

Untuk informasi tentang memulihkan versi objek sebelumnya, lihat the section called “Memulihkan versi sebelumnya”.

Mengambil metadata versi objek

Jika Anda hanya ingin mengambil metadata suatu objek (dan bukan kontennya), Anda menggunakan operasi `HEAD`. Secara default, Anda mendapatkan metadata versi terbaru. Untuk mengambil metadata versi objek tertentu, Anda menentukan ID versinya.

Untuk mengambil metadata versi objek

1. Atur `versionId` ke ID versi objek yang ingin Anda ambil metadatanya.
2. Kirim permintaan `HEAD Object versionId`.

Example — Mengambil metadata dari objek berversi

Permintaan berikut mengambil metadata versi `3HL4kqCxf3vjVBH40Nrjfkd` dari `my-image.jpg`.

```
HEAD /my-image.jpg?versionId=3HL4kqCxf3vjVBH40Nrjfkd HTTP/1.1
```

```
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Berikut ini menunjukkan respons sampel.

```
HTTP/1.1 200 OK
x-amz-id-2: ef8yU9AS1ed4OpIszj7UDNEHGran
x-amz-request-id: 318BC8BC143432E5
x-amz-version-id: 3HL4kqtJlcpXroDTDmjVBH40Nrjfkd
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2006 12:00:00 GMT
ETag: "fba9dede5f27731c9771645a39863328"
Content-Length: 434234
Content-Type: text/plain
Connection: close
Server: AmazonS3
```

Memulihkan versi sebelumnya

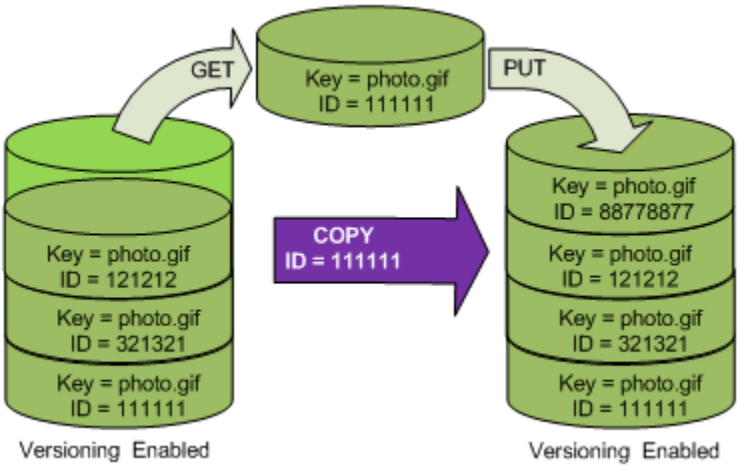
Anda dapat menggunakan Penentuan Versi untuk mengambil versi sebelumnya dari objek. Ada dua pendekatan untuk melakukannya:

- Salin versi objek sebelumnya ke dalam bucket yang sama.

  Objek yang disalin menjadi versi saat ini dari objek tersebut dan semua versi objek dipertahankan.
- Hapus versi objek saat ini secara permanen.

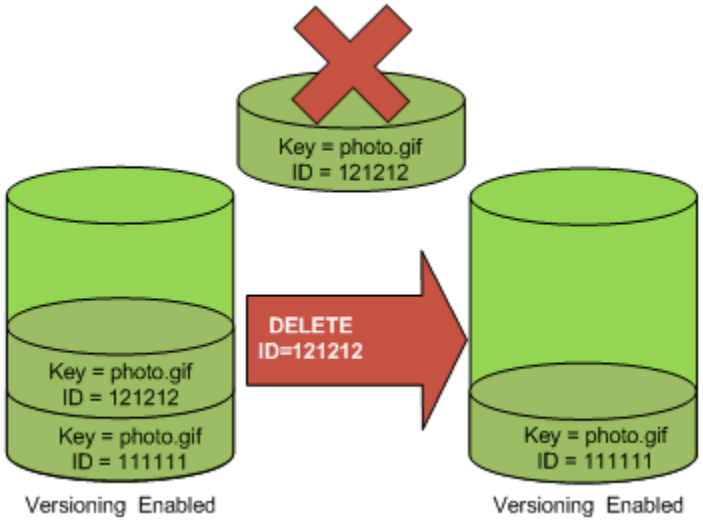
  Ketika versi objek saat ini dihapus, Anda, akibatnya, mengubah versi sebelumnya menjadi versi saat ini dari objek tersebut.

Karena semua versi objek dipertahankan, Anda dapat membuat versi yang lebih awal dari versi saat ini dengan menyalin versi tertentu dari objek tersebut ke dalam bucket yang sama. Pada gambar berikut, objek sumber (ID = 111111) disalin ke dalam bucket yang sama. Amazon S3 memasok ID baru (88778877) dan menjadi versi objek saat ini. Jadi, bucket memiliki objek versi asli (111111) dan salinannya (88778877). Untuk informasi selengkapnya tentang mendapatkan versi sebelumnya dan kemudian mengunggahnya untuk menjadikannya versi saat ini, lihat Mengambil versi objek dari bucket dengan dukungan Penentuan Versi dan Mengunggah objek.



GET berikutnya mengambil versi 88778877.

Gambar berikut menunjukkan bagaimana menghapus versi saat ini (121212) dari suatu objek menjadikan versi sebelumnya (111111) sebagai objek saat ini. Untuk informasi selengkapnya tentang menghapus objek, lihat Menghapus satu objek.



GET berikutnya mengambil versi 111111.

> **Note**
>
> Untuk mengembalikan versi objek dalam batch, Anda dapat menggunakan operasi `CopyObject`. Operasi salin dapat menyalin setiap objek yang ditentukan dalam manifes. Namun, ketahuilah bahwa objek tidak selalu disalin dalam urutan yang sama seperti yang tercantum dalam manifes. Untuk bucket berversi, jika mempertahankan urutan versi terkini/

nonterkini bersifat penting, Anda harus menyalin semua versi nonterkini terlebih dahulu. Kemudian, setelah pekerjaan pertama selesai, salin versi terbaru pada pekerjaan berikutnya.

Untuk mengembalikan versi objek sebelumnya

Untuk panduan selengkapnya tentang memulihkan objek yang dihapus, lihat Bagaimana cara mengambil objek Amazon S3 yang telah dihapus dalam bucket berkemampuan versi? di pusat AWS re:Post pengetahuan.

Menggunakan konsol S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di dalam daftar Bucket, pilih nama bucket yang berisi objek.
3. Di daftar Objek, pilih nama objek.
4. Pilih Versi.

   Amazon S3 menampilkan semua versi untuk objek tersebut.
5. Pilih kotak centang di sebelah ID Versi untuk versi yang ingin Anda ambil.
6. Pilih Tindakan, pilih Unduh, dan simpan objek.

Anda juga dapat melihat, mengunduh, dan menghapus versi objek di panel ikhtisar objek. Untuk informasi selengkapnya, lihat Melihat properti objek di konsol Amazon S3.

⚠ Important

Anda dapat membatalkan penghapusan objek hanya jika objek tersebut dihapus sebagai versi terbaru (saat ini). Anda tidak dapat membatalkan penghapusan objek versi sebelumnya. Untuk informasi selengkapnya, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.

Menggunakan AWS SDKs

Untuk informasi tentang menggunakan yang lain AWS SDKs, lihat Pusat AWS Pengembang.

Python

Contoh kode Python berikut memulihkan versi objek berversi sebelumnya dengan menghapus semua versi yang terjadi setelah versi rollback yang ditentukan.

```
def rollback_object(bucket, object_key, version_id):
    """
    Rolls back an object to an earlier version by deleting all versions that
    occurred after the specified rollback version.

    Usage is shown in the usage_demo_single_object function at the end of this
 module.

    :param bucket: The bucket that holds the object to roll back.
    :param object_key: The object to roll back.
    :param version_id: The version ID to roll back to.
    """
    # Versions must be sorted by last_modified date because delete markers are
    # at the end of the list even when they are interspersed in time.
    versions = sorted(
        bucket.object_versions.filter(Prefix=object_key),
        key=attrgetter("last_modified"),
        reverse=True,
    )

    logger.debug(
        "Got versions:\n%s",
        "\n".join(
            [
                f"\t{version.version_id}, last modified {version.last_modified}"
                for version in versions
            ]
        ),
    )

    if version_id in [ver.version_id for ver in versions]:
        print(f"Rolling back to version {version_id}")
        for version in versions:
            if version.version_id != version_id:
                version.delete()
                print(f"Deleted version {version.version_id}")
            else:
                break
```

```
        print(f"Active version is now {bucket.Object(object_key).version_id}")
    else:
        raise KeyError(
            f"{version_id} was not found in the list of versions for "
f"{object_key}."
        )
```

Menghapus versi objek dari bucket dengan dukungan Penentuan Versi

Anda dapat menghapus versi objek dari bucket Amazon S3 kapan pun Anda inginkan. Selain itu, Anda juga dapat menentukan aturan konfigurasi siklus hidup untuk objek yang memiliki siklus hidup yang ditentukan dengan baik guna meminta Amazon S3 untuk menghentikan versi objek saat ini atau menghapus versi objek nonterkini secara permanen. Jika bucket Anda mengaktifkan versi atau pembuatan versi ditangguhkan, tindakan konfigurasi siklus hidup berfungsi sebagai berikut:

- Tindakan `Expiration` berlaku untuk versi objek saat ini. Alih-alih menghapus versi objek saat ini, Amazon S3 mempertahankan versi saat ini sebagai versi bukan saat ini dengan menambahkan penanda hapus, yang kemudian menjadi versi saat ini.
- Tindakan `NoncurrentVersionExpiration` tersebut berlaku untuk versi objek nonterkini, dan Amazon S3 secara permanen menghapus versi objek ini. Anda tidak dapat memulihkan kembali objek yang dihapus secara permanen.

Untuk informasi selengkapnya tentang Siklus Hidup S3, lihat Mengelola siklus hidup objek dan Contoh konfigurasi Siklus Hidup S3.

Untuk melihat berapa banyak versi objek saat ini dan bukan saat ini yang dimiliki bucket, Anda dapat menggunakan metrik Lensa Penyimpanan Amazon S3. Lensa Penyimpanan S3 adalah fitur analisis penyimpanan cloud yang dapat Anda gunakan untuk mendapatkan visibilitas seluruh organisasi ke dalam penggunaan dan aktivitas penyimpanan objek. Untuk informasi selengkapnya, lihat Menggunakan Lensa Penyimpanan S3 untuk mengoptimalkan biaya penyimpanan Anda. Untuk daftar lengkap metrik, lihat glosarium metrik Lensa Penyimpanan S3.

> **Note**
>
> Tarif Amazon S3 normal berlaku untuk setiap versi objek yang disimpan dan ditransfer, termasuk versi objek noncurrent. Untuk informasi selengkapnya, lihat Harga Amazon S3.

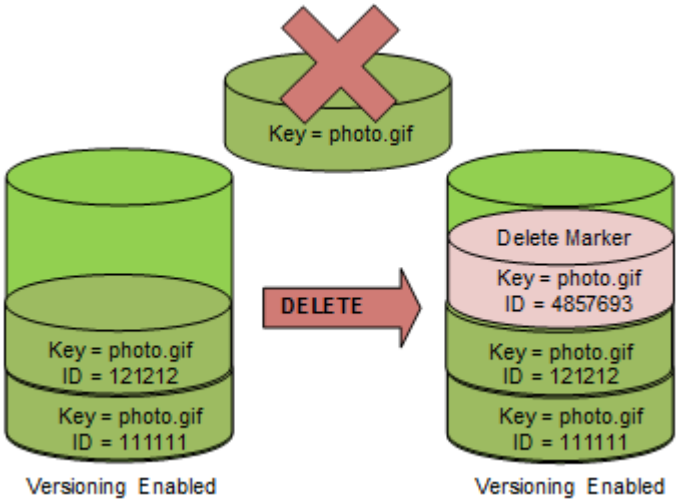
Menghapus kasus penggunaan permintaan

Permintaan `DELETE` memiliki kasus penggunaan berikut:

- Saat Penentuan Versi diaktifkan, `DELETE` sederhana tidak dapat menghapus objek secara permanen. (Permintaan `DELETE` sederhana adalah permintaan yang tidak menentukan ID versi.) Alih-alih demikian, Amazon S3 memasukkan penanda hapus dalam bucket, dan marker tersebut menjadi versi objek saat ini dengan ID baru.

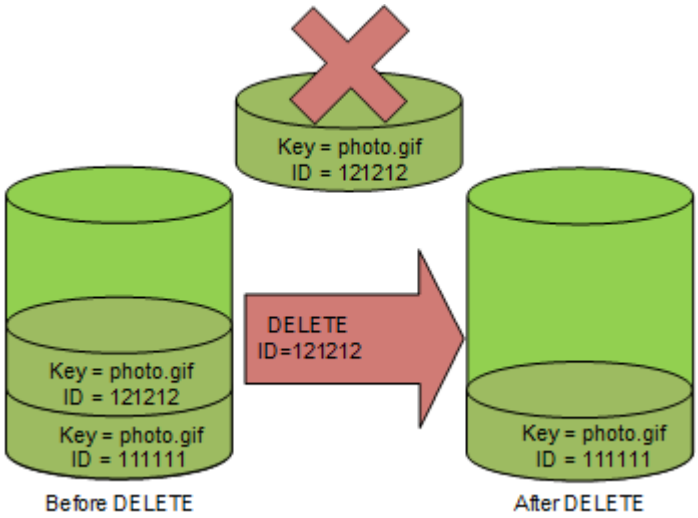
  Saat Anda mencoba untuk `GET` objek yang versinya saat ini adalah penanda hapus, Amazon S3 berperilaku seolah-olah objek telah dihapus (meskipun belum dihapus) dan mengembalikan pesan kesalahan 404. Untuk informasi selengkapnya, lihat Bekerja dengan penanda hapus.

  Gambar berikut menunjukkan bahwa `DELETE` sederhana tidak benar-benar menghapus objek tertentu. Sebaliknya, Amazon S3 menyisipkan penanda hapus.



- Untuk menghapus objek dengan Penentuan Versi secara permanen, Anda harus menggunakan `DELETE Object versionId`.

  Gambar berikut menunjukkan bahwa menghapus versi objek tertentu secara permanen menghapus objek tersebut.



Untuk menghapus versi objek

Anda dapat menghapus versi objek di Amazon S3 menggunakan konsol AWS SDKs, REST API, atau. AWS Command Line Interface

Menggunakan konsol S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di dalam daftar Bucket, pilih nama bucket yang berisi objek.
3. Di daftar Objek, pilih nama objek.
4. Pilih Versi.

   Amazon S3 menampilkan semua versi untuk objek tersebut.
5. Pilih kotak centang di samping ID Versi untuk versi yang ingin Anda hapus secara permanen.
6. Pilih Hapus.
7. Di Hapus objek secara permanen?, masukkan **`permanently delete`**.

   > ⚠ Warning
   >
   > Saat Anda menghapus versi objek secara permanen, tindakan itu tidak dapat dibatalkan.
8. Pilih Hapus objek.

   Amazon S3 menghapus versi objek.

Menggunakan AWS SDKs

Untuk contoh menghapus objek menggunakan AWS SDKs untuk Java, .NET, dan PHP, lihatMenghapus objek Amazon S3. Contoh untuk menghapus objek dalam bucket tanpa versi dan dengan dukungan Penentuan Versi adalah sama. Namun, untuk bucket dengan dukungan Penentuan Versi, Amazon S3 menetapkan nomor versi. Jika tidak, nomor versi adalah null.

Untuk informasi tentang menggunakan yang lain AWS SDKs, lihat Pusat AWS Pengembang.

Python

Contoh kode Python berikut secara permanen menghapus objek berversi dengan menghapus semua versinya.

```
def permanently_delete_object(bucket, object_key):
    """
    Permanently deletes a versioned object by deleting all of its versions.

    Usage is shown in the usage_demo_single_object function at the end of this
 module.

    :param bucket: The bucket that contains the object.
    :param object_key: The object to delete.
    """
    try:
        bucket.object_versions.filter(Prefix=object_key).delete()
        logger.info("Permanently deleted all versions of object %s.", object_key)
    except ClientError:
        logger.exception("Couldn't delete all versions of %s.", object_key)
        raise
```

Penggunaan API REST

Untuk menghapus versi objek tertentu

- Di `DELETE`, tentukan ID versi.

Example — Menghapus versi tertentu

Contoh berikut menghapus versi `UIORUnfnd89493jJFJ` dari `photo.gif`.

```
DELETE /photo.gif?versionId=UIORUnfnd89493jJFJ HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:xQE0diMbLRepdf3YB+FIEXAMPLE=
Content-Type: text/plain
Content-Length: 0
```

Menggunakan AWS CLI

Perintah berikut menghapus objek bernama test.txt dari bucket bernama *amzn-s3-demo-bucket1*. Untuk menghapus versi objek tertentu, Anda harus menjadi pemilik bucket dan harus menggunakan subsumber daya Id versi.

```
aws s3api delete-object --bucket amzn-s3-demo-bucket1 --key test.txt --version-
id versionID
```

Untuk informasi selengkapnya tentang `delete-object`, lihat delete-object di Referensi Perintah AWS CLI .

Untuk informasi selengkapnya tentang menghapus versi objek, lihat topik berikut ini:

- Bekerja dengan penanda hapus
- Menghapus penanda hapus untuk membuat versi yang lebih lama menjadi versi saat ini
- Menghapus objek dari bucket dengan dukungan penghapusan MFA

Bekerja dengan penanda hapus

Penanda hapus di Amazon S3 adalah placeholder (atau penanda) untuk objek berversi yang ditentukan dalam permintaan `DELETE` sederhana. Permintaan `DELETE` sederhana adalah permintaan yang tidak menentukan ID versi. Karena objek berada dalam bucket dengan dukungan Penentuan Versi, objek tidak dihapus. Namun, penanda hapus membuat Amazon S3 berperilaku seolah-olah objek telah dihapus. Anda dapat menggunakan panggilan `DELETE` API Amazon S3 pada penanda hapus. Untuk melakukan ini, Anda harus membuat `DELETE` permintaan dengan menggunakan pengguna AWS Identity and Access Management (IAM) atau peran dengan izin yang sesuai.

Penanda hapus memiliki nama kunci (atau kunci) dan ID versi seperti objek lainnya. Namun, penanda hapus berbeda dengan objek lain dalam hal berikut:

- Penanda hapus tidak memiliki data yang terkait dengannya.
- Penanda penghapusan tidak terkait dengan nilai daftar kontrol akses (ACL).
- Jika Anda mengeluarkan permintaan `GET` untuk penanda hapus, permintaan `GET` tidak mengambil apa pun karena penanda hapus tidak memiliki data. Khususnya, ketika permintaan `GET` Anda tidak menentukan `versionId`, Anda mendapatkan pesan kesalahan 404 (Tidak Ditemukan).

Penanda hapus menambah sedikit biaya penyimpanan di Amazon S3. Ukuran penyimpanan penanda hapus sama dengan ukuran nama kunci dari penanda hapus. Nama kunci adalah deretan karakter Unicode. Pengodean UTF-8 untuk nama kunci menambahkan 1-4 byte penyimpanan ke bucket Anda untuk setiap karakter dalam nama. Penanda hapus disimpan di kelas penyimpanan S3 Standard.

Jika ingin mengetahui berapa banyak penanda hapus yang Anda miliki dan kelas penyimpanan apa yang disimpan, Anda dapat menggunakan Lensa Penyimpanan Amazon S3. Untuk informasi selengkapnya, lihat Menilai aktivitas penyimpanan dan penggunaan dengan Lensa Penyimpanan Amazon S3 dan Glosarium metrik Lensa Penyimpanan Amazon S3.

Untuk informasi selengkapnya tentang nama kunci, lihat Menamai objek Amazon S3. Untuk informasi tentang menghapus penanda hapus, lihat Mengelola penanda hapus.

Hanya Amazon S3 yang dapat membuat penanda hapus, dan hal ini dilakukan setiap kali Anda mengirim permintaan `DeleteObject` atas objek dalam bucket dengan dukungan Penentuan Versi atau Penentuan Versi ditangguhkan. Objek yang ditentukan dalam permintaan `DELETE` tidak benar-benar dihapus. Alih-alih demikian, penanda hapus menjadi versi objek saat ini. Nama kunci objek (atau kunci) menjadi kunci dari penanda hapus.

Saat Anda mendapatkan objek tanpa menentukan `versionId` dalam permintaan, jika versinya saat ini adalah penanda hapus, Amazon S3 akan merespons dengan berikut:

- Pesan kesalahan 404 (Tidak Ditemukan)
- Header respons, `x-amz-delete-marker: true`

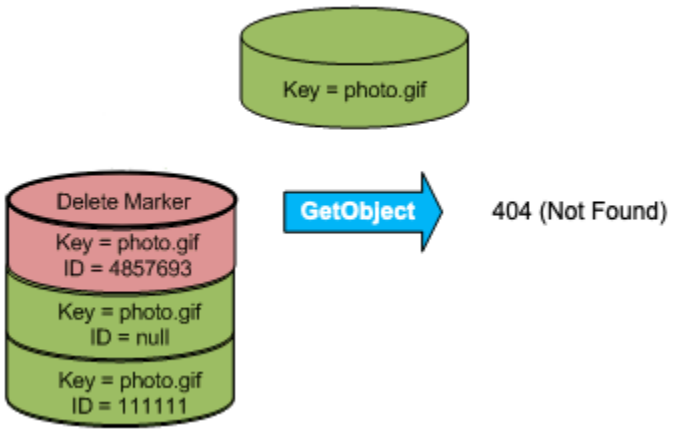
Saat Anda mendapatkan objek dengan menentukan `versionId` dalam permintaan, jika versi yang ditentukan adalah penanda hapus, Amazon S3 akan merespons dengan berikut:

- Pesan kesalahan 405 (Metode Tidak Diizinkan)
- Header respons, `x-amz-delete-marker: true`
- Header respons, `Last-Modified: timestamp` (hanya saat menggunakan operasi HeadObjectatau GetObjectAPI)

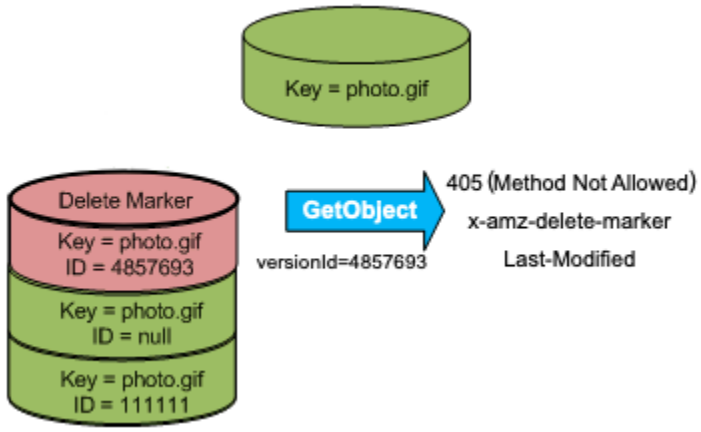
Header respons `x-amz-delete-marker: true` memberi tahu bahwa objek yang diakses adalah penanda hapus. Header respons ini tidak pernah mengembalikan `false`, karena ketika nilainya adalah `false`, versi objek saat ini atau yang ditentukan bukanlah penanda hapus.

Header respons `Last-Modified` menyediakan waktu pembuatan penanda hapus.

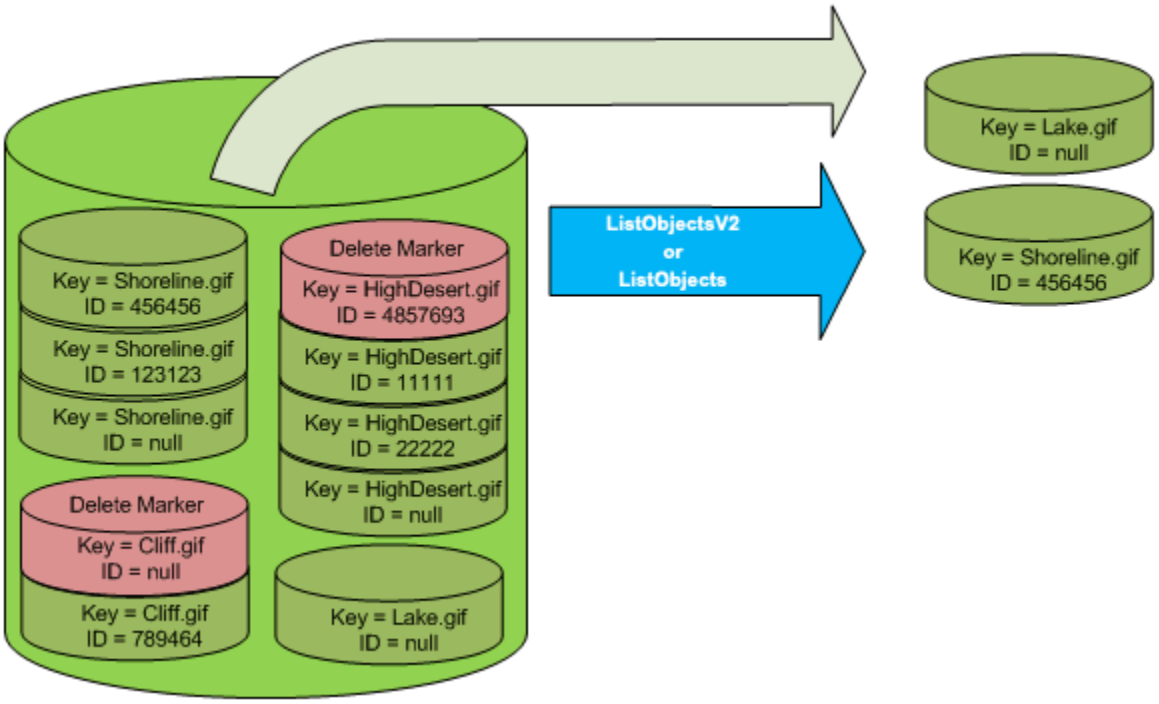
Gambar berikut menunjukkan cara panggilan API `GetObject` pada sebuah objek yang versinya saat ini adalah penanda hapus merespons dengan pesan kesalahan 404 (Tidak Ditemukan) dan header respons menyertakan `x-amz-delete-marker: true`.



Jika Anda melakukan panggilan `GetObject` pada objek dengan menentukan `versionId` dalam permintaan Anda, dan jika versi yang ditentukan adalah penanda hapus, Amazon S3 merespons dengan pesan kesalahan 405 (Metode Tidak Diizinkan) dan header respons menyertakan `x-amz-delete-marker: true` dan `Last-Modified: timestamp`.



Satu-satunya cara untuk mencantumkan penanda hapus (dan versi lain dari objek) adalah dengan menggunakan subsumber daya `versions` di permintaan ListObjectVersions. Gambar berikut menunjukkan bahwa permintaan ListObjectsV2 atau ListObjects tidak mengembalikan objek yang versinya saat ini adalah penanda hapus.



Mengelola penanda hapus

Mengonfigurasi siklus hidup untuk membersihkan penanda hapus kedaluwarsa secara otomatis

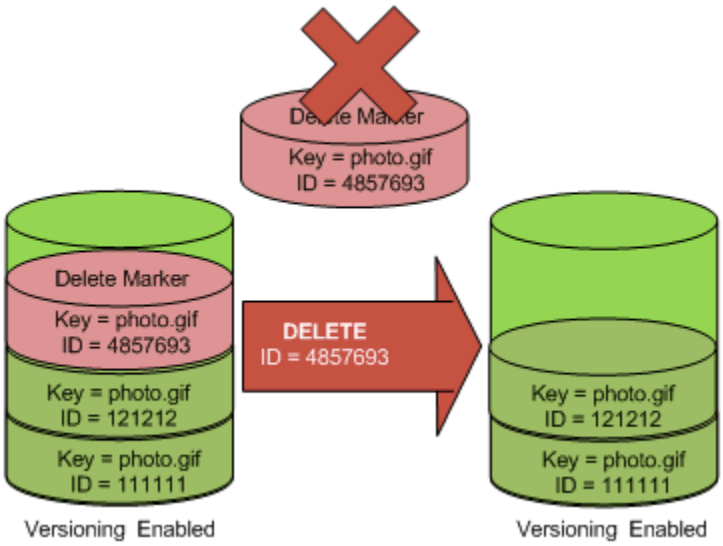
Penanda hapus objek yang kedaluwarsa adalah penanda yang semua versi objeknya dihapus dan hanya satu penanda hapus yang tersisa. Jika konfigurasi siklus hidup diatur untuk menghapus versi saat ini, atau tindakan `ExpiredObjectDeleteMarker` diatur secara eksplisit, Amazon S3

menghapus penanda hapus objek yang kedaluwarsa. Sebagai contoh, lihat Menghapus penanda penghapusan objek yang kedaluwarsa dalam bucket berkemampuan versi.

Menghapus penanda hapus untuk membuat versi yang lebih lama menjadi versi saat ini

Saat Anda menghapus sebuah objek dalam bucket dengan dukungan Penentuan Versi, semua versi tetap berada dalam bucket, dan Amazon S3 membuat penanda hapus untuk objek tersebut. Untuk membatalkan penghapusan objek, Anda harus menghapus penanda hapus ini. Untuk informasi selengkapnya tentang Penentuan Versi dan penanda hapus, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.

Untuk menghapus penanda hapus secara permanen, Anda harus menyertakan ID versinya dalam permintaan `DeleteObject versionId`. Gambar berikut menunjukkan cara permintaan `DeleteObject versionId` menghapus penanda hapus secara permanen.

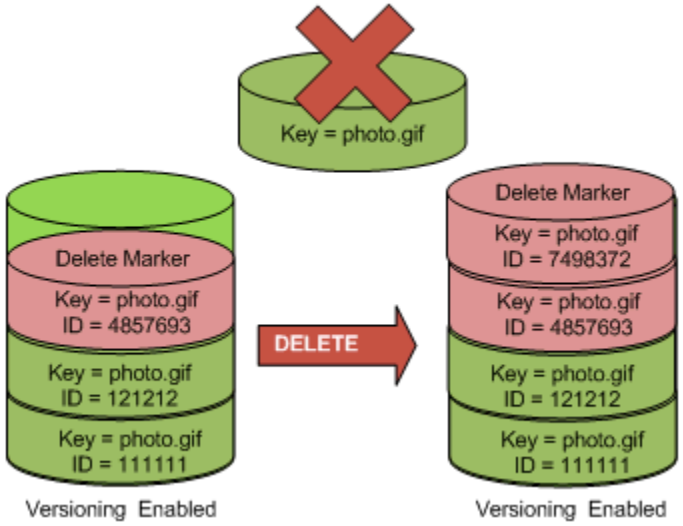


Efek dari menghapus penanda hapus adalah bahwa permintaan `GET` sederhana sekarang akan mengambil ID versi objek saat ini (121212).

> **Note**
>
> Jika Anda menggunakan permintaan `DeleteObject` yang versinya saat ini adalah penanda hapus (tanpa menentukan ID versi penanda hapus), Amazon S3 tidak akan menghapus penanda hapus, melainkan `PUTs` penanda hapus lainnya.

Untuk menghapus penanda hapus dengan ID versi `NULL`, Anda harus meneruskan `NULL` sebagai ID versi dalam permintaan `DeleteObject`. Gambar berikut menunjukkan bagaimana permintaan

`DeleteObject` sederhana dibuat tanpa ID versi yang versinya saat ini adalah penanda hapus, tidak menghapus apa pun, tetapi menambahkan penanda hapus dengan ID versi unik (7498372).



Menggunakan konsol S3

Gunakan langkah-langkah berikut untuk memulihkan objek yang dihapus yang bukan merupakan folder dari bucket S3 Anda, termasuk objek yang ada di dalam folder tersebut.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Dari daftar Bucket, pilih nama bucket yang Anda inginkan.
3. Untuk melihat daftar versi objek di bucket, pilih tombol Buat daftar versi. Anda akan dapat melihat penanda hapus untuk objek yang dihapus.
4. Untuk membatalkan penghapusan objek, Anda harus menghapus penanda hapus ini. Centang kotak di samping penanda hapus objek yang akan dipulihkan, lalu pilih Hapus.
5. Konfirmasi penghapusan pada halaman Hapus objek.
   a. Untuk Hapus objek secara permanen?, masukkan **`permanently delete`**.
   b. Pilih Hapus objek.

> **Note**
>
> Anda tidak dapat menggunakan konsol Amazon S3 untuk membatalkan penghapusan folder. Anda harus menggunakan AWS CLI atau SDK. Sebagai contoh, lihat Bagaimana saya bisa

mengambil objek Amazon S3 yang telah dihapus di bucket dengan dukungan Penentuan Versi? di Pusat Pengetahuan AWS .

Penggunaan API REST

Untuk menghapus penanda hapus secara permanen

1. Tetapkan `versionId` ke ID versi ke penanda hapus yang ingin Anda hapus.
2. Kirim permintaan `DELETE Object versionId`.

Example — Menghapus penanda hapus

Contoh berikut menghapus penanda hapus untuk versi `photo.gif` 4857693.

```
DELETE /photo.gif?versionId=4857693 HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Saat Anda menghapus penanda hapus, Amazon S3 menyertakan berikut ini di dalam respons.

```
204 NoContent
x-amz-version-id: versionID
x-amz-delete-marker: true
```

Menggunakan AWS SDKs

Untuk informasi tentang menggunakan yang lain AWS SDKs, lihat Pusat AWS Pengembang.

Python

Contoh kode Python berikut menunjukkan cara menghapus penanda hapus dari suatu objek dan dengan demikian menjadikan versi bukan saat ini yang terkini, versi objek saat ini.

```
def revive_object(bucket, object_key):
    """
    Revives a versioned object that was deleted by removing the object's active
    delete marker.
    A versioned object presents as deleted when its latest version is a delete
 marker.
```

```
    By removing the delete marker, we make the previous version the latest version
    and the object then presents as *not* deleted.

    Usage is shown in the usage_demo_single_object function at the end of this
module.

    :param bucket: The bucket that contains the object.
    :param object_key: The object to revive.
    """
    # Get the latest version for the object.
    response = s3.meta.client.list_object_versions(
        Bucket=bucket.name, Prefix=object_key, MaxKeys=1
    )

    if "DeleteMarkers" in response:
        latest_version = response["DeleteMarkers"][0]
        if latest_version["IsLatest"]:
            logger.info(
                "Object %s was indeed deleted on %s. Let's revive it.",
                object_key,
                latest_version["LastModified"],
            )
            obj = bucket.Object(object_key)
            obj.Version(latest_version["VersionId"]).delete()
            logger.info(
                "Revived %s, active version is now %s  with body '%s'",
                object_key,
                obj.version_id,
                obj.get()["Body"].read(),
            )
        else:
            logger.warning(
                "Delete marker is not the latest version for %s!", object_key
            )
    elif "Versions" in response:
        logger.warning("Got an active version for %s, nothing to do.", object_key)
    else:
        logger.error("Couldn't get any version info for %s.", object_key)
```

Menghapus objek dari bucket dengan dukungan penghapusan MFA

Jika konfigurasi Penentuan Versi bucket didukung penghapusan MFA, pemilik bucket harus menyertakan header permintaan `x-amz-mfa` dalam permintaan untuk secara permanen menghapus versi objek atau mengubah status Penentuan Versi bucket. Permintaan yang mencakup `x-amz-mfa` harus menggunakan HTTPS.

Nilai header adalah rangkaian nomor seri perangkat autentikasi Anda, spasi, dan kode autentikasi yang ditampilkan di atasnya. Jika Anda tidak menyertakan header permintaan ini, permintaan akan gagal.

Untuk informasi selengkapnya tentang perangkat autentikasi, lihat Autentikasi Multi-faktor.

Example — Menghapus objek dari bucket dengan dukungan penghapusan MFA

Contoh berikut menghapus `my-image.jpg` (dengan versi yang ditentukan), yang ada dalam bucket yang dikonfigurasi dengan dukungan penghapusan MFA.

Perhatikan ruang antara *`[SerialNumber]`* dan*`[AuthenticationCode]`*. Untuk informasi selengkapnya, lihat DeleteObject dalam Referensi API Amazon Simple Storage Service.

```
DELETE /my-image.jpg?versionId=3HL4kqCxf3vjVBH40Nrjfkd HTTPS/1.1
Host: bucketName.s3.amazonaws.com
x-amz-mfa: 20899872 301749
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Untuk informasi selengkapnya tentang mengaktifkan penghapusan MFA, lihat Mengonfigurasi penghapusan MFA.

Mengonfigurasi izin objek berversi

Izin untuk objek di Amazon S3 ditetapkan pada tingkat versi. Setiap versi mempunyai pemilik objeknya sendiri. Akun AWS Yang membuat versi objek adalah pemiliknya. Jadi, Anda dapat mengatur izin yang berbeda-beda untuk berbagai versi dari objek yang sama. Untuk melakukannya, Anda harus menentukan ID versi objek yang izinnya ingin Anda tetapkan di permintaan `PUT Object versionId acl`. Untuk penjelasan rinci dan petunjuk tentang penggunaan ACLs, lihatIdentity and Access Management untuk Amazon S3.

Example — Mengatur izin untuk versi objek

Permintaan berikut menetapkan izin penerima, `BucketOwner@amazon.com`, ke `FULL_CONTROL` pada kunci, `my-image.jpg`, ID versi, `3HL4kqtJvjVBH40Nrjfkd`.

```
PUT /my-image.jpg?acl&versionId=3HL4kqtJvjVBH40Nrjfkd HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
Content-Length: 124

<AccessControlPolicy>
  <Owner>
    <ID>75cc57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
    <DisplayName>mtd@amazon.com</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:type="CanonicalUser">
        <ID>a9a7b886d6fd24a52fe8ca5bef65f89a64e0193f23000e241bf9b1c61be666e9</ID>
        <DisplayName>BucketOwner@amazon.com</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
  </AccessControlPolicy>
```

Demikian pula, untuk mendapatkan izin versi objek tertentu, Anda harus menentukan ID versinya di permintaan `GET Object versionId acl`. Anda perlu memasukkan ID versi karena, secara default, `GET Object acl` mengembalikan izin versi objek saat ini.

Example — Mengambil izin untuk versi objek tertentu

Dalam contoh berikut, Amazon S3 mengembalikan izin untuk kunci, `my-image.jpg`, ID versi, `DVBH40Nr8X8gUMLUo`.

```
GET /my-image.jpg?versionId=DVBH40Nr8X8gUMLUo&acl HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU
```

Untuk informasi selengkapnya, lihat GetObjectAcl dalam Referensi API Amazon Simple Storage Service.

Bekerja dengan objek dalam bucket dengan Penentuan Versi ditangguhkan

Di Amazon S3, Anda dapat menangguhkan Penentuan Versi untuk berhenti mengumpulkan versi baru dari objek yang sama dalam bucket. Anda mungkin melakukan ini karena Anda hanya menginginkan satu versi objek dalam bucket. Atau, mungkin Anda tidak ingin menambah biaya untuk beberapa versi.

Saat Anda menangguhkan Penentuan Versi, objek yang ada di bucket tidak berubah. Perubahannya adalah cara Amazon S3 menangani objek di masa mendatang. Topik dalam bagian ini menjelaskan berbagai operasi objek dalam bucket dengan Penentuan Versi ditangguhkan, termasuk menambahkan, mengambil, dan menghapus objek.

Untuk informasi selengkapnya tentang Penentuan Versi S3, lihat Mempertahankan beberapa versi objek dengan S3 Versioning. Untuk informasi selengkapnya tentang pengambilan versi objek, lihat Mengambil versi objek dari bucket dengan dukungan Penentuan Versi.

Topik

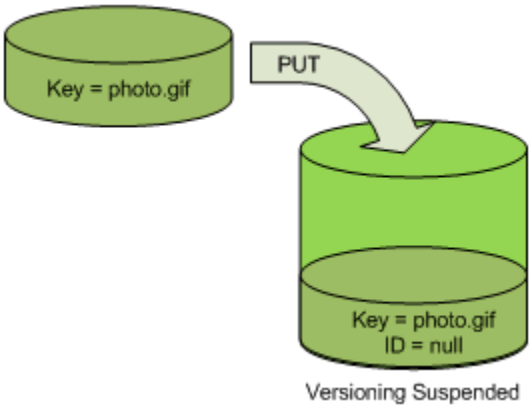
- Menambahkan objek ke bucket dengan Penentuan Versi ditangguhkan
- Mengambil objek dari bucket dengan Penentuan Versi ditangguhkan
- Menghapus objek dari bucket dengan Penentuan Versi ditangguhkan

Menambahkan objek ke bucket dengan Penentuan Versi ditangguhkan

Anda dapat menambahkan objek ke bucket dengan Penentuan Versi ditangguhkan di Amazon S3 untuk membuat objek dengan ID versi null atau menimpa versi objek apa pun dengan ID versi yang cocok.

Setelah Anda menangguhkan Penentuan Versi di bucket, Amazon S3 secara otomatis menambahkan ID versi `null` ke setiap objek berikutnya yang disimpan (menggunakan `PUT`, `POST`, atau `CopyObject`) di bucket tersebut.

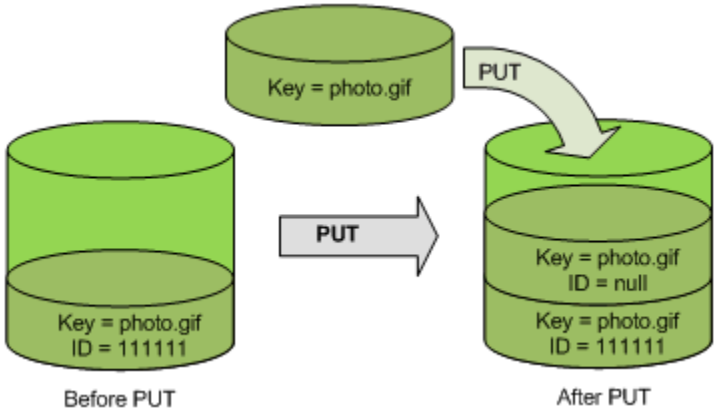
Gambar berikut menunjukkan cara Amazon S3 menambahkan ID versi dari `null` ke objek saat ditambahkan ke bucket dengan Penentuan Versi ditangguhkan.



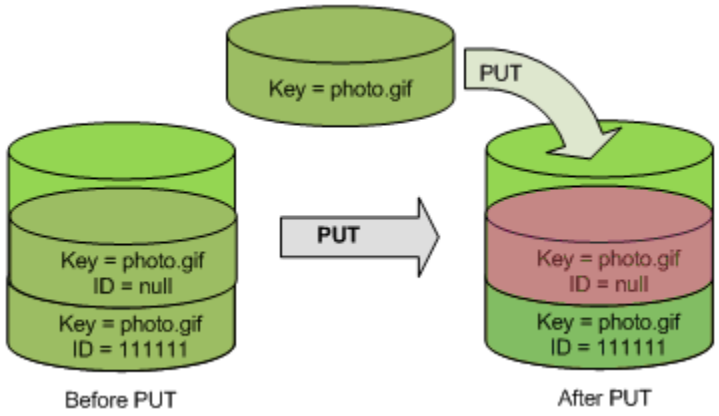
Jika versi null sudah ada dalam bucket dan Anda menambahkan objek lain dengan kunci yang sama, objek tambahan akan menimpa versi null asli.

Apabila terdapat objek berversi dalam bucket, versi yang Anda PUT menjadi versi objek saat ini. Gambar berikut menunjukkan cara menambahkan objek ke bucket yang berisi objek berversi yang tidak menimpa objek yang sudah berada dalam bucket.

Dalam hal ini, versi 111111 sudah ada dalam bucket. Amazon S3 melampirkan ID versi null pada objek yang ditambahkan dan menyimpannya dalam bucket. Versi 111111 tidak ditimpa.



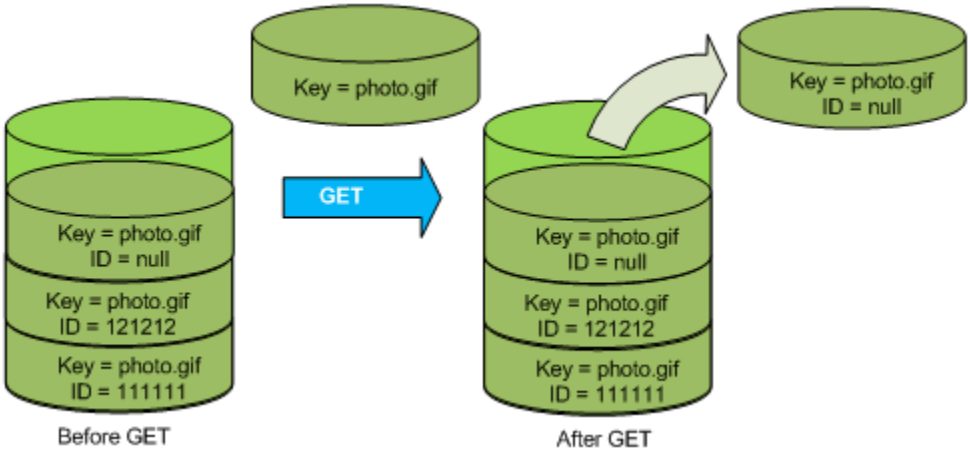
Jika versi null sudah ada di dalam bucket, versi null akan ditimpa, seperti yang ditunjukkan pada gambar berikut.



Meskipun kunci dan ID versi (`null`) dari versi null adalah sama sebelum dan sesudah PUT, isi versi null yang semula disimpan di bucket diganti dengan isi objek PUT ke dalam bucket.

Mengambil objek dari bucket dengan Penentuan Versi ditangguhkan

Permintaan `GET Object` mengembalikan versi objek saat ini, baik apakah Anda telah mengaktifkan Penentuan Versi pada bucket maupun tidak. Gambar berikut menunjukkan cara `GET` sederhana mengembalikan versi objek saat ini.



Menghapus objek dari bucket dengan Penentuan Versi ditangguhkan

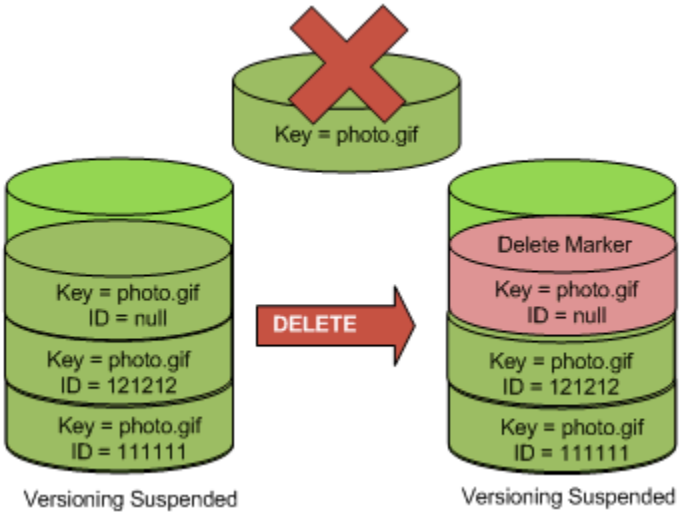
Anda dapat menghapus objek dari bucket dengan Penentuan Versi ditangguhkan untuk menghapus objek dengan ID versi null.

Jika Penentuan Versi ditangguhkan untuk sebuah bucket, permintaan `DELETE`:

- Hanya dapat menghapus objek yang ID versinya adalah `null`.
- Tidak menghapus apa pun jika tidak ada versi null dari objek dalam bucket.

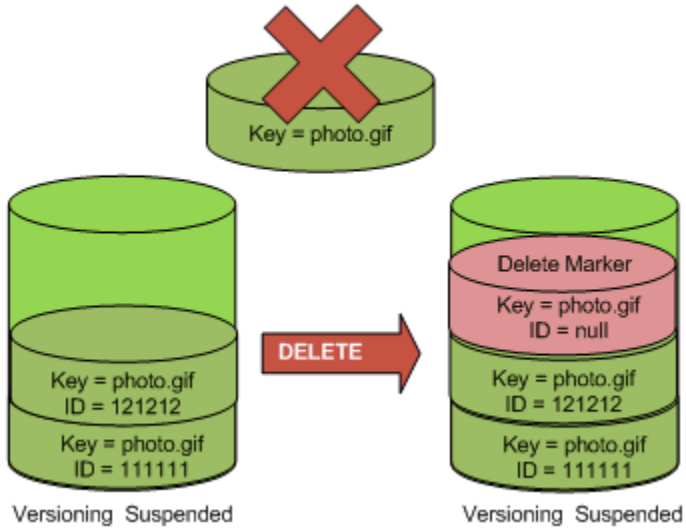
- Memasukkan penanda hapus ke dalam bucket.

Jika pembuatan versi bucket ditangguhkan, operasi akan menghapus objek yang memiliki nol. `versionId` Jika ada ID versi, Amazon S3 menyisipkan penanda hapus yang menjadi versi objek saat ini. Gambar berikut menunjukkan bagaimana sederhana `DELETE` menghapus versi null dan Amazon S3 menyisipkan penanda hapus di tempatnya sebagai gantinya dengan `null` ID versi.

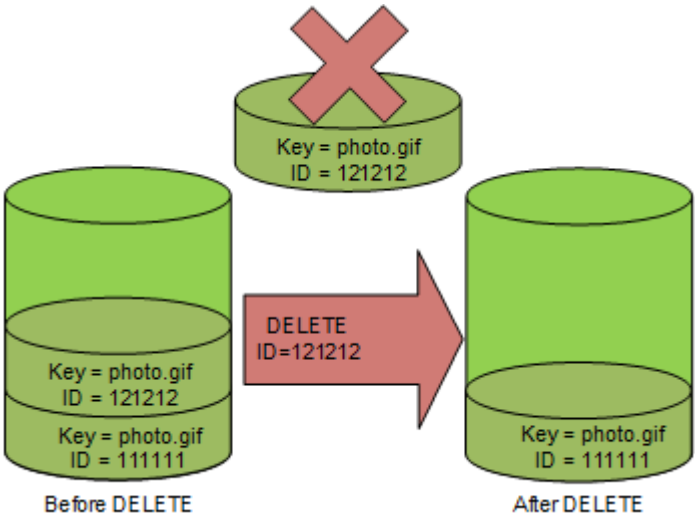


Untuk menghapus objek yang memiliki a secara permanen`versionId`, Anda harus menyertakan objek `versionId` dalam permintaan. Karena penanda hapus tidak berisi konten apa pun, Anda akan kehilangan konten untuk `null` versi saat penanda hapus menggantikannya.

Gambar berikut menunjukkan bucket yang tidak memiliki versi null. Dalam hal ini, tidak `DELETE` menghapus apa-apa. Sebagai gantinya, Amazon S3 hanya menyisipkan penanda hapus.



Bahkan di bucket dengan Penentuan Versi ditangguhkan, pemilik bucket dapat menghapus secara permanen versi tertentu dengan menyertakan ID versi dalam permintaan DELETE. Gambar berikut menunjukkan bahwa menghapus versi objek tertentu akan secara permanen menghapus objek tersebut. Hanya pemilik bucket yang dapat menghapus versi objek tertentu.



Pemecahan masalah pembuatan versi

Topik berikut ini dapat membantu Anda memecahkan beberapa masalah umum Penentuan Versi Amazon S3.

Topik

- Saya ingin memulihkan objek yang secara tidak sengaja dihapus dalam bucket dengan Penentuan Versi diaktifkan
- Saya ingin menghapus objek berversi secara permanen
- Saya mengalami penurunan kinerja setelah mengaktifkan Penentuan Versi bucket

Saya ingin memulihkan objek yang secara tidak sengaja dihapus dalam bucket dengan Penentuan Versi diaktifkan

Secara umum, ketika versi objek dihapus dari bucket S3, tidak ada cara bagi Amazon S3 untuk memulihkannya. Namun, jika Anda sudah mengaktifkan Penentuan Versi S3 di bucket S3, permintaan DELETE yang tidak menentukan ID versi tidak dapat menghapus objek secara permanen. Sebagai gantinya, penanda hapus ditambahkan sebagai placeholder. Penanda hapus ini menjadi versi objek saat ini.

Untuk memverifikasi apakah objek yang dihapus telah dihapus secara permanen atau dihapus sementara (ditandai dengan penanda hapus), lakukan hal berikut:

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Di dalam daftar Bucket, pilih nama bucket yang berisi objek.
4. Pada daftar Objek, Aktifkan tombol Tampilkan versi di sebelah kanan bilah pencarian, lalu cari objek yang dihapus di bilah pencarian. Tombol beralih ini hanya tersedia jika Penentuan Versi sebelumnya telah diaktifkan di bucket.

   Anda juga dapat menggunakan Inventaris S3 untuk mencari objek yang dihapus.
5. Jika Anda tidak dapat menemukan objek setelah mengalihkan Tampilkan versi atau membuat laporan inventaris, dan Anda juga tidak dapat menemukan penanda hapus objek, penghapusan bersifat permanen dan objek tidak dapat dipulihkan.

Anda juga dapat memverifikasi status objek yang dihapus dengan menggunakan operasi `HeadObject` API dari AWS Command Line Interface (AWS CLI). Untuk menggunakannya, ganti perintah `head-object` dengan *`user input placeholders`* informasi Anda sendiri:

```
aws s3api head-object --bucket amzn-s3-demo-bucket --key index.html
```

Jika Anda menjalankan perintah `head-object` pada objek berversi yang versinya saat ini adalah penanda hapus, Anda akan menerima kesalahan 404 Tidak Ditemukan. Misalnya:

Terjadi kesalahan (404) saat memanggil HeadObject operasi: Tidak Ditemukan

Jika Anda menjalankan perintah `head-object` pada objek berversi dan memberikan ID versi objek, Amazon S3 mengambil metadata objek, mengonfirmasi bahwa objek tersebut masih ada dan tidak dihapus secara permanen.

```
aws s3api head-object --bucket amzn-s3-demo-bucket --key index.html --
version-id versionID
```

```
{
"AcceptRanges": "bytes",
"ContentType": "text/html",
"LastModified": "Thu, 16 Apr 2015 18:19:14 GMT",
```

```
"ContentLength": 77,
"VersionId": "Zg5HyL7m.eZU9iM7AVlJkrqAiE.0UG4q",
"ETag": "\"30a6ec7e1a9ad79c203d05a589c8b400\"",
"Metadata": {}
}
```

Jika objek ditemukan dan versi terbaru adalah penanda hapus, versi objek sebelumnya masih akan ada. Karena penanda hapus adalah versi objek saat ini, Anda dapat memulihkan objek dengan menghapus penanda hapus.

Setelah Anda menghapus penanda hapus secara permanen, versi terbaru kedua dari objek akan menjadi versi objek saat ini, sehingga objek Anda akan kembali tersedia. Untuk gambaran visual tentang bagaimana objek dipulihkan, lihat Menghapus penanda hapus.

Untuk menghapus versi objek tertentu, Anda harus menjadi pemilik bucket. Untuk menghapus penanda hapus secara permanen, Anda harus menyertakan ID versinya dalam permintaan `DeleteObject`. Untuk menghapus penanda hapus, gunakan perintah berikut, lalu ganti *`user input placeholders`* dengan informasi Anda sendiri:

```
aws s3api delete-object --bucket amzn-s3-demo-bucket --key index.html --
version-id versionID
```

Untuk informasi selengkapnya tentang perintah `delete-object`, lihat delete-object di Referensi Perintah AWS CLI . Untuk informasi selengkapnya tentang cara menghapus penanda hapus secara permanen, lihat. Mengelola penanda hapus

Saya ingin menghapus objek berversi secara permanen

Pada bucket dengan Penentuan Versi diaktifkan, permintaan `DELETE` tanpa ID versi tidak dapat menghapus objek secara permanen. Sebaliknya, permintaan semacam itu akan menyisipkan penanda hapus.

Untuk menghapus objek berversi secara permanen, Anda dapat memilih dari metode berikut:

- Buat aturan Siklus Hidup S3 untuk menghapus versi tidak terkini secara permanen. Untuk menghapus versi tidak terkini secara permanen, pilih Hapus versi objek tidak terkini secara permanen, lalu masukkan nomor di bawah Hari setelah objek menjadi tidak terkini. Anda dapat secara opsional menentukan jumlah versi yang lebih baru untuk dipertahankan dengan memasukkan nilai di bawah Jumlah versi yang lebih baru untuk dipertahankan. Untuk informasi selengkapnya tentang membuat aturan ini, lihat Menyetel konfigurasi Siklus Hidup S3.

- Hapus versi tertentu dengan menyertakan ID versi dalam permintaan DELETE. Untuk informasi selengkapnya, lihat Cara menghapus objek berversi secara permanen.
- Buat aturan siklus hidup untuk mengakhiri versi saat ini. Untuk mengakhiri versi objek saat ini, pilih Akhiri versi objek saat ini, lalu tambahkan angka di bawah Hari setelah pembuatan objek. Untuk informasi selengkapnya tentang membuat aturan siklus hidup ini, lihat Mengatur konfigurasi Siklus Hidup S3.
- Untuk menghapus semua objek berversi dan menghapus penanda secara permanen, buat dua aturan siklus hidup: satu untuk mengakhiri versi saat ini dan menghapus versi objek yang tidak terkini secara permanen, dan yang lainnya untuk menghapus penanda hapus objek yang kedaluwarsa.

Dalam bucket dengan Penentuan Versi yang diaktifkan, permintaan DELETE yang tidak menentukan ID versi hanya dapat menghapus objek dengan ID versi NULL. Jika objek diunggah saat Penentuan Versi diaktifkan, permintaan DELETEyang tidak menentukan ID versi akan membuat penanda hapus dari objek tersebut.

> **Note**
>
> Untuk bucket berkemampuan Kunci Objek S3, permintaan objek DELETE dengan ID versi objek yang dilindungi menyebabkan kesalahan 403 Akses Ditolak. Permintaan objek DELETE tanpa ID versi menambahkan penanda hapus sebagai versi terbaru dari objek dengan respons 200 OK. Objek yang dilindungi oleh Kunci Objek tidak dapat dihapus secara permanen sampai periode retensi dan penahanan hukumnya dihapus. Untuk informasi selengkapnya, lihat the section called “Cara kerja Kunci Objek S3”.

Saya mengalami penurunan kinerja setelah mengaktifkan Penentuan Versi bucket

Penurunan kinerja dapat terjadi pada bucket berkemampuan Penentuan Versi jika ada terlalu banyak penanda hapus atau objek berversi, dan jika praktik terbaik tidak diikuti.

Terlalu banyak penanda hapus

Setelah Anda mengaktifkan Penentuan Versi pada bucket, permintaan DELETE tanpa ID versi yang dibuat pada objek akan membuat penanda hapus dengan ID versi unik. Konfigurasi siklus hidup dengan aturan Akhiri versi objek saat ini menambahkan penanda hapus dengan ID versi unik ke setiap objek. Penanda hapus yang berlebihan dapat mengurangi kinerja dalam bucket.

Saat Penentuan Versi ditangguhkan pada bucket, Amazon S3 menandai ID versi sebagai NULL pada objek yang baru dibuat. Tindakan pengakhiran dalam bucket yang ditangguhkan Penentuan Versi menyebabkan Amazon S3 membuat penanda hapus dengan NULL sebagai ID versi. Dalam bucket yang ditangguhkan Penentuan Versi, penanda hapus NULL dibuat untuk permintaan penghapusan apa pun. Penanda hapus NULL ini juga disebut penanda hapus objek kedaluwarsa ketika semua versi objek dihapus dan hanya satu penanda hapus yang tersisa. Jika terlalu banyak penanda hapus NULL yang terakumulasi, akan terjadi penurunan kinerja dalam bucket.

Terlalu banyak objek berversi

Jika bucket berkemampuan Penentuan Versi berisi objek dengan jutaan versi, peningkatan kesalahan 503 Layanan Tidak Tersedia dapat terjadi. Jika Anda menyadari adanya peningkatan signifikan dalam jumlah HTTP 503 Layanan Tidak Tersedia yang diterima untuk permintaan objek PUT atau DELETE ke bucket dengan Penentuan Versi yang diaktifkan, Anda kemungkinan memiliki satu atau beberapa objek dalam bucket yang memiliki jutaan versi. Jika Anda memiliki objek dengan jutaan versi, Amazon S3 secara otomatis membatasi permintaan ke bucket tersebut. Permintaan pembatasan melindungi bucket Anda dari jumlah lalu lintas permintaan yang berlebihan, yang dapat berpotensi mengganggu permintaan lain yang dilakukan ke bucket yang sama.

Untuk menentukan objek mana yang memiliki jutaan versi, gunakan Inventaris S3. S3 Inventory menghasilkan laporan yang menyediakan daftar file datar dari objek dalam bucket. Untuk informasi selengkapnya, lihat Membuat katalog dan menganalisis data Anda dengan S3 Inventory.

Untuk memverifikasi apakah ada banyak objek berversi dalam bucket, gunakan metrik Lensa Penyimpanan S3 untuk melihat Jumlah objek versi saat ini, Jumlah objek versi tidak terkini, dan Jumlah objek penanda hapus. Untuk informasi lebih lanjut tentang metrik Lensa Penyimpanan, silakan lihat Glosarium metrik Lensa Penyimpanan Amazon S3.

Tim Amazon S3 menganjurkan pelanggan untuk menyelidiki aplikasi yang berulang kali menimpa objek yang sama, yang berpotensi menciptakan jutaan versi untuk objek tersebut, sehingga memastikan apakah aplikasi bekerja sebagaimana mestinya. Misalnya, aplikasi menimpa objek yang sama setiap menit selama seminggu dapat membuat lebih dari sepuluh ribu versi. Kami merekomendasikan menyimpan kurang dari seratus ribu versi untuk setiap objek. Jika Anda memiliki kasus penggunaan yang membutuhkan jutaan versi untuk satu atau lebih objek, hubungi AWS Dukungan tim untuk bantuan dalam menentukan solusi yang lebih baik.

Praktik terbaik

Untuk mencegah masalah penurunan performa terkait Penentuan Versi, kami sarankan Anda menerapkan praktik terbaik berikut:

- Aktifkan aturan siklus hidup untuk mengakhiri versi objek sebelumnya. Misalnya, Anda dapat membuat aturan siklus hidup untuk mengakhiri versi tidak terkini setelah 30 hari objek menjadi tidak terkini. Anda juga dapat mempertahankan beberapa versi tidak terkini jika Anda tidak ingin menghapus semuanya. Untuk informasi selengkapnya, lihat Mengatur konfigurasi Siklus Hidup S3.
- Aktifkan aturan siklus hidup untuk menghapus penanda penghapusan objek kedaluwarsa yang tidak memiliki objek data terkait di bucket. Untuk informasi selengkapnya, lihat Menghapus penanda hapus objek kedaluwarsa.

Untuk praktik terbaik pengoptimalan kinerja Amazon S3 tambahan, lihat Pola desain praktik terbaik.

Mengunci objek dengan Object Lock

Kunci Objek S3 dapat membantu Anda mencegah penghapusan atau penimpaan objek Amazon S3 selama jangka waktu tertentu atau tanpa batas waktu. Object Lock menggunakan model write-once-read-many(WORM) untuk menyimpan objek. Anda dapat menggunakan Object Lock untuk membantu memenuhi persyaratan peraturan yang memerlukan penyimpanan WORM, atau untuk menambahkan lapisan perlindungan lain terhadap perubahan atau penghapusan objek.

> **Note**
>
> Kunci Objek S3 telah dinilai oleh Cohasset Associates untuk digunakan di lingkungan yang tunduk pada peraturan SEC 17a-4, CFTC, dan FINRA. Untuk informasi selengkapnya tentang bagaimana Object Lock berhubungan dengan peraturan ini, lihat Cohasset Associates Penilaian Kepatuhan.

Kunci Objek menyediakan dua cara untuk mengelola retensi objek: periode retensi dan penahanan legal. Versi objek dapat memiliki periode retensi, penahanan legal, atau keduanya.

- Periode retensi — Periode retensi menentukan periode waktu tetap selama versi objek tetap terkunci. Anda dapat mengatur periode retensi unik untuk objek individual. Selain itu, Anda dapat mengatur periode retensi default pada bucket S3. Anda juga dapat membatasi periode retensi minimum dan maksimum yang diijinkan dengan kunci `s3:object-lock-remaining-retention-days` kondisi dalam kebijakan bucket. Ini membantu Anda menetapkan rentang periode retensi dan dengan membatasi periode retensi yang mungkin lebih pendek atau lebih lama dari rentang ini.

- Penahanan legal—Penahanan legal memberikan perlindungan yang sama dengan periode retensi, tetapi tidak memiliki tanggal kedaluwarsa. Penahanan legal akan tetap berlaku sampai Anda menghapusnya secara eksplisit. Penahanan hukum independen dari periode retensi dan ditempatkan pada versi objek individual.

Kunci Objek hanya berfungsi di bucket yang mengaktifkan Penentuan Versi S3. Saat Anda mengunci versi objek, Amazon S3 menyimpan informasi kunci di metadata untuk versi objek tersebut. Menempatkan periode retensi atau penahanan legal pada objek hanya melindungi versi yang ditentukan dalam permintaan. Periode retensi dan penahanan hukum tidak mencegah versi baru objek dibuat, atau menghapus penanda untuk ditambahkan di atas objek. Untuk informasi tentang Penentuan Versi S3, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.

Jika Anda memasukkan objek ke dalam bucket yang sudah berisi objek terproteksi dengan nama kunci objek yang sama, Amazon S3 akan membuat versi baru dari objek tersebut. Versi objek yang diproteksi saat ini tetap terkunci sesuai dengan konfigurasi retensinya.

Cara kerja Kunci Objek S3

Topik

- Periode retensi
- Mode retensi
- Penahanan legal
- Praktik terbaik untuk menggunakan S3 Object Lock
- Izin yang diperlukan

Periode retensi

Periode retensi melindungi versi objek untuk jangka waktu tertentu. Saat Anda menempatkan periode retensi pada versi objek, Amazon S3 menyimpan stempel waktu dalam metadata versi objek untuk menunjukkan waktu berakhirnya periode retensi. Setelah periode retensi berakhir, versi objek dapat ditimpa atau dihapus.

Anda dapat menempatkan periode retensi secara eksplisit pada versi objek individual atau pada properti bucket sehingga berlaku untuk semua objek di dalam bucket secara otomatis. Saat Anda menerapkan periode retensi ke versi objek secara eksplisit, Anda menentukan Simpan Hingga Tanggal untuk versi objek. Amazon S3 menyimpan tanggal ini di metadata versi objek.

Anda juga dapat mengatur periode retensi di properti bucket. Saat menetapkan periode retensi pada suatu bucket, Anda menentukan durasi, baik dalam jumlah hari atau tahun, untuk melindungi setiap versi objek yang ditempatkan di dalam bucket tersebut. Saat Anda menempatkan objek di dalam bucket, Amazon S3 menghitung Simpan Hingga Tanggal untuk versi objek dengan menambahkan durasi yang ditentukan ke stempel waktu Penentuan Versi objek. Kemudian versi objek akan dilindungi persis seperti saat Anda memasang kunci individual dengan periode retensi tersebut pada versi objek.

> **Note**
>
> Jika Anda memiliki versi objek PUT yang memiliki mode dan periode retensi individual eksplisit dalam bucket, pengaturan Kunci Objek individual versi objek akan menimpa pengaturan retensi properti bucket apa pun.

Seperti semua pengaturan Kunci Objek lainnya, periode retensi berlaku untuk versi objek individu. Berbagai versi dari suatu objek dapat memiliki mode dan periode retensi yang berbeda-beda.

Misalkan, Anda memiliki suatu objek yang sudah berada pada periode retensi selama 15 hari dari total 30 hari, dan Anda PUT objek tersebut ke dalam Amazon S3 dengan nama yang sama dan periode retensi selama 60 hari. Dalam hal ini, permintaan PUT Anda berhasil, dan Amazon S3 akan membuat versi baru objek dengan periode retensi 60 hari. Versi terdahulu akan mempertahankan periode retensi aslinya dan dapat dihapus dalam 15 hari.

Setelah menerapkan pengaturan retensi ke versi objek, Anda dapat memperpanjang periode retensi. Caranya, kirimkan permintaan Kunci Objek baru untuk versi objek dengan Simpan Hingga Tanggal yang lebih lama dari yang saat ini dikonfigurasi untuk versi objek. Amazon S3 menggantikan periode retensi yang ada dengan periode baru yang lebih lama. Setiap pengguna yang mempunyai izin untuk menempatkan periode retensi objek dapat memperpanjang periode retensi untuk versi objek. Untuk menetapkan periode retensi, Anda harus memiliki izin `s3:PutObjectRetention`.

Saat menetapkan periode retensi pada objek atau bucket S3, Anda harus memilih salah satu dari dua mode retensi: kepatuhan atau tata kelola.

Mode retensi

Kunci Objek S3 menyediakan dua mode retensi yang menerapkan tingkat perlindungan berbeda pada objek Anda:

- Mode kepatuhan

- Mode tata kelola

Di mode kepatuhan, versi objek terlindungi tidak dapat ditimpa atau dihapus oleh pengguna mana pun, termasuk pengguna root di Akun AWS Anda. Ketika sebuah objek dikunci dalam mode kepatuhan, mode retensinya tidak dapat diubah, dan periode retensinya tidak dapat disingkat. Mode kepatuhan membantu memastikan bahwa versi objek tidak dapat ditimpa atau dihapus selama periode retensi.

> **Note**
>
> Satu-satunya cara untuk menghapus objek di bawah mode kepatuhan sebelum tanggal retensi berakhir adalah dengan menghapus yang terkait Akun AWS.

Di mode tata kelola, pengguna tidak dapat menimpa atau menghapus versi objek atau mengubah pengaturan kuncinya kecuali memiliki izin khusus. Dengan mode tata kelola, Anda melindungi objek agar tidak dihapus oleh sebagian besar pengguna, tetapi Anda masih dapat memberikan izin kepada beberapa pengguna untuk mengubah pengaturan penyimpanan atau menghapus objek jika diperlukan. Anda juga dapat menggunakan mode tata kelola untuk menguji pengaturan periode retensi sebelum membuat periode retensi mode kepatuhan.

Untuk menimpa atau menghapus pengaturan retensi mode tata kelola, Anda harus memiliki izin `s3:BypassGovernanceRetention` dan harus secara eksplisit menyertakan `x-amz-bypass-governance-retention:true` sebagai header permintaan dengan permintaan apa pun yang memerlukan penimpaan mode tata kelola.

> **Note**
>
> Secara default, konsol Amazon S3 menyertakan header `x-amz-bypass-governance-retention:true`. Jika Anda mencoba menghapus objek yang dilindungi oleh mode tata kelola dan memiliki izin `s3:BypassGovernanceRetention`, operasi akan berhasil.

Penahanan legal

Dengan Kunci Objek Anda juga dapat menempatkan penahanan legal pada versi objek. Seperti periode retensi, penahanan legal mencegah penimpaan atau penghapusan versi objek. Namun,

penahanan legal tidak memiliki jangka waktu tertentu dan tetap berlaku sampai penahanan tersebut dihapus. Penahanan legal dapat ditempatkan dan dihapus dengan bebas oleh pengguna mana pun yang memiliki izin `s3:PutObjectLegalHold`.

Penahanan legal terpisah dari periode retensi. Menempatkan penahanan legal pada versi objek tidak memengaruhi mode retensi atau periode retensi untuk versi objek tersebut.

Misalnya, Anda menempatkan penahanan legal pada versi objek dan versi objek juga dilindungi oleh periode retensi. Jika periode retensi berakhir, objek tidak kehilangan perlindungan WORM. Bahkan, penahanan legal akan terus melindungi objek hingga pengguna yang berwenang secara eksplisit menghapus penahanan legal. Demikian halnya, jika Anda menghapus penahanan legal saat periode retensi versi objek masih berlaku, versi objek akan tetap terlindungi hingga periode retensi berakhir.

Praktik terbaik untuk menggunakan S3 Object Lock

Pertimbangkan untuk menggunakan mode Tata Kelola jika Anda ingin melindungi objek agar tidak dihapus oleh sebagian besar pengguna selama periode retensi yang telah ditentukan sebelumnya, tetapi pada saat yang sama ingin beberapa pengguna dengan izin khusus memiliki fleksibilitas untuk mengubah pengaturan retensi atau menghapus objek.

Pertimbangkan untuk menggunakan mode Kepatuhan jika Anda tidak ingin pengguna mana pun, termasuk pengguna root di Anda Akun AWS, dapat menghapus objek selama periode retensi yang telah ditentukan sebelumnya. Anda dapat menggunakan mode ini jika Anda memiliki persyaratan untuk menyimpan data yang sesuai.

Anda dapat menggunakan Legal Hold ketika Anda tidak yakin berapa lama Anda ingin objek Anda tetap tidak berubah. Ini bisa jadi karena Anda memiliki audit eksternal data Anda yang akan datang dan ingin menjaga objek tetap tidak berubah sampai audit selesai. Sebagai alternatif, Anda mungkin memiliki proyek yang sedang berlangsung menggunakan kumpulan data yang ingin Anda pertahankan hingga proyek selesai.

Izin yang diperlukan

Operasi Kunci Objek memerlukan izin khusus. Tergantung pada operasi yang sedang dijalankan, Anda mungkin memerlukan salah satu izin berikut ini:

- `s3:BypassGovernanceRetention`
- `s3:GetBucketObjectLockConfiguration`
- `s3:GetObjectLegalHold`

- `s3:GetObjectRetention`
- `s3:PutBucketObjectLockConfiguration`
- `s3:PutObjectLegalHold`
- `s3:PutObjectRetention`

Untuk daftar lengkap izin Amazon S3 dengan deskripsi, lihat Tindakan, sumber daya, dan kunci kondisi untuk Amazon S3 di Referensi Otorisasi Layanan.

Untuk informasi selengkapnya tentang izin operasi S3 API menurut jenis sumber daya S3, lihat. Izin yang diperlukan untuk operasi API Amazon S3

Untuk informasi tentang menggunakan ketentuan dengan izin, lihat Contoh kebijakan bucket menggunakan tombol kondisi.

Pertimbangan Kunci Objek

Kunci Objek Amazon S3 dapat membantu mencegah penghapusan atau penimpaan objek selama jangka waktu tertentu atau tanpa batas waktu.

Anda dapat menggunakan konsol Amazon S3, AWS Command Line Interface (AWS CLI) AWS SDKs, atau Amazon S3 REST API untuk melihat atau menyetel informasi Kunci Objek. Untuk informasi umum tentang kemampuan Kunci Objek S3, lihat Mengunci objek dengan Object Lock.

> **Important**
>
> - Setelah mengaktifkan Kunci Objek pada bucket, Anda tidak dapat menonaktifkan Kunci Objek atau menangguhkan Penentuan Versi untuk bucket tersebut.
> - Bucket S3 dengan Kunci Objek tidak dapat digunakan sebagai bucket tujuan untuk log akses server. Untuk informasi selengkapnya, lihat the section called “Pencatatan akses server”.

Topik

- Izin untuk melihat informasi kunci
- Melewati mode tata kelola
- Menggunakan Kunci Objek dengan Replikasi S3
- Menggunakan Object Lock dengan enkripsi

- Menggunakan Kunci Objek dengan Inventaris Amazon S3
- Mengelola kebijakan Siklus Hidup S3 dengan Object Lock
- Mengelola penanda hapus dengan Object Lock
- Menggunakan Lensa Penyimpanan S3 dengan Kunci Objek
- Mengunggah objek ke bucket yang diaktifkan Object Lock
- Mengonfigurasi peristiwa dan pemberitahuan
- Menetapkan batas periode retensi dengan kebijakan bucket

Izin untuk melihat informasi kunci

Anda dapat secara terprogram melihat status Object Lock dari versi objek Amazon S3 dengan menggunakan HeadObject atau GetObjectoperasi. Kedua operasi tersebut akan mengembalikan mode retensi, simpan hingga tanggal, dan status penahanan legal untuk versi objek tertentu. Selain itu, Anda dapat melihat status Object Lock untuk beberapa objek di bucket S3 menggunakan S3 Inventory.

Untuk melihat mode retensi dan periode retensi versi objek, Anda harus memiliki izin `s3:GetObjectRetention`. Untuk melihat status kontrol legal versi objek, Anda harus memiliki izin `s3:GetObjectLegalHold`. Untuk melihat konfigurasi retensi default bucket, Anda harus memiliki izin `s3:GetBucketObjectLockConfiguration`. Jika Anda membuat permintaan untuk konfigurasi Kunci Objek pada bucket yang tidak memiliki Kunci Objek S3 yang aktif, Amazon S3 akan memberikan pesan kesalahan.

Melewati mode tata kelola

Jika Anda memiliki `s3:BypassGovernanceRetention` izin, Anda dapat melakukan operasi pada versi objek yang terkunci dalam mode tata kelola seolah-olah objek tersebut tidak dilindungi. Operasi ini termasuk menghapus versi objek, memperpendek periode retensi, atau menghapus periode retensi Kunci Objek dengan menempatkan permintaan `PutObjectRetention` baru dengan parameter kosong.

Untuk melewati mode tata kelola, Anda harus secara eksplisit menunjukkan permintaan bahwa Anda ingin melewati mode ini. Untuk melakukannya, sertakan `x-amz-bypass-governance-retention:true` header dengan permintaan operasi `PutObjectRetention` API Anda, atau gunakan parameter yang setara dengan permintaan yang dibuat melalui AWS CLI atau AWS SDKs. Konsol S3 secara otomatis menerapkan header ini untuk permintaan yang dibuat melalui konsol S3 jika Anda memiliki izin `s3:BypassGovernanceRetention`.

> **Note**
>
> Melewati mode tata kelola tidak akan memengaruhi status penahanan legal versi objek. Jika versi objek memiliki penahanan legal yang aktif, penahanan legal tersebut tetap berlaku dan mencegah permintaan untuk menimpa atau menghapus versi objek.

Menggunakan Kunci Objek dengan Replikasi S3

Anda dapat menggunakan Kunci Objek dengan Replikasi S3 untuk mengaktifkan penyalinan otomatis dan asinkron dari objek terkunci dan metadata retensinya, di seluruh bucket S3. Ini berarti bahwa untuk objek yang direplikasi, Amazon S3 mengambil konfigurasi kunci objek dari bucket sumber. Dengan kata lain, jika bucket sumber mengaktifkan Object Lock, bucket tujuan juga harus mengaktifkan Object Lock. Jika objek langsung diunggah ke bucket tujuan (di luar S3 Replication), objek tersebut akan diatur Object Lock pada bucket tujuan. Saat Anda menggunakan replikasi, objek di bucket sumber direplikasi ke satu atau lebih bucket tujuan.

Untuk mengatur replikasi pada bucket dengan Object Lock diaktifkan, Anda dapat menggunakan konsol S3, Amazon S3 REST API AWS CLI, atau. AWS SDKs

> **Note**
>
> Untuk menggunakan Object Lock dengan replikasi, Anda harus memberikan dua izin tambahan pada bucket S3 sumber dalam peran AWS Identity and Access Management (IAM) yang Anda gunakan untuk mengatur replikasi. Dua izin tambahan tersebut adalah `s3:GetObjectRetention` dan`s3:GetObjectLegalHold`. Jika peran tersebut memiliki pernyataan izin `s3:Get*`, pernyataan itu memenuhi persyaratan. Untuk informasi selengkapnya, lihat Menyiapkan izin untuk replikasi langsung.
> Untuk informasi umum tentang Replikasi S3, lihat. Mereplikasi objek di dalam dan di seluruh Wilayah
> Untuk contoh pengaturan Replikasi S3, lihat Contoh untuk mengonfigurasi replikasi langsung.

Menggunakan Object Lock dengan enkripsi

Amazon S3 mengenkripsi semua objek baru secara default. Anda dapat menggunakan Object Lock dengan objek terenkripsi Anda. Untuk informasi selengkapnya, lihat Melindungi data dengan enkripsi.

Meskipun Object Lock dapat membantu mencegah objek Amazon S3 dihapus atau ditimpa, itu tidak melindungi dari kehilangan akses ke kunci enkripsi atau kunci enkripsi yang dihapus. Misalnya, jika Anda mengenkripsi objek Anda dengan enkripsi AWS KMS sisi server dan AWS KMS kunci Anda dihapus, objek Anda mungkin menjadi tidak dapat dibaca.

Menggunakan Kunci Objek dengan Inventaris Amazon S3

Anda dapat mengonfigurasi Inventaris Amazon S3 untuk membuat daftar objek di bucket S3 sesuai jadwal yang sudah ditentukan. Anda dapat mengonfigurasi Inventaris Amazon S3 untuk menyertakan metadata Kunci Objek berikut untuk objek Anda:

- Simpan hingga tanggal
- Mode retensi
- Status penahanan legal

Untuk informasi selengkapnya, lihat Membuat katalog dan menganalisis data Anda dengan S3 Inventory.

Mengelola kebijakan Siklus Hidup S3 dengan Object Lock

Konfigurasi manajemen siklus aktif objek akan tetap berfungsi secara normal pada objek yang terlindungi, termasuk menempatkan penanda hapus. Namun, versi objek yang terkunci tidak dapat dihapus oleh kebijakan kedaluwarsa Siklus Hidup S3. Object Lock dipertahankan terlepas dari kelas penyimpanan mana objek berada di dan sepanjang transisi Siklus Hidup S3 antar kelas penyimpanan.

Untuk informasi selengkapnya tentang mengelola siklus hidup objek, lihat Mengelola siklus hidup objek.

Mengelola penanda hapus dengan Object Lock

Meskipun tidak dapat menghapus versi objek yang dilindungi, Anda tetap dapat membuat penanda hapus untuk objek tersebut. Memasang penanda hapus pada sebuah objek tidak akan menghapus objek atau versi objeknya. Namun, hal ini membuat Amazon S3 bertindak seolah-olah objek tersebut telah dihapus. Untuk informasi selengkapnya, lihat Bekerja dengan penanda hapus.

> **Note**
>
> Penanda hapus tidak dilindungi WORM, terlepas dari periode retensi atau kontrol legal yang diterapkan pada objek yang mendasarinya.

Menggunakan Lensa Penyimpanan S3 dengan Kunci Objek

untuk melihat metrik terkait jumlah byte penyimpanan yang diaktifkan oleh Kunci Objek dan jumlah objek, Anda dapat menggunakan Lensa Penyimpanan Amazon S3. Lensa Penyimpanan S3 adalah fitur analisis penyimpanan cloud yang dapat Anda gunakan untuk mendapatkan visibilitas seluruh organisasi ke dalam penggunaan dan aktivitas penyimpanan objek.

Untuk informasi selengkapnya, lihat Menggunakan Lensa Penyimpanan S3 untuk melindungi data.

Untuk daftar lengkap metrik, lihat Glosarium metrik Lensa Penyimpanan Amazon S3.

Mengunggah objek ke bucket yang diaktifkan Object Lock

`x-amz-sdk-checksum-algorithm`Header `Content-MD5` atau diperlukan untuk setiap permintaan untuk mengunggah objek dengan periode retensi yang dikonfigurasi menggunakan Object Lock. Header tesis adalah cara untuk memverifikasi integritas objek Anda selama upload.

Saat mengunggah objek dengan konsol Amazon S3, S3 secara otomatis menambahkan header. `Content-MD5` Anda dapat secara opsional menentukan fungsi checksum tambahan dan nilai checksum melalui konsol sebagai header. `x-amz-sdk-checksum-algorithm` Jika Anda menggunakan PutObjectAPI, Anda harus menentukan `Content-MD5` header, `x-amz-sdk-checksum-algorithm` header, atau keduanya untuk mengonfigurasi periode retensi Object Lock.

Untuk informasi selengkapnya, lihat Memeriksa integritas objek di Amazon S3.

Mengonfigurasi peristiwa dan pemberitahuan

Anda dapat menggunakan Pemberitahuan Acara Amazon S3 untuk melacak akses dan perubahan pada konfigurasi dan data Object Lock Anda dengan menggunakan. AWS CloudTrail Untuk informasi tentang CloudTrail, lihat Apa itu AWS CloudTrail? dalam AWS CloudTrail User Guide.

Anda juga dapat menggunakan Amazon CloudWatch untuk menghasilkan peringatan berdasarkan data ini. Untuk informasi tentang CloudWatch, lihat Apa itu Amazon CloudWatch? di Panduan CloudWatch Pengguna Amazon.

Menetapkan batas periode retensi dengan kebijakan bucket

Anda dapat menetapkan periode retensi minimum dan maksimum yang diizinkan untuk bucket menggunakan kebijakan bucket. Periode retensi maksimum adalah 100 tahun.

Contoh berikut menunjukkan kebijakan bucket yang menggunakan kunci kondisi `s3:object-lock-remaining-retention-days` untuk menetapkan periode retensi maksimum selama 10 hari.

```
{
    "Version": "2012-10-17",
    "Id": "SetRetentionLimits",
    "Statement": [
        {
            "Sid": "SetRetentionPeriod",
            "Effect": "Deny",
            "Principal": "*",
            "Action": [
                "s3:PutObjectRetention"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*",
            "Condition": {
                "NumericGreaterThan": {
                    "s3:object-lock-remaining-retention-days": "10"
                }
            }
        }
    ]
}
```

> **Note**
>
> Jika bucket Anda adalah bucket tujuan konfigurasi replikasi, Anda dapat mengatur periode retensi minimum dan maksimum yang diizinkan untuk replika objek yang dibuat dengan menggunakan replikasi. Untuk melakukannya, Anda harus mengizinkan tindakan `s3:ReplicateObject` dalam kebijakan bucket. Untuk informasi selengkapnya tentang izin replikasi, lihat the section called “Menyiapkan izin”.

Untuk informasi selengkapnya tentang kebijakan bucket, lihat topik berikut:

- Kunci tindakan, sumber daya, dan kondisi untuk Amazon S3 di Referensi Otorisasi Layanan

Untuk informasi selengkapnya tentang izin operasi S3 API menurut jenis sumber daya S3, lihat. Izin yang diperlukan untuk operasi API Amazon S3

- Operasi objek
- Contoh kebijakan bucket menggunakan tombol kondisi

Mengonfigurasi Kunci Objek S3

Dengan Amazon S3 Object Lock, Anda dapat menyimpan objek di bucket tujuan umum Amazon S3 dengan menggunakan write-once-read-manymodel (WORM). Anda dapat menggunakan Kunci Objek S3 untuk mencegah objek terhapus atau ditimpa selama jangka waktu tertentu atau tanpa batas waktu. Untuk informasi umum tentang kemampuan Kunci Objek, lihat Mengunci objek dengan Object Lock.

Sebelum Anda mengunci objek apa pun, Anda harus mengaktifkan S3 Versioning dan Object Lock pada bucket tujuan umum. Setelah itu, Anda dapat mengatur periode retensi, penyimpanan yang sah, atau keduanya.

Untuk bekerja dengan Kunci Objek, Anda harus memiliki izin tertentu. Untuk daftar izin yang terkait dengan berbagai operasi Kunci Objek, lihat the section called “Izin yang diperlukan”.

> **Important**
>
> - Setelah mengaktifkan Kunci Objek pada bucket, Anda tidak dapat menonaktifkan Kunci Objek atau menangguhkan Penentuan Versi untuk bucket tersebut.
> - Bucket S3 dengan Kunci Objek tidak dapat digunakan sebagai bucket tujuan untuk log akses server. Untuk informasi selengkapnya, lihat the section called “Pencatatan akses server”.

Topik

- Aktifkan Object Lock saat membuat bucket tujuan umum S3 baru
- Aktifkan Kunci Objek pada bucket S3 yang ada
- Mengatur atau memodifikasi penyimpanan yang sah pada objek S3
- Menetapkan atau memodifikasi periode retensi pada objek S3
- Menetapkan atau memodifikasi periode retensi default pada bucket S3

Aktifkan Object Lock saat membuat bucket tujuan umum S3 baru

Anda dapat mengaktifkan Object Lock saat membuat bucket tujuan umum S3 baru dengan menggunakan konsol Amazon S3 AWS Command Line Interface ,AWS CLI() AWS SDKs, atau Amazon S3 REST API.

Menggunakan konsol S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Pilih Buat bucket.

   Halaman Buat bucket terbuka.
4. Untuk Nama bucket, masukkan nama untuk bucket Anda.

   > **Note**
   >
   > Setelah membuat bucket, Anda tidak dapat mengubah namanya. Untuk informasi selengkapnya tentang penamaan bucket, lihat Aturan penamaan bucket tujuan umum.
5. Untuk Wilayah, pilih Wilayah AWS tempat Anda ingin ember berada.
6. Di bawah Kepemilikan Objek, pilih untuk menonaktifkan atau mengaktifkan daftar kontrol akses (ACLs) dan mengontrol kepemilikan objek yang diunggah di bucket Anda.
7. Di bawah Pengaturan Blokir Akses Publik untuk bucket ini, pilih pengaturan Blokir Akses Publik yang ingin Anda terapkan ke bucket.
8. Di bawah Penentuan Versi Bucket, pilih Diaktifkan.

   Kunci Objek hanya berfungsi dengan bucket berversi.
9. (Opsional) Di bawah Tanda, Anda dapat memilih untuk menambahkan tanda ke bucket Anda. Tanda adalah pasangan nilai kunci yang digunakan untuk mengkategorikan penyimpanan dan mengalokasikan biaya.
10. Di bawah Pengaturan lanjutan, cari Kunci Objek dan pilih Aktifkan.

    Anda harus memahami bahwa mengaktifkan Kunci Objek akan membuat objek di dalam bucket ini terkunci secara permanen.
11. Pilih Buat bucket.

Menggunakan AWS CLI

Contoh `create-bucket` berikut membuat bucket S3 baru bernama *amzn-s3-demo-bucket1* dengan Kunci Objek yang diaktifkan:

```
aws s3api create-bucket --bucket amzn-s3-demo-bucket1 --object-lock-enabled-for-bucket
```

Untuk informasi dan contoh selengkapnya, lihat create-bucketdalam AWS CLI Command Reference.

> **Note**
>
> Anda dapat menjalankan AWS CLI perintah dari konsol dengan menggunakan AWS CloudShell. AWS CloudShell adalah shell pra-otentikasi berbasis browser yang dapat Anda luncurkan langsung dari file. AWS Management Console Untuk informasi lebih lanjut, lihat Apa itu CloudShell? dalam AWS CloudShell User Guide.

Penggunaan API REST

Anda dapat menggunakan API REST untuk membuat bucket S3 baru dengan Kunci Objek yang diaktifkan. Untuk informasi selengkapnya, silakan lihat CreateBucket dalam Referensi API Layanan Penyimpanan Sederhana Amazon.

Menggunakan AWS SDKs

Untuk contoh cara mengaktifkan Object Lock saat membuat bucket S3 baru dengan AWS SDKs, lihat Contoh kode di Referensi API Amazon S3.

Untuk contoh cara mendapatkan konfigurasi Object Lock saat ini dengan AWS SDKs, lihat Contoh kode di Referensi API Amazon S3.

Untuk skenario interaktif yang mendemonstrasikan fitur Object Lock yang berbeda menggunakan AWS SDKs, lihat Contoh kode di Referensi API Amazon S3.

Untuk informasi umum tentang penggunaan yang berbeda AWS SDKs, lihat Mengembangkan dengan Amazon S3 menggunakan Referensi API Amazon S3 AWS SDKs di Amazon S3.

Aktifkan Kunci Objek pada bucket S3 yang ada

Anda dapat mengaktifkan Object Lock untuk bucket S3 yang ada dengan menggunakan konsol Amazon S3, AWS CLI API REST Amazon S3 AWS SDKs, atau Amazon S3.

Menggunakan konsol S3

> **Note**
>
> Kunci Objek hanya berfungsi dengan bucket berversi.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Di daftar Bucket, pilih nama bucket yang ingin Anda aktifkan Kunci Objek.
4. Pilih tab Properti.
5. Di bawah Properti, gulir ke bawah ke bagian Kunci Objek, lalu pilih Edit.
6. Di bawah Kunci Objek, pilih Aktifkan.

   Anda harus memahami bahwa mengaktifkan Kunci Objek akan membuat objek di dalam bucket ini terkunci secara permanen.
7. Pilih Simpan perubahan.

Menggunakan AWS CLI

`put-object-lock-configuration`Contoh perintah berikut menetapkan periode retensi Kunci Objek 50 hari pada bucket bernama *`amzn-s3-demo-bucket1`*:

```
aws s3api put-object-lock-configuration --bucket amzn-s3-demo-bucket1 --object-
lock-configuration='{ "ObjectLockEnabled": "Enabled", "Rule": { "DefaultRetention":
 { "Mode": "COMPLIANCE", "Days": 50 }}}'
```

Untuk informasi dan contoh selengkapnya, lihat put-object-lock-configurationdalam AWS CLI Command Reference.

> **Note**
>
> Anda dapat menjalankan AWS CLI perintah dari konsol dengan menggunakan AWS CloudShell. AWS CloudShell adalah shell pra-otentikasi berbasis browser yang dapat Anda

luncurkan langsung dari file. AWS Management Console Untuk informasi lebih lanjut, lihat Apa itu CloudShell? dalam AWS CloudShell User Guide.

Penggunaan API REST

Anda dapat menggunakan API REST Amazon S3 untuk mengaktifkan Kunci Objek pada bucket S3 yang sudah ada. Untuk informasi selengkapnya, silakan lihat PutObjectLockConfiguration dalam Referensi API Layanan Penyimpanan Sederhana Amazon.

Menggunakan AWS SDKs

Untuk contoh cara mengaktifkan Object Lock untuk bucket S3 yang ada dengan AWS SDKs, lihat Contoh kode di Referensi API Amazon S3.

Untuk contoh cara mendapatkan konfigurasi Object Lock saat ini dengan AWS SDKs, lihat Contoh kode di Referensi API Amazon S3.

Untuk skenario interaktif yang mendemonstrasikan fitur Object Lock yang berbeda menggunakan AWS SDKs, lihat Contoh kode di Referensi API Amazon S3.

Untuk informasi umum tentang penggunaan yang berbeda AWS SDKs, lihat Mengembangkan dengan Amazon S3 menggunakan Referensi API Amazon S3 AWS SDKs di Amazon S3.

Mengatur atau memodifikasi penyimpanan yang sah pada objek S3

Anda dapat menyetel atau menghapus penahanan hukum pada objek S3 dengan menggunakan konsol Amazon S3 AWS CLI, AWS SDKs, atau Amazon S3 REST API.

⚠ Important

- Jika Anda ingin mengatur penyimpanan yang sah pada suatu objek, bucket objek harus sudah mengaktifkan Kunci Objek.
- Jika Anda memiliki versi objek PUT yang memiliki mode dan periode retensi individual eksplisit dalam bucket, pengaturan Kunci Objek individual versi objek akan menimpa pengaturan retensi properti bucket apa pun.

Untuk informasi selengkapnya, lihat the section called “Penahanan legal”.

Menggunakan konsol S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Di daftar Bucket, pilih nama bucket yang berisi objek yang ingin Anda tetapkan atau ubah penyimpanan sahnya.
4. Dalam daftar Objek, pilih objek yang ingin Anda atur atau ubah penyimpanan sahnya.
5. Pada halaman Properti objek, temukan bagian Penyimpanan yang Sah Kunci Objek, lalu pilih Edit.
6. Pilih Aktifkan untuk menetapkan penyimpanan yang sah atau Nonaktifkan untuk menghapus penyimpanan yang sah.
7. Pilih Simpan perubahan.

Menggunakan AWS CLI

Contoh `put-object-legal-hold` berikut menetapkan penyimpanan yang sah pada objek *my-image.fs* dalam bucket bernama *amzn-s3-demo-bucket1*:

```
aws s3api put-object-legal-hold --bucket amzn-s3-demo-bucket1 --key my-image.fs --
legal-hold="Status=ON"
```

Contoh `put-object-legal-hold` berikut menghapus penyimpanan yang sah pada objek *my-image.fs* dalam bucket bernama *amzn-s3-demo-bucket1*:

```
aws s3api put-object-legal-hold --bucket amzn-s3-demo-bucket1 --key my-image.fs --
legal-hold="Status=OFF"
```

Untuk informasi dan contoh selengkapnya, lihat put-object-legal-holddalam AWS CLI Command Reference.

> **Note**
>
> Anda dapat menjalankan AWS CLI perintah dari konsol dengan menggunakan AWS CloudShell. AWS CloudShell adalah shell pra-otentikasi berbasis browser yang dapat Anda

luncurkan langsung dari file. AWS Management Console Untuk informasi lebih lanjut, lihat Apa itu CloudShell? dalam AWS CloudShell User Guide.

Penggunaan API REST

Anda dapat menggunakan API REST untuk menetapkan atau memodifikasi penyimpanan yang sah pada objek. Untuk informasi selengkapnya, silakan lihat PutObjectLegalHold dalam Referensi API Layanan Penyimpanan Sederhana Amazon.

Menggunakan AWS SDKs

Untuk contoh cara menyetel penahanan hukum pada objek dengan AWS SDKs, lihat Contoh kode di Referensi API Amazon S3.

Untuk contoh cara mendapatkan status penahanan hukum saat ini dengan AWS SDKs, lihat Contoh kode di Referensi API Amazon S3.

Untuk skenario interaktif yang mendemonstrasikan fitur Object Lock yang berbeda menggunakan AWS SDKs, lihat Contoh kode di Referensi API Amazon S3.

Untuk informasi umum tentang penggunaan yang berbeda AWS SDKs, lihat Mengembangkan dengan Amazon S3 menggunakan Referensi API Amazon S3 AWS SDKs di Amazon S3.

Menetapkan atau memodifikasi periode retensi pada objek S3

Anda dapat mengatur atau mengubah periode retensi pada objek S3 menggunakan konsol Amazon S3 AWS CLI, AWS SDKs, atau Amazon S3 REST API.

Important

- Jika Anda ingin mengatur periode retensi pada objek, bucket objek harus sudah mengaktifkan Kunci Objek.
- Jika Anda memiliki versi objek PUT yang memiliki mode dan periode retensi individual eksplisit dalam bucket, pengaturan Kunci Objek individual versi objek akan menimpa pengaturan retensi properti bucket apa pun.
- Satu-satunya cara untuk menghapus objek di bawah mode kepatuhan sebelum tanggal retensi berakhir adalah dengan menghapus yang terkait Akun AWS.

Untuk informasi selengkapnya, lihat Periode retensi.

Menggunakan konsol S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Di daftar Bucket, pilih nama bucket yang berisi objek yang ingin Anda tetapkan atau ubah periode retensinya.
4. Dalam daftar Objek, pilih objek yang ingin Anda tetapkan atau ubah periode retensinya.
5. Pada halaman Properti object, temukan bagian Retensi Kunci Objek, lalu pilih Edit.
6. Di bawah Retensi, pilih Aktifkan untuk menetapkan periode retensi atau Nonaktifkan untuk menghapus periode retensi.
7. Jika Anda memilih Aktifkan, di bawah Mode retensi, pilih mode Tata Kelola atau mode Kepatuhan. Untuk informasi selengkapnya, lihat Mode retensi.
8. Di bawah Pertahankan hingga tanggal, pilih tanggal yang Anda inginkan untuk mengakhiri periode retensi. Selama periode ini, objek Anda dilindungi WORM dan tidak dapat ditimpa atau dihapus. Untuk informasi selengkapnya, lihat Periode retensi.
9. Pilih Simpan perubahan.

Menggunakan AWS CLI

Contoh `put-object-retention` berikut menetapkan periode retensi pada objek *`my-image.fs`* dalam bucket berama *`amzn-s3-demo-bucket1`* hingga 1 Januari 2025:

```
aws s3api put-object-retention --bucket amzn-s3-demo-bucket1 --key my-image.fs --
retention='{ "Mode": "GOVERNANCE", "RetainUntilDate": "2025-01-01T00:00:00" }'
```

Untuk informasi dan contoh selengkapnya, lihat put-object-retentiondalam AWS CLI Command Reference.

> **Note**
>
> Anda dapat menjalankan AWS CLI perintah dari konsol dengan menggunakan AWS CloudShell. AWS CloudShell adalah shell pra-otentikasi berbasis browser yang dapat Anda

luncurkan langsung dari file. AWS Management Console Untuk informasi lebih lanjut, lihat Apa itu CloudShell? dalam AWS CloudShell User Guide.

Penggunaan API REST

Anda dapat menggunakan API REST untuk menetapkan periode retensi pada objek. Untuk informasi selengkapnya, silakan lihat PutObjectRetention dalam Referensi API Layanan Penyimpanan Sederhana Amazon.

Menggunakan AWS SDKs

Untuk contoh cara menyetel periode retensi pada objek dengan AWS SDKs, lihat Contoh kode di Referensi API Amazon S3.

Untuk contoh cara mendapatkan periode retensi pada objek dengan AWS SDKs, lihat Contoh kode di Referensi API Amazon S3.

Untuk skenario interaktif yang mendemonstrasikan fitur Object Lock yang berbeda menggunakan AWS SDKs, lihat Contoh kode di Referensi API Amazon S3.

Untuk informasi umum tentang penggunaan yang berbeda AWS SDKs, lihat Mengembangkan dengan Amazon S3 menggunakan Referensi API Amazon S3 AWS SDKs di Amazon S3.

Menetapkan atau memodifikasi periode retensi default pada bucket S3

Anda dapat menyetel atau mengubah periode retensi default pada bucket S3 menggunakan konsol Amazon S3 AWS CLI, AWS SDKs, atau Amazon S3 REST API. Anda menentukan durasi, baik dalam hitungan hari atau tahun, untuk lamanya perlindungan terhadap setiap versi objek yang ditempatkan di dalam bucket.

⚠ Important

- Jika Anda ingin menetapkan periode retensi default pada bucket, bucket harus sudah mengaktifkan Kunci Objek.
- Jika Anda memiliki versi objek PUT yang memiliki mode dan periode retensi individual eksplisit dalam bucket, pengaturan Kunci Objek individual versi objek akan menimpa pengaturan retensi properti bucket apa pun.

- Satu-satunya cara untuk menghapus objek di bawah mode kepatuhan sebelum tanggal retensi berakhir adalah dengan menghapus yang terkait Akun AWS.

Untuk informasi selengkapnya, lihat Periode retensi.

Menggunakan konsol S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Di daftar Bucket, pilih nama bucket yang ingin Anda tetapkan atau ubah periode retensi defaultnya.
4. Pilih tab Properti.
5. Di bawah Properti, gulir ke bawah ke bagian Kunci Objek, lalu pilih Edit.
6. Di bawah Retensi default, pilih Aktifkan untuk menetapkan retensi default atau Nonaktifkan untuk menghapus retensi default.
7. Jika Anda memilih Aktifkan, di bawah Mode retensi, pilih mode Tata Kelola atau mode Kepatuhan. Untuk informasi selengkapnya, lihat Mode retensi.
8. Di bawah Periode retensi default, pilih jumlah hari atau tahun yang Anda inginkan untuk periode retensi. Objek yang ditempatkan di bucket ini akan terkunci selama beberapa hari atau tahun. Untuk informasi selengkapnya, lihat Periode retensi.
9. Pilih Simpan perubahan.

Menggunakan AWS CLI

Contoh `put-object-lock-configuration` perintah berikut menetapkan periode retensi Kunci Objek 50 hari pada bucket yang bernama *amzn-s3-demo-bucket1* menggunakan mode kepatuhan:

```
aws s3api put-object-lock-configuration --bucket amzn-s3-demo-bucket1 --object-
lock-configuration='{ "ObjectLockEnabled": "Enabled", "Rule": { "DefaultRetention":
 { "Mode": "COMPLIANCE", "Days": 50 }}}'
```

Contoh `put-object-lock-configuration` berikut menghapus konfigurasi retensi default pada bucket:

```
aws s3api put-object-lock-configuration --bucket amzn-s3-demo-bucket1 --object-lock-
configuration='{ "ObjectLockEnabled": "Enabled"}'
```

Untuk informasi dan contoh selengkapnya, lihat put-object-lock-configurationdalam AWS CLI Command Reference.

> **Note**
>
> Anda dapat menjalankan AWS CLI perintah dari konsol dengan menggunakan AWS CloudShell. AWS CloudShell adalah shell pra-otentikasi berbasis browser yang dapat Anda luncurkan langsung dari file. AWS Management Console Untuk informasi lebih lanjut, lihat Apa itu CloudShell? dalam AWS CloudShell User Guide.

Penggunaan API REST

Anda dapat menggunakan REST API untuk menyetel periode retensi default pada bucket S3 yang ada. Untuk informasi selengkapnya, silakan lihat PutObjectLockConfiguration dalam Referensi API Layanan Penyimpanan Sederhana Amazon.

Menggunakan AWS SDKs

Untuk contoh cara menyetel periode retensi default pada bucket S3 yang ada dengan AWS SDKs, lihat Contoh kode di Referensi API Amazon S3.

Untuk skenario interaktif yang mendemonstrasikan fitur Object Lock yang berbeda menggunakan AWS SDKs, lihat Contoh kode di Referensi API Amazon S3.

Untuk informasi umum tentang penggunaan yang berbeda AWS SDKs, lihat Mengembangkan dengan Amazon S3 menggunakan Referensi API Amazon S3 AWS SDKs di Amazon S3.

Mencadangkan data Amazon S3 Anda

Amazon S3 terintegrasi secara native dengan AWS Backup, layanan berbasis kebijakan yang dikelola sepenuhnya yang dapat Anda gunakan untuk menentukan kebijakan pencadangan secara terpusat guna melindungi data Anda di Amazon S3. Setelah Anda menentukan kebijakan pencadangan dan menetapkan sumber daya Amazon S3 ke kebijakan AWS Backup , mengotomatiskan pembuatan cadangan Amazon S3 dan menyimpan cadangan dengan aman di brankas cadangan terenkripsi yang Anda tetapkan dalam paket cadangan.

Saat menggunakan AWS Backup untuk Amazon S3, Anda dapat melakukan tindakan berikut:

- Membuat pencadangan berkelanjutan dan pencadangan berkala. Pencadangan berkelanjutan berguna untuk point-in-time pemulihan, dan pencadangan berkala berguna untuk memenuhi kebutuhan retensi data jangka panjang Anda.
- Mengotomatiskan penjadwalan dan retensi pencadangan dengan mengonfigurasi kebijakan pencadangan secara terpusat.
- Memulihkan pencadangan data Amazon S3 dari waktu tertentu yang telah Anda tentukan.

Bersamaan dengan itu AWS Backup, Anda dapat menggunakan S3 Versioning dan S3 Replikasi untuk membantu memulihkan dari penghapusan yang tidak disengaja dan melakukan operasi pemulihan diri Anda sendiri.

Prasyarat

Anda harus mengaktifkan S3 Versioning pada bucket Anda sebelum AWS Backup dapat mencadangkannya.

> **Note**
>
> Kami merekomendasikan Anda untuk menetapkan aturan kedaluwarsa siklus hidup untuk bucket dengan kemampuan Penentuan Versi aktif yang sedang dicadangkan. Jika Anda tidak menetapkan periode kedaluwarsa siklus hidup, biaya penyimpanan Amazon S3 dapat mengalami peningkatan karena AWS Backup mempertahankan semua versi data Amazon S3 Anda.

Memulai

AWS Backup Untuk memulai Amazon S3, lihat Membuat cadangan Amazon S3 di Panduan Pengembang.AWS Backup

Pembatasan dan batasan

Untuk mempelajari tentang batasan, lihat Membuat pencadangan Amazon S3 di Panduan Developer AWS Backup .

Optimalisasi Biaya

Amazon S3 menawarkan berbagai fitur dan kelas penyimpanan untuk membantu Anda mengoptimalkan biaya di seluruh siklus hidup data Anda. Kelas penyimpanan menawarkan fleksibilitas untuk mengelola biaya Anda, dengan menyediakan tingkat akses data yang berbeda dengan biaya yang sesuai, tanpa biaya di muka atau komitmen untuk berapa banyak konten yang Anda simpan. Seperti AWS layanan lainnya, Anda membayar saat Anda pergi dan hanya membayar untuk apa yang Anda gunakan.

Kelas penyimpanan Amazon S3 dibuat khusus untuk menyediakan penyimpanan biaya terendah untuk pola akses yang berbeda. Ini termasuk:

- Standar S3 untuk penyimpanan tujuan umum dari data yang sering diakses.
- Amazon S3 Express One Zone untuk data berkinerja tinggi yang sering diakses di Zona Ketersediaan Tunggal.
- S3 Intelligent-Tiering untuk secara otomatis mengoptimalkan biaya untuk data dengan pola akses yang tidak diketahui atau berubah.
- S3 Standard-IA (S3 Standard-IA) dan S3 One Zone-IA (S3 One Zone-IA) untuk data yang berumur panjang, tetapi lebih jarang diakses.
- S3 Glacier Instant Retrieval untuk data arsip yang membutuhkan akses segera.
- S3 Glacier untuk data arsip yang tidak memerlukan akses langsung tetapi membutuhkan fleksibilitas untuk mengambil set data besar tanpa biaya.
- S3 Glacier Deep Archive untuk arsip jangka panjang dan pelestarian digital dengan biaya penyimpanan terendah di cloud.

Anda dapat memindahkan objek ke kelas penyimpanan yang paling hemat biaya kapan saja. Selain itu, Amazon S3 menyediakan fitur untuk mengelola siklus hidup data Anda. Misalnya, Anda dapat menggunakan konfigurasi Siklus Hidup S3 untuk mengotomatiskan objek transisi ke kelas penyimpanan yang lebih hemat biaya, atau menghapus objek kedaluwarsa secara otomatis berdasarkan aturan yang Anda tentukan.

Fitur seperti Analisis Kelas Penyimpanan S3, penandaan alokasi biaya, dan laporan penagihan dan penggunaan membantu Anda menganalisis pola biaya dan penggunaan Anda.

Topik

- Pelaporan penagihan dan penggunaan untuk Amazon S3
- Memahami dan mengelola kelas penyimpanan Amazon S3
- Mengelola siklus hidup objek

Pelaporan penagihan dan penggunaan untuk Amazon S3

Saat menggunakan Amazon S3, Anda tidak perlu membayar biaya di muka atau berkomitmen untuk berapa banyak konten yang akan Anda simpan. Seperti yang lain Layanan AWS, Anda membayar saat Anda pergi dan hanya membayar untuk apa yang Anda gunakan.

AWS memberikan laporan berikut untuk Amazon S3:

- Laporan penagihan — Beberapa laporan yang memberikan tampilan tingkat tinggi dari semua aktivitas Layanan AWS yang Anda gunakan, termasuk Amazon S3. AWS selalu menagih pemilik ember S3 untuk biaya Amazon S3, kecuali ember dibuat sebagai ember Peminta Pembayaran. Untuk informasi selengkapnya tentang Pembayaran Pemohon, lihat Menggunakan Requester Pays ember tujuan umum untuk transfer dan penggunaan penyimpanan. Untuk informasi selengkapnya tentang laporan penagihan, lihat AWS Billing laporan untuk Amazon S3.
- Laporan penggunaan–Ringkasan aktivitas untuk layanan tertentu, yang dikumpulkan berdasarkan jam, hari, atau bulan. Anda dapat memilih jenis penggunaan dan pengoperasian yang akan disertakan. Anda juga dapat memilih cara pengumpulan data. Untuk informasi selengkapnya, lihat AWS laporan penggunaan untuk Amazon S3.

Topik berikut menyediakan informasi tentang pelaporan penagihan dan penggunaan untuk Amazon S3.

Topik

- Menggunakan tanda alokasi biaya bucket S3
- AWS Billing laporan untuk Amazon S3
- AWS laporan penggunaan untuk Amazon S3
- Memahami laporan AWS penagihan dan penggunaan Anda untuk Amazon S3
- Penagihan untuk respons kesalahan Amazon S3

Menggunakan tanda alokasi biaya bucket S3

Untuk melacak biaya penyimpanan atau kriteria lain untuk proyek individu atau kelompok proyek, beri label pada bucket Amazon S3 Anda menggunakan tanda alokasi biaya. Tanda alokasi biaya adalah pasangan nilai kunci yang Anda kaitkan dengan bucket S3. Setelah Anda mengaktifkan tag alokasi biaya, AWS gunakan tag untuk mengatur biaya sumber daya Anda pada laporan alokasi biaya Anda. Tanda alokasi biaya hanya dapat digunakan untuk melabeli bucket. Untuk informasi tentang tanda yang digunakan untuk melabeli objek, lihat Mengategorikan penyimpanan Anda menggunakan tag.

Laporan alokasi biaya mencantumkan AWS penggunaan akun Anda berdasarkan kategori produk dan pengguna akun tertaut. Laporan ini berisi item baris yang sama dengan laporan penagihan terperinci (lihat Memahami laporan AWS penagihan dan penggunaan Anda untuk Amazon S3) dan kolom tambahan untuk kunci tanda Anda.

AWS menyediakan dua jenis tag alokasi biaya, tag AWS-generated dan tag yang ditentukan pengguna. AWS mendefinisikan, membuat, dan menerapkan `createdBy` tag AWS yang dihasilkan untuk Anda setelah peristiwa Amazon CreateBucket S3. Anda menentukan, membuat, dan menerapkan tanda yang ditentukan pengguna ke bucket S3.

Anda harus mengaktifkan kedua jenis tanda secara terpisah di konsol Manajemen Penagihan dan Biaya sebelum tanda tersebut muncul di laporan penagihan Anda. Untuk informasi selengkapnya tentang tag AWS yang dihasilkan, lihat Tag Alokasi Biaya AWS yang Dihasilkan.

- Untuk membuat tanda di konsol, lihat Melihat properti untuk bucket tujuan umum S3.
- Untuk membuat tanda menggunakan API Amazon S3, lihat MEMBERI Penanda Bucket dalam Referensi API Amazon Simple Storage Service.
- Untuk membuat tag menggunakan AWS CLI, lihat put-bucket-taggingdi AWS CLI Command Reference.
- Untuk informasi selengkapnya tentang mengaktifkan tanda, lihat Menggunakan tanda alokasi biaya di Panduan Pengguna AWS Billing .

Tanda alokasi biaya yang ditentukan pengguna

Tanda alokasi biaya yang ditentukan pengguna memiliki komponen berikut:

- Kunci tanda. Kunci tanda adalah nama tanda. Misalnya, pada tanda project/Trinity, project adalah kuncinya. Kunci tanda adalah string yang peka huruf besar/kecil yang dapat berisi 1 hingga 128 karakter Unicode.

- Nilai tanda. Nilai tanda adalah string yang diperlukan. Misalnya, dalam tanda project/Trinity, Trinity adalah nilainya. Nilai tanda adalah string yang peka huruf besar/kecil yang dapat berisi 0 hingga 256 karakter Unicode.

Untuk detail tentang karakter yang diizinkan untuk tanda yang ditentukan pengguna dan batasan lainnya, lihat Pembatasan Tanda yang Ditentukan Pengguna di Panduan Pengguna AWS Billing . Untuk informasi selengkapnya tentang tanda yang ditentukan pengguna, lihat Tanda Alokasi Biaya yang Ditentukan Pengguna di Panduan Pengguna AWS Billing .

Tanda bucket S3

Setiap bucket S3 memiliki rangkaian tanda. Rangkaian tanda berisi semua tanda yang ditetapkan pada bucket tersebut. Rangkaian tanda dapat berisi 50 tanda atau kosong. Kunci dalam rangkaian tanda harus unik, tetapi nilai dalam rangkaian tanda tidak harus unik. Misalnya, Anda dapat memiliki nilai yang sama dalam kumpulan tag bernamaproject/Trinity and cost-center/Trinity.

Dalam sebuah bucket, jika Anda menambahkan tanda yang memiliki kunci sama dengan tanda yang ada, nilai yang baru akan menimpa nilai yang lama.

AWS tidak menerapkan makna semantik apa pun pada tag Anda. Kami menafsirkan tanda hanya sebagai string karakter.

Untuk menambahkan, membuat daftar, mengedit, atau menghapus tag, Anda dapat menggunakan konsol Amazon S3, AWS Command Line Interface (AWS CLI), atau Amazon S3 API.

Info Selengkapnya

- Menggunakan Tanda Alokasi Biaya di Panduan Pengguna AWS Billing
- Memahami laporan AWS penagihan dan penggunaan Anda untuk Amazon S3
- AWS Billing laporan untuk Amazon S3

AWS Billing laporan untuk Amazon S3

Tagihan bulanan Anda dari AWS memisahkan informasi penggunaan dan biaya Anda berdasarkan Layanan AWS dan fungsinya. Ada beberapa AWS Billing laporan yang tersedia: laporan bulanan, laporan alokasi biaya, dan laporan penagihan terperinci. Untuk informasi tentang cara melihat laporan penagihan, lihat Melihat Tagihan Anda di Panduan Pengguna AWS Billing .

Untuk melacak AWS penggunaan Anda dan memberikan perkiraan biaya yang terkait dengan akun Anda, Anda dapat mengaturnya AWS Cost and Usage Reports. Untuk informasi lebih lanjut, lihat Apa itu AWSCost and Usage Reports? dalam Panduan Ekspor AWS Data.

Anda juga dapat mengunduh laporan penggunaan yang akan memberikan informasi yang lebih lengkap tentang penggunaan penyimpanan Amazon S3 dibandingkan laporan penagihan. Untuk informasi selengkapnya, lihat AWS laporan penggunaan untuk Amazon S3.

Tabel berikut mencantumkan biaya yang terkait dengan penggunaan Amazon S3.

Biaya	Komentar
Penyimpanan	Anda membayar untuk penyimpanan objek dalam bucket S3. Tarif yang dikenakan biaya tergantung pada ukuran objek Anda, berapa lama Anda menyimpan objek selama sebulan, dan kelas penyimpanan. Amazon S3 menawarkan kelas penyimpanan berikut: Standar S3, S3 Express One Zone, S3 Intelligent-Tiering, S3 Standard-IA (IA untuk akses yang jarang), S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive, atau Reduced Redundancy Storage (RRS). Untuk informasi selengkapnya tentang kelas penyimpanan, lihat Memahami dan mengelola kelas penyimpanan Amazon S3. Ketahuilah bahwa jika Anda mengaktifkan Versi S3, Anda dikenakan biaya untuk setiap versi objek yang dipertahankan. Untuk informasi selengkapnya tentang penentuan versi, lihat Cara kerja Penentuan Versi S3.
Ember tujuan umum	Anda tidak ditagih untuk 2000 ember tujuan umum pertama yang Anda buat di akun Anda. Namun, ada tarif per ember untuk setiap bucket yang Anda buat di luar tahun 2000

Biaya	Komentar
	pertama. Tarif ini ditagih per ember/bulan. Untuk informasi tentang harga bucket tujuan umum, lihat Harga Amazon S3.
Pemantauan dan otomatisasi	Anda membayar biaya pemantauan dan otomatisasi bulanan per objek yang disimpan di kelas penyimpanan S3 Intelligent-Tiering untuk memantau pola akses dan memindahkan objek di antara tingkat akses di S3 Intelligent-Tiering.
Permintaan	Anda membayar permintaan, misalnya, `GET` permintaan, yang dibuat terhadap ember dan objek S3 Anda. Ini termasuk permintaan siklus hidup. Tarif untuk permintaan tergantung pada jenis permintaan yang Anda buat. Untuk informasi tentang harga permintaan, lihat Harga Amazon S3.
Pengambilan	Anda membayar untuk mengambil objek yang disimpan dalam penyimpanan S3 Standard-IA, S3 One Zone-IA, S3 Glacier Instant, S3 Glacier Flexible Retrieval, dan penyimpanan S3 Glacier Deep Archive.
Menghapus lebih awal	Jika Anda menghapus objek yang disimpan di penyimpanan S3 Standard-IA, S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval, atau S3 Glacier Deep Archive sebelum melewati target penyimpan an minimum, Anda akan membayar biaya penghapusan awal untuk objek tersebut.
Manajemen penyimpanan	Anda membayar untuk fitur manajemen penyimpanan (Inventaris Amazon S3, analitik, dan penandaan objek) yang diaktifkan di bucket akun Anda.

Biaya	Komentar
Bandwidth	Anda membayar untuk semua bandwidth ke dan dari Amazon S3, kecuali untuk: • Data yang ditransfer dari internet • Data ditransfer ke instans Amazon Elastic Compute Cloud (Amazon EC2), jika instans Wilayah AWS sama dengan bucket S3 • Data ditransfer ke Amazon CloudFront (CloudFront) Anda juga membayar biaya untuk data apa pun yang ditransfer menggunakan Amazon S3 Transfer Acceleration.

Untuk informasi terperinci tentang biaya penggunaan Amazon S3 untuk penyimpanan, transfer data, dan layanan, lihat Harga Amazon S3 dan Amazon S3. FAQs

Untuk informasi tentang memahami kode dan singkatan yang digunakan dalam laporan penagihan dan penggunaan untuk Amazon S3, lihat. Memahami laporan AWS penagihan dan penggunaan Anda untuk Amazon S3

Info selengkapnya

- AWS laporan penggunaan untuk Amazon S3
- Menggunakan tanda alokasi biaya bucket S3
- AWS Billing dan Manajemen Biaya
- Harga Amazon S3

AWS laporan penggunaan untuk Amazon S3

Saat mengunduh laporan penggunaan, Anda dapat memilih untuk menggabungkan data penggunaan berdasarkan jam, hari, atau bulan. Laporan penggunaan Amazon S3 mencantumkan operasi

berdasarkan jenis penggunaan dan. Wilayah AWS Untuk laporan yang lebih mendetail tentang penggunaan penyimpanan Amazon S3 Anda, unduh laporan penggunaan AWS yang dibuat secara dinamis. Anda dapat memilih jenis penggunaan, operasi, dan periode waktu yang akan disertakan. Anda juga dapat memilih cara pengumpulan data. Untuk informasi selengkapnya tentang laporan penggunaan, lihat Laporan AWS Penggunaan di Panduan Pengguna Ekspor AWS Data.

Laporan penggunaan Amazon S3 mencakup informasi berikut:

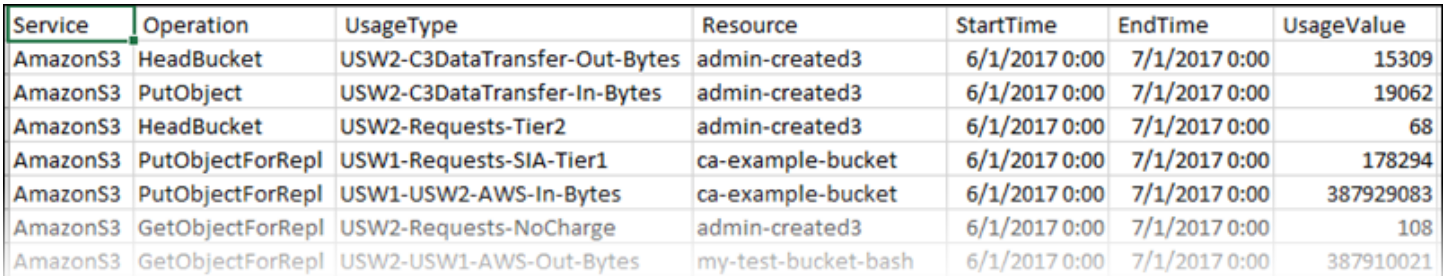
- Layanan–Amazon S3
- Operasi–Operasi yang dilakukan pada bucket atau objek Anda. Untuk penjelasan terperinci tentang operasi Amazon S3, lihat Melacak Operasi dalam Laporan Penggunaan Anda.
- UsageType – Salah satu nilai berikut:
  - Kode yang mengidentifikasi jenis penyimpanan
  - Kode yang mengidentifikasi jenis permintaan
  - Kode yang mengidentifikasi jenis pengambilan
  - Kode yang mengidentifikasi jenis transfer data
  - Kode yang mengidentifikasi penghapusan awal dari penyimpanan S3 Intelligent-Tiering, S3 Standard-IA, S3 One Zone-Infrequent Access (S3 One Zone-IA), S3 Glacier Flexible Retrieval, atau penyimpanan S3 Glacier Deep Archive
  - `StorageObjectCount`–Jumlah objek yang disimpan dalam bucket tertentu

  Untuk penjelasan terperinci tentang jenis penggunaan Amazon S3, lihat Memahami laporan AWS penagihan dan penggunaan Anda untuk Amazon S3.
- Resource — Nama bucket atau tabel yang terkait dengan penggunaan yang tercantum.
- StartTime – Waktu mulai pada hari saat penggunaan berlaku, dalam Waktu Universal Terkoordinasi (UTC).
- EndTime— Waktu akhir hari penggunaan berlaku, di Coordinated Universal Time (UTC).
- UsageValue— Salah satu nilai volume berikut. Unit pengukuran yang umum untuk data adalah gigabyte (GB). Namun, tergantung pada layanan dan laporannya, terabyte (TB) mungkin juga akan digunakan.
  - Jumlah permintaan selama periode waktu yang ditentukan
  - Jumlah data yang ditransfer
  - Jumlah data yang disimpan dalam jam tertentu

- Jumlah data yang terkait dengan restorasi dari penyimpanan S3 Standard-IA, S3 One Zone-IA, S3 Glacier Flexible Retrieval, atau S3 Glacier Deep Archive

> **Tip**
>
> Untuk informasi terperinci tentang setiap permintaan yang diterima Amazon S3 untuk objek Anda, aktifkan logging akses server untuk bucket Anda. Untuk informasi selengkapnya, lihat Pencatatan permintaan dengan pencatatan akses server.

Anda dapat mengunduh laporan penggunaan dalam bentuk file XML atau file nilai yang dipisahkan oleh koma (CSV). Berikut ini adalah contoh laporan penggunaan CSV yang dibuka di aplikasi spreadsheet.



Service	Operation	UsageType	Resource	StartTime	EndTime	UsageValue
AmazonS3	HeadBucket	USW2-C3DataTransfer-Out-Bytes	admin-created3	6/1/2017 0:00	7/1/2017 0:00	15309
AmazonS3	PutObject	USW2-C3DataTransfer-In-Bytes	admin-created3	6/1/2017 0:00	7/1/2017 0:00	19062
AmazonS3	HeadBucket	USW2-Requests-Tier2	admin-created3	6/1/2017 0:00	7/1/2017 0:00	68
AmazonS3	PutObjectForRepl	USW1-Requests-SIA-Tier1	ca-example-bucket	6/1/2017 0:00	7/1/2017 0:00	178294
AmazonS3	PutObjectForRepl	USW1-USW2-AWS-In-Bytes	ca-example-bucket	6/1/2017 0:00	7/1/2017 0:00	387929083
AmazonS3	GetObjectForRepl	USW2-Requests-NoCharge	admin-created3	6/1/2017 0:00	7/1/2017 0:00	108
AmazonS3	GetObjectForRepl	USW2-USW1-AWS-Out-Bytes	my-test-bucket-bash	6/1/2017 0:00	7/1/2017 0:00	387910021

Untuk informasi selengkapnya, lihat Memahami laporan AWS penagihan dan penggunaan Anda untuk Amazon S3.

Mengunduh Laporan AWS Penggunaan

Anda dapat mengunduh laporan penggunaan sebagai file XML/CSV.

Untuk mengunduh laporan penggunaan

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah judul, pilih nama pengguna atau ID akun Anda, lalu pilih Billing and Cost Management.
3. Di panel navigasi, pilih Halaman Legacy dan pilih Laporan biaya dan penggunaan.
4. Di bawah Laporan AWS Penggunaan, pilih Buat Laporan Penggunaan.
5. Pada halaman Laporan penggunaan Unduh, pilih pengaturan berikut:
   - Layanan - Pilih Amazon Simple Storage Service.

- Jenis Penggunaan–Untuk penjelasan detail tentang jenis penggunaan Amazon S3, lihat Memahami laporan AWS penagihan dan penggunaan Anda untuk Amazon S3.
- Operasi–Untuk penjelasan detail tentang operasi Amazon S3, lihat Melacak Operasi dalam Laporan Penggunaan Anda.
- Periode Waktu–Periode waktu yang ingin dicakup dalam laporan.
- Granularitas Laporan–Apakah Anda ingin agar laporan menyertakan subtotal berdasarkan jam, hari, atau bulan.

6. Pilih Unduh, pilih format unduhan (Laporan XHTML atau Laporan CSV), lalu ikuti petunjuk untuk membuka atau menyimpan laporan.

Info selengkapnya

- Memahami laporan AWS penagihan dan penggunaan Anda untuk Amazon S3
- AWS Billing laporan untuk Amazon S3

Memahami laporan AWS penagihan dan penggunaan Anda untuk Amazon S3

Laporan penagihan dan penggunaan Amazon S3 menggunakan kode dan singkatan. Untuk jenis penggunaan dalam tabel berikut, ganti, *regionregion1*, dan *region2* dengan singkatan dari daftar ini:

- APE1: Asia Pasifik (Hong Kong)
- APN1: Asia Pasifik (Tokyo)
- APN2: Asia Pasifik (Seoul)
- APN3: Asia Pasifik (Osaka)
- APS1: Asia Pasifik (Singapura)
- APS2: Asia Pasifik (Sydney)
- APS3: Asia Pasifik (Mumbai)
- APS4: Asia Pasifik (Jakarta)
- APS5: Asia Pasifik (Hyderabad)
- APS6: Asia Pasifik (Melbourne)
- APS7: Asia Pasifik (Malaysia)

- APS9: Asia Pasifik (Thailand)
- CAN1: Kanada (Tengah)
- CAN2: Kanada Barat (Calgary)
- CNN1: Tiongkok (Beijing)
- CNW1: Tiongkok (Ningxia)
- AFS1: Afrika (Cape Town)
- EUC2: Eropa (Zurich)
- EUN1: Eropa (Stockholm)
- EUS2: Eropa (Spanyol)
- EUC1: Eropa (Frankfurt)
- EU: Eropa (Irlandia)
- EUS1: Eropa (Milan)
- EUW2: Eropa (London)
- EUW3: Eropa (Paris)
- ILC1: Israel (Israel)
- MEC1: Timur Tengah (UEA)
- MES1: Timur Tengah (Bahrain)
- MXC1: Meksiko (Tengah)
- SAE1: South America (São Paulo)
- UGW1: AWS GovCloud (AS-Barat)
- UGE1: AWS GovCloud (AS-Timur)
- USE1 (atau tidak ada awalan): US East (Virginia N.)
- USE2: AS Timur (Ohio)
- USW1: AS Barat (California N.)
- USW2: AS Barat (Oregon)

Untuk tipe penggunaan S3 Multi-Region Access Points dalam tabel berikut, ganti *`regiongroup1`* dan *`regiongroup2`* dengan singkatan dari daftar ini:

- AP: Asia Pasifik
- AU: Australia

- EU: Eropa
- IN: India
- NA: Amerika Utara
- SA: Amerika Selatan

Kelompok wilayah adalah pengelompokan geografis dari beberapa. Wilayah AWS Untuk informasi selengkapnya, lihat Wilayah dan Zona Ketersediaan. Untuk informasi tentang harga menurut Wilayah AWS, lihat Harga Amazon S3.

Kolom pertama dalam tabel berikut mencantumkan jenis penggunaan yang muncul dalam laporan penagihan dan penggunaan Anda. Unit pengukuran yang umum untuk data adalah gigabyte (GB). Namun, tergantung pada layanan dan laporannya, terabyte (TB) mungkin juga akan digunakan.

Jenis Penggunaan	Unit	Granularitas	Deskripsi
region1-*region2*-AWS-In-A Bytes	GB	Per Jam	Jumlah data yang dipercepat yang ditransfer ke *region1* dari *region2*
region1-*region2*-AWS-In-A Bytes-T1	GB	Per Jam	Jumlah data akselerasi T1 yang ditransfer ke *region1* dari*region2*, di mana T1 mengacu pada CloudFront permintaan ke titik kehadiran (POPs) di Amerika Serikat, Eropa, dan Jepang
region1-*region2*-AWS-In-A Bytes-T2	GB	Per Jam	Jumlah data akselerasi T2 yang ditransfer ke *region1* dari*region2*, di mana T2 mengacu pada CloudFront permintaan POPs di

Jenis Penggunaan	Unit	Granularitas	Deskripsi
			semua lokasi tepi lainnya AWS
region1-*region2*-AWS-In-Bytes	GB	Per Jam	Jumlah data yang ditransfer *region1* dari *region2*
region1-*region2*-AWS-Out-ABytes	GB	Per Jam	Jumlah data yang dipercepat yang ditransfer dari *region1* ke *region2*
region1-*region2*-AWS-Out-ABytes-T1	GB	Per Jam	Jumlah data akselerasi T1 yang ditransfer dari *region1* ke*region2*, di mana T1 mengacu pada CloudFront permintaan POPs di Amerika Serikat, Eropa, dan Jepang
region1-*region2*-AWS-Out-ABytes-T2	GB	Per Jam	Jumlah data akselerasi T2 yang ditransfer dari *region1* ke*region2*, di mana T2 mengacu pada CloudFront permintaan POPs di semua lokasi tepi lainnya AWS
region1-*region2*-AWS-Out-Bytes	GB	Per Jam	Jumlah data yang ditransfer dari *region1* ke *region2*
region-BatchOperations-Jobs	Hitungan	Per Jam	Jumlah pekerjaan Operasi Batch S3 yang dijalankan

Jenis Penggunaan	Unit	Granularitas	Deskripsi
region-BatchOperations-Objects	Hitung	Per Jam	Jumlah operasi objek yang dilakukan oleh Operasi Batch S3
region-Bulk-Retrieval-Bytes	GB	Per Jam	Jumlah data yang diambil dengan permintaan Massal S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive
region-BytesDeleted-GDA	GB	Bulanan	Jumlah data yang dihapus oleh `DeleteObject` operasi dari penyimpanan S3 Glacier Deep Archive
region-BytesDeleted-GIR	GB	Bulanan	Jumlah data yang dihapus oleh `DeleteObject` operasi dari penyimpanan S3 Glacier Instant Retrieval.
region-BytesDeleted-GLACIER	GB	Bulanan	Jumlah data yang dihapus oleh `DeleteObject` operasi dari penyimpanan S3 Glacier Flexible Retrieval
region-BytesDeleted-INT	GB	Bulanan	Jumlah data yang dihapus oleh `DeleteObject` operasi dari penyimpanan S3 Intelligent-Tiering

Jenis Penggunaan	Unit	Granularitas	Deskripsi
region-BytesDeleted-RRS	GB	Bulanan	Jumlah data yang dihapus oleh DeleteObject operasi dari penyimpanan Reduced Redundancy Storage (RRS)
region-BytesDeleted-SIA	GB	Bulanan	Jumlah data yang dihapus oleh DeleteObject operasi dari penyimpanan IA standar S3
region-BytesDeleted-STANDARD	GB	Bulanan	Jumlah data yang dihapus oleh DeleteObject operasi dari penyimpanan Standar S3
region-BytesDeleted-ZIA	GB	Bulanan	Jumlah data yang dihapus oleh DeleteObject operasi dari penyimpanan IA Zona Satu S3
region-C3DataTransfer-In-Bytes	GB	Per Jam	Jumlah data yang ditransfer ke Amazon S3 dari Amazon EC2 dalam waktu yang sama Wilayah AWS
region-C3DataTransfer-Out-Bytes	GB	Per Jam	Jumlah data yang ditransfer dari Amazon S3 ke Amazon EC2 dalam waktu yang sama Wilayah AWS

Jenis Penggunaan	Unit	Granularitas	Deskripsi
region-CloudFront-In-Bytes	GB	Per Jam	Jumlah data yang ditransfer Wilayah AWS ke suatu CloudFront distribusi
region-CloudFront-Out-Bytes	GB	Per Jam	Jumlah data yang ditransfer dari CloudFront distribusi Wilayah AWS ke suatu
region-DataTransfer-In-Bytes	GB	Per Jam	Jumlah data yang ditransfer ke Amazon S3 dari internet
region-DataTransfer-Out-Bytes	GB	Per Jam	Jumlah data yang ditransfer dari Amazon S3 ke internet[1]
region-DataTransfer-Regional-Bytes	GB	Per Jam	Jumlah data yang ditransfer dari Amazon S3 ke AWS sumber daya yang sama Wilayah AWS
region-EarlyDelete-ByteHrs	GB-Jam	Per Jam	Penggunaan penyimpanan prorata untuk objek yang dihapus dari penyimpanan S3 Glacier Flexible Retrieval sebelum komitmen minimum 90 hari berakhir[2]

Jenis Penggunaan	Unit	Granularitas	Deskripsi
region-EarlyDelete-GDA	GB-Jam	Per Jam	Penggunaan penyimpanan prorata untuk objek yang dihapus dari penyimpanan S3 Glacier Deep Archive sebelum komitmen minimum 180 hari berakhir [2]
region-EarlyDelete-GIR	GB-Jam	Per Jam	Penggunaan penyimpanan prorata untuk objek yang dihapus dari S3 Glacier Instant Retrieval sebelum komitmen minimum 90 hari berakhir.
region-EarlyDelete-GIR-SmObjects	GB-Jam	Per Jam	Penggunaan penyimpanan prorata untuk objek kecil (lebih kecil dari 128 KB) yang dihapus dari S3 Glacier Instant Retrieval sebelum komitmen minimum 90 hari berakhir.
region-EarlyDelete-SIA	GB-Jam	Per Jam	Penggunaan penyimpanan prorata untuk objek yang dihapus dari S3 Standard-IA sebelum komitmen minimum 30 hari berakhir[3]

Jenis Penggunaan	Unit	Granularitas	Deskripsi
region-EarlyDelete-SIA-SmObjects	GB-Jam	Per Jam	Penggunaan penyimpanan prorata untuk objek kecil (lebih kecil dari 128 KB) yang dihapus dari S3 Standard-IA sebelum komitmen minimum 30 hari berakhir[3]
region-EarlyDelete-ZIA	GB-Jam	Per Jam	Penggunaan penyimpanan prorata untuk objek yang dihapus dari S3 One Zone-IA sebelum komitmen minimum 30 hari berakhir[3]
region-EarlyDelete-ZIA-SmObjects	GB-Jam	Per Jam	Penggunaan penyimpanan prorata untuk objek kecil (lebih kecil dari 128 KB) yang dihapus dari S3 One Zone-IA sebelum komitmen minimum 30 hari berakhir[3]
region-Expedited-Retrieval-Bytes	GB	Per Jam	Jumlah data yang diambil dengan permintaan S3 Glacier Flexible Retrieval yang Dipercepat
Global-Bucket-Hrs-FreeTier	Bucket	Bulanan	Jumlah bucket tujuan umum di akun Anda dalam tingkat bebas tingkat akun bucket 2000

Jenis Penggunaan	Unit	Granularitas	Deskripsi
`Global-Bucket-Hrs`	Bucket	Bulanan	Jumlah bucket tujuan umum di akun Anda di luar tingkat bebas tingkat akun bucket 2000
`region``-Inventory-Objects Listed`	Objek	Per Jam	Jumlah objek yang tercantum untuk grup objek (objek dikelompokkan berdasarkan bucket atau prefiks) dengan daftar inventaris
`region``-Metadata-Updates`	Pembaruan	Per Jam	Per biaya pembaruan untuk pembaruan yang diproses oleh S3 Metadata
`region``-Monitoring-Automation-INT`	Objek	Per Jam	Jumlah objek unik yang dipantau dan disusun secara otomatis dalam kelas penyimpanan S3 Intelligent-Tiering
`region``-MRAP-Out-Bytes`	GB	Per Jam	Jumlah data yang ditransfer melalui titik akhir Titik Akses Multi-Wilayah S3 dari bucket di Wilayah (harga perutean data MRAP).

Jenis Penggunaan	Unit	Granularitas	Deskripsi
region-MRAP-In-Bytes	GB	Per Jam	Jumlah data yang ditransfer melalui titik akhir Titik Akses Multi-Wilayah S3 dari bucket di Wilayah (harga perutean data MRAP).
regiongroup1-regiongroup2--MRAP-Out-Bytes	GB	Per Jam	Jumlah data yang ditransfer melalui titik akhir S3 Multi-Region Access Points dari bucket in *regiongroup1* ke klien yang *regiongroup2* berada di luar jaringan. AWS
regiongroup1-regiongroup2--MRAP-In-Bytes	GB	Per Jam	Jumlah data yang ditransfer melalui titik akhir Titik Akses Multi-Wilayah S3 ke bucket in *regiongroup1* dari klien yang *regiongroup2* berada di luar jaringan. AWS
region-OverwriteBytes-Copy-GDA	GB	Bulanan	Jumlah data yang ditimpa oleh `CopyObject` operasi dari penyimpanan S3 Glacier Deep Archive

Jenis Penggunaan	Unit	Granularitas	Deskripsi
region-OverwriteBytes-Copy-GIR	GB	Bulanan	Jumlah data yang ditimpa oleh CopyObject operasi dari penyimpanan S3 Glacier Instant Retrieval.
region-OverwriteBytes-Copy-GLACIER	GB	Bulanan	Jumlah data yang ditimpa oleh CopyObject operasi dari penyimpanan S3 Glacier Flexible Retrieval
region-OverwriteBytes-Copy-INT	GB	Bulanan	Jumlah data yang ditimpa oleh CopyObject operasi dari penyimpanan S3 Intelligent-Tiering
region-OverwriteBytes-Copy-RRS	GB	Bulanan	Jumlah data yang ditimpa oleh CopyObject operasi dari penyimpanan Reduced Redundancy Storage (RRS)
region-OverwriteBytes-Copy-SIA	GB	Bulanan	Jumlah data yang ditimpa oleh CopyObject operasi dari penyimpanan IA standar S3
region-OverwriteBytes-Copy-STANDARD	GB	Bulanan	Jumlah data yang ditimpa oleh CopyObject operasi dari penyimpanan Standar S3

Jenis Penggunaan	Unit	Granularitas	Deskripsi
region-OverwriteBytes-Copy-ZIA	GB	Bulanan	Jumlah data yang ditimpa oleh CopyObject operasi dari penyimpanan IA Zona Satu S3
region-OverwriteBytes-Put-GDA	GB	Bulanan	Jumlah data yang ditimpa oleh PutObject operasi dari penyimpanan S3 Glacier Deep Archive
region-OverwriteBytes-Put-GIR	GB	Bulanan	Jumlah data yang ditimpa oleh PutObject operasi dari penyimpanan S3 Glacier Instant Retrieval.
region-OverwriteBytes-Put-GLACIER	GB	Bulanan	Jumlah data yang ditimpa oleh PutObject operasi dari penyimpanan S3 Glacier Flexible Retrieval
region-OverwriteBytes-Put-INT	GB	Bulanan	Jumlah data yang ditimpa oleh PutObject operasi dari penyimpanan S3 Intelligent-Tiering
region-OverwriteBytes-Put-RRS	GB	Bulanan	Jumlah data yang ditimpa oleh PutObject operasi dari penyimpanan Reduced Redundancy Storage (RRS)

Jenis Penggunaan	Unit	Granularitas	Deskripsi
region-OverwriteBytes-Put-SIA	GB	Bulanan	Jumlah data yang ditimpa oleh PutObject operasi dari penyimpanan IA standar S3
region-OverwriteBytes-Put-STANDARD	GB	Bulanan	Jumlah data yang ditimpa oleh PutObject operasi dari penyimpanan Standar S3
region-OverwriteBytes-Put-ZIA	GB	Bulanan	Jumlah data yang ditimpa oleh PutObject operasi dari penyimpanan IA Zona Satu S3
region1-*region2*-S3RTC-In-Bytes	GB	Bulanan	Jumlah data yang ditransfer untuk Kontrol Waktu Replikasi S3 (S3 RTC) dari *region2* ke *region1* olehPutObject ReplTime ,,,, GetObject ReplTime InitiateMultipartUploadReplTime , UploadPartReplTime dan operasi CompleteMultipartUploadReplTime WriteACLReplTime

Jenis Penggunaan	Unit	Granularitas	Deskripsi
region1-*region2*-S3RTC-Out-Bytes	GB	Bulanan	Jumlah data yang ditransfer untuk Kontrol Waktu Replikasi S3 (S3 RTC) dari *region1* ke *region2* olehPutObject ReplTime ,,,, GetObject ReplTime InitiateMultipartUploadReplTime , UploadPartReplTime dan operasi CompleteMultipartUploadReplTime WriteACLReplTime
region-Requests-GDA-Tier1	Hitungan	Per Jam	JumlahPUT,,,, COPY POST CreateMultipartUpload UploadPart , atau CompleteMultipartUpload permintaan pada objek S3 Glacier Deep Archive 6
region-Requests-GDA-Tier2	Hitungan	Per Jam	Jumlah GET dan HEAD permintaan pada objek S3 Glacier Deep Archive
region-Requests-GDA-Tier3	Hitungan	Per Jam	Jumlah permintaan pemulihan standar S3 Glacier Deep Archive

Jenis Penggunaan	Unit	Granularitas	Deskripsi
region-Requests-GDA-Tier5	Hitung	Per Jam	Jumlah permintaan pemulihan Massal S3 Glacier Deep Archive
region-Requests-GIR-Tier1	Hitung	Per Jam	JumlahPUT,COPY, atau POST permintaan pada objek Pengambilan Instan S3 Glacier.
region-Requests-GIR-Tier2	Hitungan	Per Jam	Jumlah GET dan semua permintaan Non-S3 Glacier Instant Retrieval -Tier1 lainnya pada objek S3 Glacier Instant Retrieval.
region-Requests-GLACIER-Tier1	Hitungan	Per Jam	JumlahPUT,,,, COPY POST CreateMultipartUpload UploadPart , atau CompleteMultipartUpload permintaan pada objek Pengambilan Fleksibel S3 Glacier 6
region-Requests-GLACIER-Tier2	Hitungan	Per Jam	Jumlah GET dan semua permintaan lainnya yang tidak tercantum pada objek Pengambilan Fleksibel S3 Glacier
region-Requests-INT-Tier1	Hitungan	Per Jam	JumlahPUT,COPY, atau POST permintaan pada objek S3 Intelligent-Tiering

Jenis Penggunaan	Unit	Granularitas	Deskripsi
region-Requests-INT-Tier2	Hitungan	Per Jam	Jumlah GET dan semua permintaan non-Tier1 lainnya untuk objek S3 Intelligent-Tiering
region-Requests-SIA-Tier1	Hitungan	Per Jam	JumlahPUT,COPY, atau POST permintaan pada objek IA standar S3
region-Requests-SIA-Tier2	Hitungan	Per Jam	Jumlah GET dan semua permintaan Non-S3 Glacier Instant Retrieval-Tier1 lainnya pada objek IA Standar S3
region-Requests-Tier1	Hitungan	Per Jam	JumlahPUT,COPY, atau POST permintaan untuk Standar S3, RRS, dan tag, ditambah LIST permintaan untuk semua ember dan objek
region-Requests-Tier2	Hitungan	Per Jam	Jumlah GET dan semua permintaan non-Tier1 lainnya
region-Requests-Tier3	Hitungan	Per Jam	Jumlah permintaan siklus hidup ke S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive dan permintaan standar S3 Glacier Flexible Retrieval

Jenis Penggunaan	Unit	Granularitas	Deskripsi
region-Requests-Tier4	Hitung	Per Jam	Jumlah transisi siklus hidup ke penyimpanan S3 Glacier Instant, S3 Intelligent-Tiering, S3 Standard-IA, atau S3 One Zone-IA
region-Requests-Tier5	Hitung	Per Jam	Jumlah permintaan pemulihan Massal S3 Glacier Flexible Retrieval
region-Requests-Tier6	Hitung	Per Jam	Jumlah permintaan pemulihan S3 Glacier Fleksibel Dipercepat
region-Requests-Tier8	Hitung	Per Jam	Jumlah permintaan Hibah Akses S3
region-Requests-XZ-Tier1	Hitungan	Per Jam	Jumlah PUT atau COPY permintaan pada objek S3 Express One Zone
region-Requests-XZ-Tier2	Hitungan	Per Jam	Jumlah GET dan semua permintaan non-S3 Express One Zone-Tier 1 lainnya pada objek S3 Express One Zone
region-Requests-ZIA-Tier1	Hitungan	Per Jam	JumlahPUT,COPY, atau POST permintaan pada objek IA Zona Satu S3

Jenis Penggunaan	Unit	Granularitas	Deskripsi
region-Requests-ZIA-Tier2	Hitungan	Per Jam	Jumlah GET dan semua permintaan non-S3 One Zone-IA-Tier 1 lainnya pada objek IA Zona Satu S3
region-Retrieval-GIR	GB	Per Jam	Jumlah data yang diambil dari penyimpanan S3 Glacier Instant Retrieval.
region-Retrieval-SIA	GB	Per Jam	Jumlah data yang diambil dari penyimpanan S3 Standard-IA
region-Retrieval-XZ	GB	Per Jam	Porsi data yang melebihi 512 KB dalam permintaan pengambilan tertentu (PUTatauCOPY) dengan penyimpanan S3 Express One Zone
region-Retrieval-ZIA	GB	Per Jam	Jumlah data yang diambil dari penyimpanan S3 One Zone-IA
region-S3DSSE-In-Bytes	GB	Bulanan	Jumlah data yang dienkripsi ganda oleh Amazon S3
region-S3DSSE-Out-Bytes	GB	Bulanan	Jumlah data terenkripsi ganda yang didekripsi oleh Amazon S3

Jenis Penggunaan	Unit	Granularitas	Deskripsi
region-S3G-DataTransfer-In-Bytes	GB	Per Jam	Jumlah data yang ditransfer ke Amazon S3 untuk memulihkan objek dari penyimpanan S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive
region-S3G-DataTransfer-Out-Bytes	GB	Per Jam	Jumlah data yang ditransfer dari Amazon S3 ke objek transisi ke penyimpanan S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive
region-Select-Returned-Bytes	GB	Per Jam	Jumlah data yang dikembalikan dengan permintaan Pilihan dari penyimpanan S3 Standard
region-Select-Returned-GIR-Bytes	GB	Per Jam	Jumlah data yang dikembalikan dengan permintaan Pilihan dari penyimpanan S3 Glacier Instant Retrieval.
region-Select-Returned-INT-Bytes	GB	Per Jam	Jumlah data yang dikembalikan dengan permintaan Pilihan dari penyimpanan S3 Intelligent-Tiering

Jenis Penggunaan	Unit	Granularitas	Deskripsi
region-Select-Returned-SIA-Bytes	GB	Per Jam	Jumlah data yang dikembalikan dengan permintaan Pilihan dari penyimpanan S3 Standard-IA
region-Select-Returned-ZIA-Bytes	GB	Per Jam	Jumlah data yang dikembalikan dengan permintaan Pilihan dari penyimpanan S3 One Zone-IA
region-Select-Scanned-Bytes	GB	Per Jam	Jumlah data yang dipindai dengan permintaan Pilihan dari penyimpanan S3 Standard
region-Select-Scanned-GIR-Bytes	GB	Per Jam	Jumlah data yang dipindai dengan permintaan Pilihan dari penyimpanan S3 Glacier Instant Retrieval.
region-Select-Scanned-INT-Bytes	GB	Per Jam	Jumlah data yang dipindai dengan permintaan Pilihan dari penyimpanan S3 Intelligent-Tiering
region-Select-Scanned-SIA-Bytes	GB	Per Jam	Jumlah data yang dipindai dengan permintaan Pilihan dari penyimpanan S3 Standard-IA

Jenis Penggunaan	Unit	Granularitas	Deskripsi
region-Select-Scanned-ZIA-Bytes	GB	Per Jam	Jumlah data yang dipindai dengan permintaan Pilihan dari penyimpanan S3 One Zone-IA
region-Standard-Retrieval-Bytes	GB	Per Jam	Jumlah data yang diambil dengan permintaan standar S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive
region-StorageAnalytics-ObjCount	Objek	Per Jam	Jumlah objek unik yang dipantau di setiap konfigurasi Analisis Kelas Penyimpanan.
region-StorageLens-ObjCount	Objek	Harian	Jumlah objek unik di setiap dasbor Lensa Penyimpanan S3 yang dilacak oleh metrik dan rekomendasi lanjutan Lensa Penyimpanan S3.
region-StorageLensFreeTier-ObjCount	Objek	Harian	Jumlah objek unik di setiap dasbor Lensa Penyimpanan S3 yang dilacak oleh metrik penggunaan Lensa Penyimpanan S3.
StorageObjectCount	Hitung	Harian	Jumlah objek yang disimpan dalam bucket tertentu

Jenis Penggunaan	Unit	Granularitas	Deskripsi
region-Tables-CompactedObjects	Objek	Per Jam	Jumlah objek yang dipadatkan dalam ember tabel Amazon S3
region-Tables-MonitoredObjects	Objek	Per Jam	Jumlah objek di ember tabel Amazon S3
region-Tables-ProcessedBytes	GB	Per Jam	Jumlah data yang diproses untuk pemadatan di bucket tabel Amazon S3
region-Tables-Requests-Tier1	Hitungan	Per Jam	Jumlah permintaan PUT di ember tabel Amazon S3
region-Tables-Requests-Tier2	Hitungan	Per Jam	Jumlah GET dan semua permintaan non-Tier1 lainnya di bucket tabel Amazon S3
region-Tables-TimedStorage-ByteHrs	GB-Bulan	Harian	Jumlah GB-bulan data disimpan di bucket tabel Amazon S3
region-TagStorage-TagHrs	Tanda-Jam	Harian	Total tanda pada semua objek dalam bucket yang dilaporkan per jam
region-TimedStorage-ByteHrs	GB-Bulan	Harian	Jumlah GB-bulan bahwa data disimpan dalam penyimpanan Standar S3
region-TimedStorage-GDA-ByteHrs	GB-Bulan	Harian	Jumlah GB-bulan data disimpan di penyimpanan S3 Glacier Deep Archive

Jenis Penggunaan	Unit	Granularitas	Deskripsi
region-TimedStorage-GDA-Staging	GB-Bulan	Harian	Jumlah GB-bulan data disimpan di penyimpanan pementasan S3 Glacier Deep Archive
region-TimedStorage-GIR-ByteHrs	GB-Bulan	Harian	Jumlah GB-bulan data yang disimpan dalam penyimpanan S3 Glacier Instant Retrieval.
region-TimedStorage-GIR-SmObjects	GB-Bulan	Harian	Jumlah GB-bulan benda kecil (lebih kecil dari 128 KB) disimpan dalam penyimpanan Pengambilan Instan S3 Glacier.
region-TimedStorage-GlacierByteHrs	GB-Bulan	Harian	Jumlah GB-bulan yang data disimpan dalam penyimpanan S3 Glacier Flexible Retrieval
region-TimedStorage-GlacierStaging	GB-Bulan	Harian	Jumlah GB-bulan yang data disimpan dalam penyimpanan pementasan S3 Glacier Flexible Retrieval
region-TimedStorage-INT-FA-ByteHrs	GB-Bulan	Harian	Jumlah GB-bulan bahwa data disimpan di tingkat Akses Sering dari penyimpanan S3 Intelligent-Tiering 5

Jenis Penggunaan	Unit	Granularitas	Deskripsi
region-TimedStorage-INT-IA-ByteHrs	GB-Bulan	Harian	Jumlah GB-bulan yang data disimpan di tingkat Akses Jarang penyimpanan S3 Intelligent-Tiering
region-TimedStorage-INT-AA-ByteHrs	GB-Bulan	Harian	Jumlah GB-bulan bahwa data disimpan di tingkat Akses Arsip penyimpanan S3 Intelligent-Tiering
region-TimedStorage-INT-AIA-ByteHrs	GB-Bulan	Harian	Jumlah GB-bulan bahwa data disimpan di tingkat Akses Instan Arsip penyimpanan S3 Intelligent-Tiering
region-TimedStorage-INT-DAA-ByteHrs	GB-Bulan	Harian	Jumlah GB-bulan data disimpan di tingkat Deep Archive Access dari penyimpanan S3 Intelligent-Tiering
region-TimedStorage-RRS-ByteHrs	GB-Bulan	Harian	Jumlah GB-bulan yang data disimpan dalam penyimpanan Reduced Redundancy Storage (RRS)
region-TimedStorage-SIA-ByteHrs	GB-Bulan	Harian	Jumlah GB-bulan bahwa data disimpan dalam penyimpanan IA standar S3

Jenis Penggunaan	Unit	Granularitas	Deskripsi
region-TimedStorage-SIA-SmObjects	GB-Bulan	Harian	Jumlah GB-bulan benda kecil (lebih kecil dari 128 KB) disimpan dalam penyimpanan IA standar S3 4
region-TimedStorage-XZ-ByteHrs	GB-Bulan	Harian	Jumlah GB-bulan data yang disimpan dalam penyimpanan S3 Express One Zone
region-TimedStorage-ZIA-ByteHrs	GB-Bulan	Harian	Jumlah GB-bulan data disimpan di penyimpanan IA Zona Satu S3
region-TimedStorage-ZIA-SmObjects	GB-Bulan	Harian	Jumlah GB-bulan benda kecil (lebih kecil dari 128 KB) disimpan di penyimpanan IA Zona Satu S3
region-Upload-XZ	GB	Per Jam	Jumlah data yang melebihi 512 KB dalam permintaan unggahan tertentu (PUTatauCOPY) dengan S3 Express One Zone

Catatan

1. Jenis `Global-Bucket-Hrs` dan `Global-Bucket-Hrs-FreeTier` penggunaan berlaku untuk ember tujuan umum dalam komersial Wilayah AWS dan AWS GovCloud (US).
2. Jika Anda menghentikan transfer sebelum prosesnya selesai, jumlah data yang ditransfer dapat melebihi jumlah data yang diterima aplikasi. Perbedaan ini dapat terjadi karena permintaan penghentian transfer tidak dapat dieksekusi secara instan, dan sejumlah data mungkin dalam

perjalanan, menunggu eksekusi permintaan penghentian. Data dalam perjalanan ini ditagih sebagai data yang ditransfer “keluar.”

3. Ketika objek yang diarsipkan ke S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval, atau kelas penyimpanan S3 Glacier Deep Archive dihapus, ditimpa, atau dialihkan ke kelas penyimpanan yang berbeda sebelum komitmen penyimpanan minimum telah berlalu, yaitu 90 hari untuk S3 Glacier Instant Retrieval dan S3 Glacier Flexible Retrieval, atau 180 hari untuk S3 Glacier S3 Glacier Deep Archive, ada biaya prorata per gigabyte untuk hari-hari yang tersisa.
4. Untuk objek yang berada dalam penyimpanan S3 Standard-IA atau S3 One Zone-IA, ketika mereka dihapus, ditimpa, atau dialihkan ke kelas penyimpanan yang berbeda sebelum 30 hari, ada biaya prorata per gigabyte untuk hari-hari yang tersisa.
5. Untuk benda-benda kecil (lebih kecil dari 128 KB) yang berada di S3 Standard-IA atau S3 One Zone-IA storage, ketika mereka dihapus, ditimpa, atau dialihkan ke kelas penyimpanan yang berbeda sebelum 30 hari, ada biaya prorata per gigabyte untuk hari-hari yang tersisa.
6. Tidak ada ukuran objek minimum yang dapat ditagih untuk objek di kelas penyimpanan S3 Intelligent-Tiering. Objek yang lebih kecil dari 128 KB tidak dipantau atau memenuhi syarat untuk peningkatan otomatis. Objek yang lebih kecil selalu disimpan di tingkat Akses Sering S3 Intelligent-Tiering.
7. Saat Anda memulai`CreateMultipartUpload`,`UploadPart`, atau `UploadPartCopy` permintaan ke kelas penyimpanan S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive, permintaan akan ditagih dengan tarif permintaan Standar S3 hingga Anda menyelesaikan unggahan multipart. Setelah unggahan selesai, `CompleteMultipartUpload` permintaan tunggal ditagih dengan `PUT` tarif untuk penyimpanan S3 Glacier tujuan. Suku cadang unggahan multibagian yang sedang berlangsung untuk kelas penyimpanan Pengambilan Fleksibel Gletser S3 `PUT` ke S3 Glacier ditagih sebagai S3 Glacier Flexible Retrieval Staging Staging Storage pada tingkat penyimpanan Standar S3 hingga unggahan selesai. Demikian pula, bagian unggahan multibagian yang sedang berlangsung untuk kelas penyimpanan S3 Glacier Deep Archive `PUT` ke S3 Glacier Deep Archive ditagih sebagai S3 Glacier Deep Archive Staging Storage dengan kecepatan penyimpanan Standar S3 hingga unggahan selesai.
8. S3 Express One Zone menerapkan biaya tetap per permintaan untuk ukuran permintaan hingga 512 KB. Biaya tambahan per GB diterapkan untuk `PUT` permintaan dan `GET` permintaan untuk porsi permintaan lebih besar dari 512 KB.
9. Untuk informasi tentang fitur yang didukung untuk kelas penyimpanan S3 Express One Zone, lihat Fitur Amazon S3 tidak didukung oleh bucket direktori .
10. Jenis penggunaan dengan unit yang ditagih dalam GB dihitung dalam byte dalam laporan penggunaan.

11. GB-Bulan diperoleh dengan mengambil jumlah total GB-jam, menggabungkannya selama sebulan, dan kemudian membaginya dengan jumlah jam di bulan itu. Untuk mempelajari lebih lanjut lihat, Pertanyaan yang Sering Diajukan: Bagaimana saya akan ditagih dan ditagih untuk penggunaan Amazon S3 saya?

> **Note**
>
> Secara umum, pemilik bucket S3 ditagih untuk permintaan dengan respons HTTP `200 OK` yang berhasil dan respons kesalahan `4XX` klien HTTP. Pemilik bucket tidak ditagih untuk respons kesalahan `5XX` server HTTP, seperti `503 Slow Down` kesalahan HTTP. Untuk informasi selengkapnya tentang kode kesalahan S3 di bawah HTTP `3XX` dan kode `4XX` status yang tidak ditagih, lihat. Penagihan untuk respons kesalahan Amazon S3 Untuk informasi selengkapnya tentang biaya penagihan jika bucket Anda dikonfigurasi sebagai bucket Requester Pays, lihat. Cara kerja Pembayaran Pemohon

Melacak Operasi dalam Laporan Penggunaan Anda

Operasi menggambarkan tindakan yang diambil pada AWS objek atau bucket Anda berdasarkan jenis penggunaan yang ditentukan. Operasi ditunjukkan dengan kode yang cukup jelas, seperti `PutObject` atau `ListBucket`. Untuk melihat tindakan pada bucket Anda yang menghasilkan jenis penggunaan tertentu, gunakan kode ini. Saat membuat laporan penggunaan, Anda dapat memilih untuk menyertakan Semua Operasi, atau operasi tertentu, misalnya, `GetObject`, untuk dilaporkan.

Info selengkapnya

- AWS laporan penggunaan untuk Amazon S3
- AWS Billing laporan untuk Amazon S3
- Harga Amazon S3
- Amazon S3 FAQs

Penagihan untuk respons kesalahan Amazon S3

Secara umum, pemilik bucket S3 ditagih untuk permintaan dengan respons HTTP `200 OK` yang berhasil dan respons kesalahan `4XX` klien HTTP. Pemilik bucket tidak ditagih untuk respons kesalahan `5XX` server HTTP, seperti `503 Slow Down` kesalahan HTTP. Untuk informasi

selengkapnya tentang biaya penagihan jika bucket Anda dikonfigurasi sebagai bucket Requester Pays, lihat. Cara kerja Pembayaran Pemohon

Tabel berikut mencantumkan kode kesalahan tertentu di bawah HTTP 3XX dan kode 4XX status yang tidak ditagih. Untuk bucket yang dikonfigurasi dengan hosting situs web, permintaan yang berlaku dan biaya lainnya akan tetap berlaku saat S3 mengembalikan dokumen kesalahan khusus atau untuk pengalihan khusus.

> **Note**
>
> Untuk `AccessDenied` (HTTP`403 Forbidden`), S3 tidak membebankan biaya kepada pemilik bucket saat permintaan dimulai di luar AWS akun individu pemilik bucket atau organisasi pemilik bucket. AWS

Kode status HTTP	Kode kesalahan	Deskripsi kode kesalahan
301 Dipindahkan Secara Permanen	PermanentRedirect	Bucket yang Anda coba akses harus ditangani menggunakan titik akhir yang ditentukan. Kirim semua permintaan future ke endpoint ini.
	PermanentRedirectControlError	Operasi API yang Anda coba akses harus ditangani menggunakan titik akhir yang ditentukan. Kirim semua permintaan future ke endpoint ini.
307 Pengalihan Sementara	TemporaryRedirect	Anda sedang dialihkan ke bucket saat server Domain Name System (DNS) sedang diperbarui.

Kode status HTTP	Kode kesalahan	Deskripsi kode kesalahan
400 Permintaan Buruk	AuthorizationHeaderMalformed	Header otorisasi yang Anda berikan tidak valid.
	AuthorizationQueryParametersError	Parameter kueri otorisasi yang Anda berikan tidak valid.
	ConnectionClosedByRequester	Kembali ke penelepon asli ketika terjadi kesalahan saat membaca WriteGetObjectResponse tubuh.
	DeviceNotActiveError	Perangkat saat ini tidak aktif.
	EndpointNotFound	Permintaan langsung ke titik akhir yang benar.
	ExpiredToken	Token yang disediakan telah kedaluwarsa.
	IllegalLocationConstraintException	Anda mencoba mengakses bucket dari Wilayah yang berbeda dari tempat bucket berada. Untuk menghindari kesalahan ini, gunakan `--region` opsi. Sebagai contoh: `aws s3 cp` *`awsexample.txt`* `s3://` *`amzn-s3-demo-bucket`* `/ --region` *`ap-east-1`* `.`

Kode status HTTP	Kode kesalahan	Deskripsi kode kesalahan	
	InvalidArgument	Kesalahan ini dapat terjadi karena alasan berikut: • Argumen yang ditentukan tidak valid. • Permintaan itu tidak memiliki header yang diperlukan. • Argumen yang ditentukan tidak lengkap atau dalam format yang salah. • Argumen yang ditentukan harus memiliki panjang lebih besar dari atau sama dengan 3.	
	InvalidBucketOwnerAWSAccountID	Nilai parameter pemilik bucket yang diharapkan harus berupa Akun AWS ID.	
	InvalidDigest	Nilai Content- MD5 atau checksum yang Anda tentukan tidak valid.	

Kode status HTTP	Kode kesalahan	Deskripsi kode kesalahan	
	InvalidEncryptionAlgorithmError	Permintaan enkripsi yang Anda tentukan tidak valid. Nilai yang valid adalah`AES256`.	
	InvalidHostHeader	Header host yang disediakan dalam permintaan menggunakan gaya pengalamatan yang salah.	
	InvalidHttpMethod	Permintaan dibuat menggunakan metode HTTP yang tidak terduga.	

Kode status HTTP	Kode kesalahan	Deskripsi kode kesalahan	
	InvalidRequest	Kesalahan ini dapat terjadi karena alasan berikut: • Permintaan menggunakan versi tanda tangan yang salah. Gunakan `AWS4-HMAC-SHA256` (Versi Tanda Tangan 4). • Titik akses hanya dapat dibuat untuk bucket yang ada. • Titik akses tidak dalam keadaan di mana ia dapat dihapus. • Titik akses hanya dapat dicantumkan untuk bucket yang ada. • Token berikutnya tidak valid. • Setidaknya satu tindakan harus ditentukan dalam aturan siklus hidup. •	

Kode status HTTP	Kode kesalahan	Deskripsi kode kesalahan	
		Setidaknya satu aturan siklus hidup harus ditentukan. • Jumlah aturan siklus hidup tidak boleh melebihi batas yang diizinkan dari 1000 aturan. • Rentang untuk `MaxResults` parameter tidak valid. • Permintaan SOAP harus dibuat melalui koneksi HTTPS. • Amazon S3 Transfer Acceleration tidak didukung untuk bucket dengan nama yang tidak sesuai dengan DNS. • Amazon S3 Transfer Acceleration tidak didukung untuk bucket dengan periode (.) dalam nama mereka. • Endpoint Amazon S3 Transfer Acceleration	

Kode status HTTP	Kode kesalahan	Deskripsi kode kesalahan	
		hanya mendukung permintaan gaya virtual. • Amazon S3 Transfer Acceleration tidak dikonfigurasi pada bucket ini. • Amazon S3 Transfer Acceleration dinonaktifkan di bucket ini. • Amazon S3 Transfer Acceleration tidak didukung pada bucket ini. Untuk bantuan, hubungi Dukungan. • Amazon S3 Transfer Acceleration tidak dapat diaktifkan di bucket ini. Untuk bantuan, hubungi Dukungan. • Nilai yang bertentangan disediakan dalam header HTTP dan parameter kueri. • Nilai yang bertentangan disediakan di header	

Kode status HTTP	Kode kesalahan	Deskripsi kode kesalahan	
		HTTP dan bidang formulir POST. • CopyObject permintaan dibuat pada objek yang berukuran lebih besar dari 5GB.	
	InvalidSessionException	Dikembalikan jika sesi tidak ada lagi karena waktunya habis atau kedaluwarsa.	
	InvalidSignature	Tanda tangan permintaan yang dihitung server tidak sesuai dengan tanda tangan yang Anda berikan. Periksa kunci akses AWS rahasia Anda dan metode penandatanganan. Untuk informasi selengkapnya, lihat Menandatangani dan mengautentikasi permintaan REST.	
	Tidak valid SOAPRequest	Badan permintaan SOAP tidak valid.	
	InvalidStorageClass	Kelas penyimpanan yang Anda tentukan tidak valid.	

Kode status HTTP	Kode kesalahan	Deskripsi kode kesalahan	
	InvalidTag	Permintaan Anda berisi input tag yang tidak valid. Misalnya, permintaan Anda mungkin berisi kunci duplikat, kunci, atau nilai yang terlalu panjang, atau tag sistem.	
	InvalidToken	Token yang disediakan cacat atau tidak valid.	
	InvaliDuri	URI yang ditentukan tidak dapat diuraikan.	
	KeyTooLongError	Kunci Anda terlalu panjang.	
	KM. DisabledException	Permintaan ditolak karena kunci KMS yang ditentukan tidak diaktifkan.	

Kode status HTTP	Kode kesalahan	Deskripsi kode kesalahan	
	KM. InvalidKeyUsageException	Permintaan ditolak karena salah satu alasan berikut: • KeyUsage Nilai kunci KMS tidak kompatibel dengan operasi API. • Algoritma enkripsi atau algoritma penandatanganan yang ditentukan untuk operasi tidak kompatibel dengan jenis bahan kunci dalam kunci KMS ()KeySpec. Untuk mengenkripsi, mendekripsi, mengenkripsi ulang, dan menghasilkan kunci data, harus ENCRYPT_DECRYPT. KeyUsage Untuk menandatangani dan memverifikasi pesan, KeyUsage harus SIGN_VERIFY. Untuk membuat dan memverifikasi kode otentikasi pesan (MACs), KeyUsage harus GENERATE_VERIFY_MAC. Untuk mendapatkan rahasia	

Kode status HTTP	Kode kesalahan	Deskripsi kode kesalahan	
		perjanjian kunci, KeyUsage harus KEY_AGREEMENT. Untuk menemukan kunci KMS, gunakan DescribeKey operasi. KeyUsage Untuk menemukan enkripsi atau algoritma penandatanganan yang didukung untuk kunci KMS tertentu, gunakan operasi. DescribeKey	

Kode status HTTP	Kode kesalahan	Deskripsi kode kesalahan	
	KM. KMSInvalidStateException	Permintaan ditolak karena status sumber daya yang ditentukan tidak valid untuk permintaan ini. Pengecualian ini berarti salah satu dari yang berikut: • Status kunci kunci KMS tidak kompatibel dengan operasi. Untuk menemukan status kunci, gunakan DescribeKey operasi. Untuk informasi selengkapnya tentang status kunci mana yang kompatibel dengan setiap operasi KMS, lihat Status kunci AWS KMS keys dalam Panduan AWS Key Management Service Pengembang. • Untuk operasi kriptografi pada kunci KMS di toko kunci khusus, pengecualian ini merupakan kegagalan umum dengan banyak kemungkinan penyebab. Untuk	

Kode status HTTP	Kode kesalahan	Deskripsi kode kesalahan	
		mengidentifikasi penyebabnya, lihat pesan kesalahan yang menyertai pengecualian.	
	KM. NotFoundException	Permintaan ditolak karena entitas atau sumber daya tertentu tidak dapat ditemukan.	
	LambdaInvalidResponse	Kembali ke penelepon asli saat WriteGetObjectResponse merespons dengan ValidationError . AWS Lambda Lihat ValidationError pesan untuk lebih jelasnya. Tidak semua kasus ValidationError menghasilkan LambdaInvalidResponse kesalahan.	

Kode status HTTP	Kode kesalahan	Deskripsi kode kesalahan	
	LambdaInvocationFailed	Pemanggilan fungsi Lambda gagal. Penelepon mungkin menerima kesalahan berikut ketika S3 Object Lambda tidak berhasil menjalankan fungsi Lambda yang dikonfigu rasi. Pesan kesalahan mungkin berisi rincian tentang kesalahan akhirnya yang dikembali kan oleh AWS Lambda layanan saat menjalank an fungsi (misalnya, kode status, kode kesalahan, pesan kesalahan, dan ID permintaan).	
	Cacat ACLError	ACL yang Anda berikan tidak terbentuk dengan baik atau tidak memvalida si skema kami yang diterbitkan.	
	Cacat POSTRequest	Isi permintaan POST Anda tidak terbentuk dengan baik multipart/ form-data.	

Kode status HTTP	Kode kesalahan	Deskripsi kode kesalahan	
	MalformedXML	XHTML yang Anda berikan tidak terbentuk dengan baik atau tidak memvalidasi skema kami yang dipublikasikan.	
	MaxPostPreDataLengthExceededError	Bidang permintaan POST Anda sebelum file upload terlalu besar.	
	MetadataTooLarge	Header metadata Anda melebihi ukuran metadata maksimum yang diizinkan.	
	MissingAttachment	Lampiran SOAP diharapkan, tetapi tidak ada yang ditemukan.	
	MissingRequestBodyError	Anda mengirim dokumen XHTML kosong sebagai permintaan.	
	MissingSecurityHeader	Permintaan Anda tidak memiliki header yang diperlukan.	
	NoLoggingStatusForKey	Tidak ada yang namanya subresource status logging untuk sebuah kunci.	

Kode status HTTP	Kode kesalahan	Deskripsi kode kesalahan	
	NotDeviceOwnerError	Perangkat yang menghasilkan token tidak dimiliki oleh pengguna yang diautentikasi.	
	ResponseInterrupted	Kembali ke penelepon asli ketika terjadi kesalahan saat membaca WriteGetObjectResponse tubuh.	
	RequestHeaderSectionTooLarge	Header permintaan dan parameter kueri yang digunakan untuk membuat permintaan melebihi ukuran maksimum yang diizinkan	
	TokenCodeInvalidError	Nomor seri dan/atau kode token yang Anda berikan tidak valid.	
	UnexpectedContent	Permintaan ini berisi konten yang tidak didukung.	
	UnsupportedArgument	Permintaan berisi argumen yang tidak didukung.	

Kode status HTTP	Kode kesalahan	Deskripsi kode kesalahan	
	UnsupportedSignature	Permintaan yang diberikan ditandatangani dengan versi STS Token yang tidak didukung atau versi tanda tangan tidak didukung.	
	UserKeyMustBeSpecified	Permintaan bucket POST harus berisi nama bidang yang ditentukan. Jika ditentukan, periksa urutan bidang.	
	IncorrectEndpoint	Bucket yang ditentuka n ada di Wilayah lain. Permintaan langsung ke titik akhir yang benar.	
	ValidationError	Kesalahan validasi dapat dikembalikan dari operasi WriteGetObjectResponse API dan dapat terjadi karena berbagai alasan. Lihat pesan kesalahan untuk detail selengkap nya.	
403 Dilarang	RequestTimeTooSkewed	Perbedaan antara waktu permintaan dan waktu server terlalu besar.	

Kode status HTTP	Kode kesalahan	Deskripsi kode kesalahan	
	SignatureDoesNotMatch	Tanda tangan permintaan yang dihitung server tidak sesuai dengan tanda tangan yang Anda berikan. Periksa kunci akses AWS rahasia Anda dan metode penandatanganan. Untuk informasi selengkapnya, lihat Otentikasi REST dan Otentikasi SOAP.	
	NotSignedUp	Akun Anda tidak mendaftar untuk layanan Amazon S3. Anda harus mendaftar sebelum dapat menggunakan Amazon S3. Anda dapat mendaftar di URL berikut: https://aws.amazon.com/s3	
	InvalidSecurity	Kredensi keamanan yang diberikan tidak valid.	
	InvalidPayer	Semua akses ke objek ini telah dinonaktifkan. Untuk bantuan lebih lanjut, lihat Hubungi Kami.	
	InvalidAccessKeyId	ID kunci AWS akses yang Anda berikan tidak ada dalam catatan kami.	

Kode status HTTP	Kode kesalahan	Deskripsi kode kesalahan	
	AccountProblem	Ada masalah dengan Anda Akun AWS yang mencegah operasi selesai dengan sukses. Untuk bantuan lebih lanjut, lihat Hubungi Kami.	
	UnauthorizedAccessError	Hanya berlaku di Wilayah Tiongkok. Dikembali kan saat permintaan dibuat ke bucket yang tidak memiliki lisensi ICP. Untuk informasi lebih lanjut, lihat ICP Recordal.	
	Tak terduga IPError	Hanya berlaku di Wilayah Tiongkok. Permintaan ini ditolak karena IP tidak terduga.	
	MissingAuthenticat ionToken	Permintaan itu tidak ditandatangani.	

Kode status HTTP	Kode kesalahan	Deskripsi kode kesalahan	
	LambdaPermissionError	Penelepon tidak berwenang untuk menjalankan fungsi Lambda. Penelepon harus memiliki izin untuk menjalankan fungsi Lambda. Periksa kebijakan yang dilampirkan ke pemanggil dan pastikan kebijakan tersebut telah diizinkan untuk digunakan `lambda:Invoke` untuk fungsi yang dikonfigurasi. Pesan kesalahan mungkin berisi detail tentang kesalahan yang akhirnya dikembalikan oleh layanan Lambda saat menjalankan fungsi (misalnya, kode status, kode kesalahan, pesan kesalahan, dan ID permintaan).	

Kode status HTTP	Kode kesalahan	Deskripsi kode kesalahan	
404 Tidak Ditemukan	LambdaNotFound	AWS Lambda Fungsinya tidak ditemukan. Fungsi, versi, atau alias Lambda yang dikonfigurasi tidak ditemukan saat mencoba memanggil nya. Pastikan konfigura si Titik Akses Lambda Objek S3 menunjuk ke ARN fungsi Lambda yang benar. Pesan kesalahan mungkin berisi rincian tentang kesalahan akhirnya yang dikembali kan oleh AWS Lambda layanan saat menjalank an fungsi (misalnya, kode status, kode kesalahan, pesan kesalahan, dan ID permintaan).	
	NoSuchAsyncRequest	Permintaan yang ditentukan tidak ditemukan.	
	NoSuchObjectLockCo nfiguration	Objek yang ditentukan tidak memiliki ObjectLock konfigurasi.	

Kode status HTTP	Kode kesalahan	Deskripsi kode kesalahan	
	NoSuchUpload	Unggahan multipart yang ditentukan tidak ada. ID unggahan mungkin tidak valid, atau unggahan multibagian mungkin telah dibatalkan atau diselesaikan.	
	NoSuchWebsiteConfiguration	Bucket yang ditentukan tidak memiliki konfigurasi situs web.	
	NoTransformationDefined	Tidak ada transformasi yang ditemukan untuk Object Lambda Access Point ini.	
	ObjectLockConfigurationNotFoundError	Konfigurasi Object Lock tidak ada untuk bucket ini.	
405 Metode Tidak Diizinkan	MethodNotAllowed	Metode yang ditentukan tidak diperbolehkan terhadap sumber daya ini.	
409 Konflik	BucketAlreadyExists	Nama bucket yang diminta tidak tersedia. Namespace bucket dibagikan oleh semua pengguna sistem. Tentukan nama yang berbeda dan coba lagi.	

Kode status HTTP	Kode kesalahan	Deskripsi kode kesalahan	
	InvalidBucketState	Permintaan tidak valid untuk status bucket saat ini.	
	OperationAborted	Operasi kondisional yang bertentangan saat ini sedang berlangsung terhadap sumber daya ini. Coba lagi.	
411 Panjang Diperlukan	MissingContentLength	Anda harus menyediakan header HTTP Content-Length.	
412 Prasyarat Gagal	RequestIsNotMultiPartContent	Permintaan bucket POST harus dari multipart/form-data tipe enclosure.	
416 Rentang yang Diminta Tidak Memuaskan	InvalidRange	Rentang yang diminta tidak berlaku untuk permintaan. Coba rentang lain.	

Memahami dan mengelola kelas penyimpanan Amazon S3

Setiap objek di Amazon S3 memiliki kelas penyimpanan yang terkait dengan objeknya. Secara default, objek di S3 disimpan di kelas penyimpanan Standar S3, namun Amazon S3 menawarkan berbagai kelas penyimpanan lain untuk objek yang Anda simpan. Pilih kelas yang sesuai dengan skenario kasus penggunaan dan kebutuhan akses kinerja. Memilih kelas penyimpanan yang dirancang untuk kasus penggunaan memungkinkan Anda mengoptimalkan biaya penyimpanan,

kinerja, dan ketersediaan untuk objek Anda. Semua kelas penyimpanan ini menawarkan daya tahan tinggi.

Bagian berikut memberikan rincian dari berbagai kelas penyimpanan dan cara mengatur kelas penyimpanan untuk objek.

Topik

- Kelas penyimpanan untuk objek yang sering diakses
- Kelas penyimpanan untuk mengoptimalkan data secara otomatis dengan pola akses yang berubah atau tidak diketahui
- Kelas penyimpanan untuk objek yang jarang diakses
- Kelas penyimpanan untuk objek yang jarang diakses
- Kelas penyimpanan untuk Amazon S3 pada Outposts
- Membandingkan kelas penyimpanan Amazon S3
- Mengatur kelas penyimpanan objek
- Analitik Amazon S3–Analisis Kelas Penyimpanan
- Mengelola biaya penyimpanan dengan Amazon S3 Intelligent-Tiering
- Memahami kelas penyimpanan S3 Glacier untuk penyimpanan data jangka panjang
- Bekerja dengan objek yang diarsipkan

Kelas penyimpanan untuk objek yang sering diakses

Untuk kasus penggunaan yang sensitif terhadap kinerja (kasus yang memerlukan waktu akses milidetik) dan data yang sering diakses, Amazon S3 menyediakan kelas penyimpanan berikut:

- Standar S3 (`STANDARD`) - Kelas penyimpanan default. Jika kelas penyimpanan tidak ditentukan saat mengunggah objek, Amazon S3 akan menetapkan kelas penyimpanan S3 Standard. Untuk membantu Anda mengoptimalkan biaya antara Standar S3 dan Standar S3 - IA dapat Anda gunakan. Analitik Amazon S3–Analisis Kelas Penyimpanan
- S3 Express One Zone (`EXPRESS_ONEZONE`) - Amazon S3 Express One Zone adalah kelas penyimpanan Amazon S3 zona tunggal berkinerja tinggi yang dibuat khusus untuk menghadirkan akses data milidetik satu digit yang konsisten untuk aplikasi Anda yang paling sensitif terhadap latensi. S3 Express One Zone adalah kelas penyimpanan objek cloud latensi terendah yang tersedia saat ini, dengan kecepatan akses data hingga 10x lebih cepat dan dengan biaya permintaan 50 persen lebih rendah dari Standar S3. Dengan S3 Express One Zone, data Anda

disimpan secara berlebihan di beberapa perangkat dalam satu Zona Ketersediaan. Untuk informasi selengkapnya, lihat S3 Express One Zone.

- Reduced Redundancy Storage (`REDUCED_REDUNDANCY`) — Kelas Reduced Redundancy Storage (RRS) dirancang untuk data yang tidak kritis dan dapat direproduksi yang dapat disimpan dengan redundansi yang lebih sedikit daripada kelas penyimpanan Standar S3.

  > **Important**
  >
  > Sebaiknya hindari penggunaan kelas penyimpanan ini. Kelas penyimpanan S3 Standard lebih hemat biaya.

  Untuk ketahanan, objek RRS memiliki perkiraan kehilangan tahunan rata-rata sebesar 0,01 persen dari objek. Jika objek RRS hilang, ketika permintaan dibuat ke objek tersebut, Amazon S3 akan menampilkan kesalahan 405.

Kelas penyimpanan untuk mengoptimalkan data secara otomatis dengan pola akses yang berubah atau tidak diketahui

S3 Intelligent-Tiering () `INTELLIGENT_TIERING` adalah kelas penyimpanan Amazon S3 yang dirancang untuk mengoptimalkan biaya penyimpanan dengan memindahkan data secara otomatis ke tingkat akses yang paling hemat biaya, tanpa dampak kinerja atau overhead operasional. S3 Intelligent-Tiering adalah satu-satunya kelas penyimpanan cloud yang memberikan penghematan biaya otomatis dengan memindahkan data pada tingkat objek granular antar tingkat akses ketika pola akses berubah. S3 Intelligent-Tiering adalah kelas penyimpanan yang ideal ketika Anda ingin mengoptimalkan biaya penyimpanan untuk data yang pola aksesnya tidak diketahui atau berubah. Tidak ada biaya pengambilan untuk S3 Intelligent-Tiering.

Dengan sedikit biaya pemantauan objek dan otomatisasi bulanan, S3 Intelligent-Tiering memantau pola akses dan secara otomatis memindahkan objek yang belum diakses ke tingkat akses yang biayanya lebih rendah. S3 Intelligent-Tiering memberikan penghematan biaya penyimpanan otomatis dalam tiga tingkat akses latensi rendah dan throughput tinggi. Untuk data yang dapat diakses secara asinkron, Anda dapat memilih untuk mengaktifkan kemampuan pengarsipan otomatis di dalam kelas penyimpanan S3 Intelligent-Tiering. S3 Intelligent-Tiering dirancang untuk ketersediaan 99,9% dan daya tahan 99,99999%.

S3 Intelligent-Tiering secara otomatis menyimpan objek ke dalam tiga tingkatan akses:

- Frequent Access–Objek yang diunggah atau ditransisikan ke S3 Intelligent-Tiering secara otomatis akan disimpan di tingkat Frequent Access.
- Infrequent Access–S3 Intelligent-Tiering memindahkan objek yang belum diakses selama 30 hari berturut-turut ke tingkat Infrequent Access.
- Archive Instant Access–Dengan S3 Intelligent-Tiering, objek yang ada yang belum diakses selama 90 hari berturut-turut secara otomatis akan dipindahkan ke tingkat Archive Instant Access.

Selain tiga tingkatan ini, S3 Intelligent-Tiering menawarkan dua tingkatan akses arsip opsional:

- Archive Access–S3 Intelligent-Tiering memberikan opsi untuk mengaktifkan tingkat Archive Access untuk data yang dapat diakses secara asinkron. Setelah aktivasi, tingkat Archive Access secara otomatis akan mengarsipkan objek yang belum diakses selama minimal 90 hari berturut-turut.
- Deep Archive Access–S3 Intelligent-Tiering memberikan opsi untuk mengaktifkan tingkat Deep Archive Access untuk data yang dapat diakses secara asinkron. Setelah aktivasi, tingkat Deep Archive Access secara otomatis akan mengarsipkan objek yang belum diakses selama minimal 180 hari berturut-turut.

> **Note**
>
> - Hanya aktifkan tingkat Archive Access selama 90 hari jika Anda ingin melewati tingkat Archive Instant Access. Tingkat Akses Arsip memberikan penyimpanan berbiaya sedikit lebih rendah dengan minute-to-hour waktu pengambilan. Tingkat Archive Instant Access memberikan akses milidetik dan kinerja dengan throughput tinggi.
> - Aktifkan tingkatan Archive Access dan Deep Archive Access hanya jika objek dapat diakses secara asinkron oleh aplikasi Anda. Jika objek yang Anda ambil disimpan di tingkat Archive Access atau Deep Archive Access, pulihkan objek terlebih dahulu dengan menggunakan `RestoreObject`.

Anda dapat memindahkan data yang baru dibuat ke S3 Intelligent-Tiering, mengaturnya sebagai kelas penyimpanan default. Anda juga dapat memilih untuk mengaktifkan salah satu atau kedua tingkatan akses arsip dengan menggunakan operasi `PutBucketIntelligentTieringConfiguration`API, konsol AWS CLI, atau Amazon S3. Untuk informasi selengkapnya tentang penggunaan S3 Intelligent-Tiering dan pengaktifan tingkatan akses arsip, lihat Menggunakan S3 Intelligent-Tiering.

Untuk mengakses objek di jenjang Archive Access atau Deep Archive Access, Anda harus mengembalikannya terlebih dahulu. Untuk informasi selengkapnya, lihat Memulihkan objek dari tingkat S3 Intelligent-Tiering Archive Access dan Deep Archive Access.

> **Note**
>
> Jika ukuran suatu objek kurang dari 128 KB, objek tersebut tidak dipantau dan tidak memenuhi syarat untuk penetapan tingkat otomatis. Objek yang lebih kecil akan selalu disimpan di tingkat Frequent Access. Untuk informasi selengkapnya tentang S3 Intelligent-Tiering, lihat Tingkat akses S3 Intelligent-Tiering.

Kelas penyimpanan untuk objek yang jarang diakses

Kelas penyimpanan S3 Standard-IA dan S3 One Zone-IA dirancang untuk data yang berumur panjang dan jarang diakses. (IA singkatan dari infrequent access.) Objek S3 Standard-IA dan S3 One Zone-IA tersedia untuk akses milidetik (mirip dengan kelas penyimpanan S3 Standard). Amazon S3 membebankan biaya pengambilan untuk objek ini, sehingga objek tersebut paling cocok untuk data yang jarang diakses. Untuk informasi harga, lihat harga Amazon S3.

Misalnya, Anda bisa memilih kelas penyimpanan S3 Standard-IA dan S3 One Zone-IA untuk melakukan hal berikut:

- Untuk menyimpan cadangan.
- Untuk data lama yang tidak sering diakses, tetapi masih membutuhkan akses milidetik. Misalnya, saat mengunggah data, Anda dapat memilih kelas penyimpanan S3 Standard, dan menggunakan konfigurasi siklus hidup untuk memberi tahu Amazon S3 guna mentransisi objek ke kelas S3 Standard-IA atau S3 One Zone-IA.

  Untuk informasi lebih lanjut tentang manajemen siklus hidup, lihat Mengelola siklus hidup objek.

> **Note**
>
> Kelas penyimpanan S3 Standard-IA dan S3 One Zone-IA cocok untuk objek yang lebih besar dari 128 KB untuk disimpan setidaknya selama 30 hari. Jika suatu objek berukuran kurang dari 128 KB, Amazon S3 akan mengenakan biaya sebesar 128 KB. Jika menghapus objek sebelum akhir periode durasi penyimpanan minimum 30 hari, Anda akan dikenakan biaya selama 30 hari. Objek yang dihapus, menimpa, atau dialihkan ke kelas penyimpanan lain

sebelum 30 hari akan dikenakan biaya penggunaan penyimpanan normal ditambah biaya prorata untuk sisa minimum 30 hari. Untuk informasi harga, lihat harga Amazon S3.

Kelas penyimpanan ini memiliki perbedaan sebagai berikut:

- S3 Standard-IA () - `STANDARD_IA` Amazon S3 menyimpan data objek secara berlebihan di beberapa Availability Zone yang terpisah secara geografis (mirip dengan kelas penyimpanan Standar S3). Objek S3 Standard-IA tahan terhadap hilangnya Zona Ketersediaan. Kelas penyimpanan ini menawarkan ketersediaan dan ketahanan yang lebih besar dibandingkan kelas S3 One Zone-IA. Untuk membantu Anda mengoptimalkan biaya antara Standar S3 dan Standar S3 - IA dapat Anda gunakan Analitik Amazon S3–Analisis Kelas Penyimpanan
- S3 One Zone-IA (`ONEZONE_IA`) - Amazon S3 menyimpan data objek hanya dalam satu Availability Zone, yang membuatnya lebih murah daripada IA Standar S3. Namun, data tersebut tidak tahan terhadap kehilangan fisik Zona Ketersediaan yang diakibatkan oleh bencana, seperti gempa bumi dan banjir. Kelas penyimpanan S3 One Zone-IA sama tahan lamanya dengan S3 Standard-IA, tapi ketersediaannya lebih sedikit dan ketahanannya kurang. Untuk perbandingan ketahanan dan ketersediaan kelas penyimpanan, lihat Membandingkan kelas penyimpanan Amazon S3 di akhir bagian ini. Untuk informasi harga, lihat harga Amazon S3. Untuk kasus penggunaan residensi dan isolasi data, Anda dapat membuat bucket direktori di AWS Local Zones dan menggunakan kelas penyimpanan S3 Express One Zone (`EXPRESS_ONEZONE`) dan S3 One Zone-IA (). `ONEZONE_IA` Untuk informasi selengkapnya tentang bucket direktori di Local Zones, lihatBeban kerja residensi data.

Sebaiknya lakukan hal berikut:

- S3 Standard-IA (`STANDARD_IA`) - Gunakan untuk salinan utama atau satu-satunya data yang tidak dapat dibuat ulang.
- S3 One Zone-IA (`ONEZONE_IA`) - Gunakan jika Anda dapat membuat ulang data jika Availability Zone gagal, untuk replika objek saat mengonfigurasi S3 Cross-Region Replication (CRR). Selain itu, untuk residensi dan isolasi data, Anda dapat membuat bucket direktori di AWS Local Zones dan menggunakan kelas penyimpanan IA Satu Zona S3.

Kelas penyimpanan untuk objek yang jarang diakses

Kelas penyimpanan S3 Glacier Instant Retrieval (GLACIER_IR), S3 Glacier Flexible Retrieval (), dan S3 GLACIER Glacier Deep Archive (DEEP_ARCHIVE) dirancang untuk penyimpanan data jangka panjang dan pengarsipan data berbiaya rendah. Kelas penyimpanan ini memerlukan durasi penyimpanan minimum dan biaya pengambilan sehingga paling efektif untuk data yang jarang diakses. Untuk informasi lebih lanjut tentang kelas penyimpanan S3 Glacier, lihat. Memahami kelas penyimpanan S3 Glacier untuk penyimpanan data jangka panjang

Amazon S3 menyediakan kelas penyimpanan S3 Glacier berikut:

- S3 Glacier Instant Retrieval (GLACIER_IR) — Gunakan untuk data jangka panjang yang jarang diakses dan membutuhkan pengambilan milidetik. Data dalam kelas penyimpanan ini tersedia untuk akses real-time.
- S3 Glacier Flexible Retrieval (GLACIER) — Gunakan untuk arsip di mana sebagian data mungkin perlu diambil dalam hitungan menit. Data dalam kelas penyimpanan ini diarsipkan, dan tidak tersedia untuk akses real-time.
- S3 Glacier Deep Archive DEEP_ARCHIVE () — Digunakan untuk pengarsipan data yang jarang perlu diakses. Data dalam kelas penyimpanan ini diarsipkan, dan tidak tersedia untuk akses real-time.

Mengambil objek yang diarsipkan

Anda dapat mengatur kelas penyimpanan objek ke S3 Glacier Flexible Retrieval (GLACIER) atau S3 DEEP_ARCHIVE Glacier Deep Archive () dengan cara yang sama seperti yang Anda lakukan untuk kelas penyimpanan lainnya seperti yang dijelaskan di bagian. Mengatur kelas penyimpanan objek Namun, objek S3 Glacier Flexible Retrieval dan S3 Glacier Deep Archive diarsipkan, dan tidak tersedia untuk akses real-time. Untuk informasi selengkapnya, lihat Memahami penyimpanan arsip di S3 Glacier Flexible Retrieval dan S3 Glacier Deep Archive.

> **Note**
>
> Saat Anda menggunakan kelas penyimpanan S3 Glacier, objek Anda tetap berada di Amazon S3. Anda tidak dapat mengaksesnya secara langsung melalui layanan Amazon S3 Glacier yang terpisah. Untuk informasi tentang layanan Amazon S3 Glacier, lihat Panduan Pengembang Amazon S3 Glacier.

Kelas penyimpanan untuk Amazon S3 pada Outposts

Dengan Amazon S3 di Outposts, Anda dapat membuat bucket S3 di AWS Outposts sumber daya serta menyimpan serta mengambil objek lokal untuk aplikasi yang memerlukan akses data lokal, pemrosesan data lokal, dan residensi data. Anda dapat menggunakan operasi dan fitur API yang sama AWS Outposts seperti yang Anda lakukan di Amazon S3, termasuk kebijakan akses, enkripsi, dan penandaan. Anda dapat menggunakan S3 di Outposts melalui AWS Management Console AWS CLI,, AWS SDKs, atau REST API.

S3 di Outposts menyediakan kelas penyimpanan baru, S3 Outposts (`OUTPOSTS`). Kelas penyimpanan S3 Outposts hanya tersedia untuk objek yang disimpan di bucket yang ada di Outposts. Jika Anda mencoba menggunakan kelas penyimpanan ini dengan bucket S3 di Wilayah AWS, `InvalidStorageClass` kesalahan terjadi. Selain itu, jika Anda mencoba menggunakan kelas penyimpanan S3 lain dengan objek yang disimpan di S3 pada bucket Outposts, kesalahan yang sama akan terjadi.

Objek yang disimpan di kelas penyimpanan S3 Outposts (`OUTPOSTS`) selalu dienkripsi menggunakan enkripsi di sisi server dengan kunci enkripsi terkelola Amazon S3 (SSE-S3). Untuk informasi selengkapnya, lihat Menggunakan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3).

Anda juga dapat secara eksplisit memilih untuk mengenkripsi objek yang disimpan di kelas penyimpanan S3 Outposts dengan menggunakan enkripsi di sisi server dengan kunci enkripsi yang disediakan pelanggan (SSE-C). Untuk informasi selengkapnya, lihat Menggunakan enkripsi di sisi server dengan kunci yang disediakan pelanggan (SSE-C).

> **Note**
>
> S3 di Outposts tidak mendukung enkripsi sisi server AWS Key Management Service dengan kunci AWS KMS() (SSE-KMS).

Untuk informasi selengkapnya tentang S3 di Outposts, lihat Apa itu S3 di Outposts di Amazon S3 pada Panduan Pengguna Outposts.

Membandingkan kelas penyimpanan Amazon S3

Tabel berikut membandingkan kelas penyimpanan, termasuk ketersediaan, daya tahan, durasi penyimpanan minimum, dan pertimbangan lainnya.

Storage Class	Designed for	Durability (designed for)	Availability (designed for)	Availability Zones	Min storage duration	Min billable object size	Other Considerations
STANDARD	Frequently accessed data	99.999999999%	99.99%	>= 3	None	None	None
STANDARD_IA	Long-lived, infrequently accessed data	99.999999999%	99.9%	>= 3	30 days	128 KB	Per GB retrieval fees apply.
INTELLIGENT_TIERING	Long-lived data with changing or unknown access patterns	99.999999999%	99.9%	>= 3	30 days	None	Monitoring and automation fees per object apply. No retrieval fees.
ONEZONE_IA	Long-lived, infrequently accessed, non-critical data	99.999999999%	99.5%	1	30 days	128 KB	Per GB retrieval fees apply. Not resilient to the loss of the Availability Zone.
GLACIER	Long-term data archiving with retrieval times ranging from minutes to hours	99.999999999%	99.99% (after you restore objects)	>= 3	90 days	None	Per GB retrieval fees apply. You must first restore archived objects before you can access them. For more information, see Restoring Archived Objects.
DEEP_ARCHIVE	Archiving rarely accessed data with a default retrieval time of 12 hours	99.999999999%	99.99% (after you restore objects)	>= 3	180 days	None	Per GB retrieval fees apply. You must first restore archived objects before you can access them. For more information, see Restoring Archived Objects.
RRS (Not recommended)	Frequently accessed, non-critical data	99.99%	99.99%	>= 3	None	None	None

* S3 Glacier Flexible Retrieval membutuhkan 40 KB metadata tambahan untuk setiap objek yang diarsipkan. Hal ini termasuk metadata sebesar 32 KB yang dikenakan tarif pada tingkat S3 Glacier Flexible Retrieval (diperlukan untuk mengidentifikasi dan mengambil data), dan data tambahan sebesar 8 KB yang dikenakan tarif S3 Standard. Tarif S3 Standard diperlukan untuk mempertahankan nama dan metadata yang ditentukan pengguna untuk objek yang diarsipkan ke S3 Glacier Flexible Retrieval. Untuk informasi selengkapnya tentang kelas penyimpanan, lihat Kelas penyimpanan Amazon S3.

** S3 Glacier Deep Archive membutuhkan 40 KB metadata tambahan untuk setiap objek yang diarsipkan. Hal ini termasuk metadata sebesar 32 KB yang dikenakan tarif S3 Glacier Deep Archive (diperlukan untuk mengidentifikasi dan mengambil data), dan data tambahan sebesar 8 KB yang dikenakan tarif S3 Standard. Tarif S3 Standard diperlukan untuk mempertahankan nama dan metadata yang ditentukan pengguna untuk objek yang diarsipkan ke Amazon S3 Glacier Deep Archive. Untuk informasi selengkapnya tentang kelas penyimpanan, lihat Kelas penyimpanan Amazon S3.

Ketahuilah bahwa semua kelas penyimpanan kecuali untuk S3 One Zone-IA (`ONEZONE_IA`) dan S3 Express One Zone (`EXPRESS_ONEZONE`) dirancang agar tahan terhadap hilangnya fisik Availability Zone akibat bencana. Pertimbangkan juga biaya, selain kebutuhan kinerja dari skenario aplikasi. Untuk harga kelas penyimpanan, lihat harga Amazon S3.

Mengatur kelas penyimpanan objek

Anda dapat menentukan kelas penyimpanan untuk objek saat Anda mengunggahnya. Jika tidak, Amazon S3 menggunakan kelas penyimpanan Standar Amazon S3 default untuk objek dalam bucket tujuan umum. Anda juga dapat mengubah kelas penyimpanan objek yang sudah disimpan di bucket tujuan umum Amazon S3 ke kelas penyimpanan lain menggunakan konsol Amazon S3 AWS SDKs,, atau AWS Command Line Interface ().AWS CLI Semua pendekatan ini menggunakan operasi API Amazon S3 untuk mengirim permintaan ke Amazon S3.

> **Note**
>
> Anda tidak dapat mengubah kelas penyimpanan objek yang disimpan dalam bucket direktori.

Anda dapat mengarahkan Amazon S3 untuk mengubah kelas penyimpanan objek secara otomatis dengan menambahkan konfigurasi Siklus Hidup S3 ke bucket. Untuk informasi selengkapnya, lihat Mengelola siklus hidup objek.

Saat menyiapkan konfigurasi Replikasi S3, Anda dapat mengatur kelas penyimpanan untuk objek yang direplikasi ke kelas penyimpanan lainnya. Tetapi, Anda tidak dapat mereplikasi objek yang disimpan di kelas penyimpanan S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive. Untuk informasi selengkapnya, lihat Elemen file konfigurasi replikasi.

Saat mengatur kelas penyimpanan secara terprogram, Anda memberikan nilai kelas penyimpanan. Berikut ini adalah daftar nama konsol untuk kelas penyimpanan dengan nilai API yang sesuai:

- Mengurangi Penyimpanan Redundansi - `REDUCED_REDUNDANCY`
- S3 Express Satu Zona - `EXPRESS_ONEZONE`
- Arsip Dalam Gletser S3 — `DEEP_ARCHIVE`
- Pengambilan Fleksibel Gletser S3 - `GLACIER`
- Pengambilan Instan Gletser S3 - `GLACIER_IR`
- Tingkat Cerdas S3 — `INTELLIGENT_TIERING`
- S3 Satu Zona - IA - `ONEZONE_IA`
- Standar S3 - `STANDARD`
- S3 Standar-IA - `STANDARD_IA`

Mengatur kelas penyimpanan pada objek baru

Untuk mengatur kelas penyimpanan saat Anda mengunggah objek, Anda dapat menggunakan metode berikut.

Menggunakan konsol S3

Untuk mengatur kelas penyimpanan saat mengunggah objek baru di konsol:

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di:. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Dalam daftar bucket, pilih nama bucket tempat Anda ingin mengunggah folder atau file Anda.
4. Pilih Unggah.
5. Di jendela Upload, pilih Properties.
6. Di bawah kelas Penyimpanan, pilih kelas penyimpanan untuk file yang Anda unggah.
7. (Opsional) Mengonfigurasi properti tambahan apa pun untuk file yang Anda unggah, Untuk informasi selengkapnya, lihat Mengunggah Objek
8. Di jendela Unggah, lakukan salah satu hal berikut:
   - Seret file dan folder ke jendela Unggah.
   - Pilih Tambahkan file atau Tambahkan folder, pilih file atau folder untuk diunggah, dan pilih Buka.
9. Di bagian bawah halaman, Pilih Unggah.

Penggunaan API REST

Anda dapat menentukan kelas penyimpanan pada objek saat Anda membuatnya menggunakan operasi`PutObject`, `POST Object` Object, dan `CreateMultipartUpload` API, menambahkan header `x-amz-storage-class` permintaan. Jika Anda tidak menambahkan header ini, Amazon S3 menggunakan kelas penyimpanan standar S3 Standard (`STANDARD`) default.

Permintaan contoh ini menggunakan `PutObject` perintah untuk mengatur kelas penyimpanan pada objek baru ke S3 Intelligent-Tiering:

```
PUT /my-image.jpg HTTP/1.1
Host: amzn-s3-demo-bucket1.s3.Region.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: authorization string
Content-Type: image/jpeg
Content-Length: 11434
Expect: 100-continue
x-amz-storage-class: INTELLIGENT_TIERING
```

Menggunakan AWS CLI

Contoh ini menggunakan `put-object` perintah untuk meng-upload *my_images.tar.bz2* to *amzn-s3-demo-bucket1* di kelas *GLACIER* penyimpanan:

```
aws s3api put-object --bucket amzn-s3-demo-bucket1 --key dir-1/my_images.tar.bz2 --
storage-class GLACIER --body my_images.tar.bz2
```

Jika ukuran objek lebih dari 5 GB, gunakan perintah berikut untuk mengatur kelas penyimpanan:

```
aws s3 cp large_test_file s3://amzn-s3-demo-bucket1 --storage-class GLACIER
```

Mengubah kelas penyimpanan untuk objek yang ada

Untuk mengatur kelas penyimpanan saat Anda mengunggah objek, Anda dapat menggunakan metode berikut.

Menggunakan konsol S3

Anda dapat mengubah kelas penyimpanan objek menggunakan konsol Amazon S3 jika ukuran objek kurang dari 5 GB. Jika ukurannya lebih besar, sebaiknya tambahkan konfigurasi S3 Lifecycle untuk mengubah kelas penyimpanan objek.

Untuk mengubah kelas penyimpanan objek di konsol:

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Dalam daftar bucket, pilih nama bucket yang berisi objek yang ingin Anda ubah.
4. Pilih kotak centang di sebelah kiri nama objek yang ingin Anda ubah.
5. Pada menu Tindakan, pilih Edit kelas penyimpanan dari daftar opsi yang muncul.
6. Pilih dari kelas penyimpanan yang tersedia untuk objek Anda.
7. Di bawah Pengaturan salinan tambahan, pilih apakah Anda ingin Menyalin setelan sumber, Jangan tentukan pengaturan, atau Tentukan pengaturan. Salin pengaturan sumber adalah opsi default. Jika Anda hanya ingin menyalin objek tanpa atribut pengaturan sumber, pilih Jangan tentukan pengaturan. Pilih Tentukan pengaturan untuk menentukan pengaturan untuk kelas penyimpanan, tag objek ACLs, metadata, enkripsi sisi server, dan checksum tambahan.
8. Pilih Simpan perubahan di sudut kanan bawah. Amazon S3 menyimpan perubahan Anda.

Penggunaan API REST

Untuk mengubah kelas penyimpanan objek yang ada, gunakan metode berikut.

Permintaan contoh ini menggunakan `PutObject` perintah untuk mengatur kelas penyimpanan untuk objek yang ada ke S3 Intelligent-Tiering:

```
PUT /my-image.jpg HTTP/1.1
Host: amzn-s3-demo-bucket1.s3.Region.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: authorization string
Content-Type: image/jpeg
Content-Length: 11434
Expect: 100-continue
x-amz-storage-class: INTELLIGENT_TIERING
```

Menggunakan AWS CLI

Contoh ini menggunakan `cp` perintah untuk mengubah kelas penyimpanan objek yang ada dari kelas penyimpanan saat ini ke kelas *DEEP_ARCHIVE* penyimpanan:

```
aws s3 cp object_S3_URI object_S3_URI --storage-class DEEP_ARCHIVE
```

Membatasi izin kebijakan akses untuk kelas penyimpanan tertentu

Saat memberikan izin kebijakan akses untuk operasi Amazon S3, Anda dapat menggunakan kunci syarat `s3:x-amz-storage-class` untuk membatasi kelas penyimpanan mana yang akan digunakan saat menyimpan objek yang diunggah. Misalnya, saat Anda memberikan izin `s3:PutObject`, Anda dapat membatasi pengunggahan objek ke kelas penyimpanan tertentu. Untuk contoh kebijakan, lihat Contoh: Membatasi unggahan objek ke objek dengan kelas penyimpanan tertentu.

Untuk informasi selengkapnya tentang ketentuan penggunaan di dalam kebijakan dan daftar lengkap kunci syarat Amazon S3, lihat topik berikut:

- Kunci tindakan, sumber daya, dan kondisi untuk Amazon S3 di Referensi Otorisasi Layanan

  Untuk informasi selengkapnya tentang izin operasi S3 API menurut jenis sumber daya S3, lihat. Izin yang diperlukan untuk operasi API Amazon S3
- Contoh kebijakan bucket menggunakan tombol kondisi

Analitik Amazon S3–Analisis Kelas Penyimpanan

Dengan menggunakan analitik Amazon S3 Analisis Kelas Penyimpanan Anda dapat menganalisis pola akses penyimpanan untuk membantu memutuskan kapan harus memindahkan data yang tepat ke kelas penyimpanan yang tepat. Fitur analitik Amazon S3 yang baru ini mengamati pola akses data untuk membantu Anda menentukan kapan harus memindahkan penyimpanan STANDARD yang jarang diakses ke kelas penyimpanan STANDARD_IA (IA, untuk akses yang jarang). Untuk informasi selengkapnya tentang kelas penyimpanan, lihat Memahami dan mengelola kelas penyimpanan Amazon S3.

Setelah analisis kelas penyimpanan mengamati pola akses yang jarang dari kumpulan data yang difilter selama periode waktu tertentu, Anda dapat menggunakan hasil analisis untuk membantu Anda meningkatkan konfigurasi siklus hidup. Anda dapat mengonfigurasi analisis kelas penyimpanan untuk menganalisis semua objek dalam bucket. Atau, Anda dapat mengonfigurasi filter untuk mengelompokkan objek yang akan dianalisis berdasarkan prefiks umum (yaitu, objek yang memiliki nama yang dimulai dengan string umum), berdasarkan tanda objek, atau berdasarkan prefiks dan tanda. Anda akan mengetahui bahwa pemfilteran berdasarkan kelompok objek merupakan cara terbaik untuk memanfaatkan analisis kelas penyimpanan.

> **Important**
>
> Analisis kelas penyimpanan hanya memberikan rekomendasi untuk kelas Standar ke Standar IA.

Anda dapat memiliki banyak filter analisis kelas penyimpanan per bucket, hingga 1.000, dan setiap filter akan memberikan hasil analisis yang terpisah. Konfigurasi beberapa filter memungkinkan Anda menganalisis kelompok objek tertentu untuk meningkatkan konfigurasi siklus hidup yang memindahkan objek ke STANDARD_IA.

Analisis kelas penyimpanan memberikan visualisasi penggunaan penyimpanan di konsol Amazon S3 yang diperbarui setiap hari. Anda juga dapat mengekspor data penggunaan harian ini ke bucket S3 dan melihatnya dalam aplikasi spreadsheet, atau dengan alat intelijen bisnis, seperti. QuickSight

Anda akan dikenakan biaya yang terkait dengan analisis kelas penyimpanan. Untuk informasi harga, lihat Manajemen dan wawasan harga Amazon S3.

Topik

- Bagaimana cara menyiapkan analisis kelas penyimpanan?
- Bagaimana cara menggunakan analisis kelas penyimpanan?
- Bagaimana cara mengekspor data analisis kelas penyimpanan?
- Mengonfigurasi analisis kelas penyimpanan

Bagaimana cara menyiapkan analisis kelas penyimpanan?

Anda mengatur analisis kelas penyimpanan dengan mengonfigurasi data objek yang ingin dianalisis. Anda dapat mengonfigurasi analisis kelas penyimpanan untuk melakukan hal berikut:

- Menganalisis seluruh konten bucket.

  Anda akan menerima analisis untuk semua objek dalam bucket.
- Menganalisis objek yang dikelompokkan bersama berdasarkan prefiks dan tanda.

  Anda dapat mengonfigurasi filter yang mengelompokkan objek untuk analisis berdasarkan prefiks, tanda objek, atau kombinasi prefiks dan tanda. Anda akan menerima analisis terpisah untuk setiap filter yang dikonfigurasi. Anda dapat memiliki banyak konfigurasi filter per bucket, hingga 1.000.

- Mengekspor data analisis.

  Saat mengonfigurasi analisis kelas penyimpanan untuk bucket atau filter, Anda dapat memilih untuk mengekspor data analisis ke file setiap hari. Analisis untuk hari itu ditambahkan ke file agar membentuk log analisis historis untuk filter yang dikonfigurasi. File ini diperbarui setiap hari di tujuan pilihan Anda. Saat memilih data yang akan diekspor, Anda menentukan bucket tujuan dan prefiks tujuan opsional tempat file akan ditulis.

Anda dapat menggunakan konsol Amazon S3, REST API, atau AWS CLI atau AWS SDKs untuk mengonfigurasi analisis kelas penyimpanan.

- Untuk informasi tentang cara mengonfigurasi analisis kelas penyimpanan di konsol Amazon S3, lihat Mengonfigurasi analisis kelas penyimpanan.
- Untuk menggunakan Amazon S3 API, gunakan PutBucketAnalyticsConfigurationREST API, atau yang setara, dari atau. AWS CLI AWS SDKs

Bagaimana cara menggunakan analisis kelas penyimpanan?

Anda menggunakan analisis kelas penyimpanan untuk mengamati pola akses data dari waktu ke waktu untuk mengumpulkan informasi guna membantu Anda meningkatkan manajemen siklus hidup penyimpanan STANDARD_IA. Setelah mengonfigurasi filter, Anda akan mulai melihat analisis data berdasarkan filter di konsol Amazon S3 dalam 24 hingga 48 jam. Namun, analisis kelas penyimpanan mengamati pola akses dari set data yang difilter selama 30 hari atau lebih guna mengumpulkan informasi untuk digunakan dalam analisis sebelum memberikan hasil. Analisis terus berjalan setelah hasil awal dan akan memperbarui hasilnya seiring perubahan pola akses

Saat Anda pertama kali mengonfigurasi filter, konsol Amazon S3 mungkin memerlukan waktu untuk menganalisis data Anda.

Analisis kelas penyimpanan mengamati pola akses dari kumpulan data yang difilter selama 30 hari atau lebih guna mengumpulkan informasi yang cukup untuk analisis. Setelah analisis kelas penyimpanan mengumpulkan informasi yang cukup, Anda akan melihat pesan di konsol Amazon S3 bahwa analisis telah selesai.

Saat melakukan analisis untuk objek yang jarang diakses, analisis kelas penyimpanan melihat kumpulan objek yang dikelompokkan bersama berdasarkan umur sejak diunggah ke Amazon S3. Analisis kelas penyimpanan menentukan apakah kelompok umur jarang diakses dengan melihat faktor-faktor berikut untuk kumpulan data yang difilter:

- Objek dalam kelas penyimpanan STANDARD yang lebih besar dari 128 KB.
- Berapa rata-rata total penyimpanan yang Anda miliki per kelompok umur.
- Rata-rata jumlah byte yang ditransfer keluar (bukan frekuensi) per kelompok umur.
- Data ekspor analitik hanya mencakup permintaan dengan data yang relevan dengan analisis kelas penyimpanan. Hal ini dapat menyebabkan perbedaan dalam jumlah permintaan, dan total unggahan serta byte permintaan dibandingkan dengan yang ditunjukkan dalam metrik penyimpanan atau yang dilacak oleh sistem internal Anda sendiri.
- Permintaan GET dan PUT yang gagal tidak dihitung untuk analisis. Namun, Anda dapat melihat permintaan gagal dalam metrik penyimpanan.

Berapa Banyak Penyimpanan yang Saya Ambil?

Konsol Amazon S3 memetakan jumlah penyimpanan dalam kumpulan data yang difilter yang telah diambil untuk periode observasi.

Berapa Persentase Penyimpanan yang Saya Ambil?

Konsol Amazon S3 juga memetakan persentase penyimpanan dalam kumpulan data yang difilter yang telah diambil untuk periode observasi.

Seperti yang telah disebutkan sebelumnya dalam topik ini, saat Anda melakukan analisis untuk objek yang jarang diakses, analisis kelas penyimpanan melihat kumpulan objek yang telah difilter yang dikelompokkan berdasarkan umur sejak diunggah ke Amazon S3. Analisis kelas penyimpanan menggunakan kelompok umur objek yang sudah ditentukan berikut ini:

- Objek Amazon S3 dengan umur kurang dari 15 hari
- Objek Amazon S3 dengan umur 15-29 hari
- Objek Amazon S3 dengan umur 30-44 hari
- Objek Amazon S3 dengan umur 45-59 hari
- Objek Amazon S3 dengan umur 60-74 hari
- Objek Amazon S3 dengan umur 75-89 hari
- Objek Amazon S3 dengan umur 90-119 hari
- Objek Amazon S3 dengan umur 120-149 hari
- Objek Amazon S3 dengan umur 150-179 hari
- Objek Amazon S3 dengan umur 180-364 hari

- Objek Amazon S3 dengan umur 365-729 hari
- Objek Amazon S3 dengan umur 730 hari dan lebih lama

Biasanya dibutuhkan waktu sekitar 30 hari untuk mengamati pola akses untuk mengumpulkan informasi yang cukup sebagai hasil analisis. Mungkin dibutuhkan waktu lebih dari 30 hari, bergantung pada pola akses unik dari data Anda. Namun, setelah mengonfigurasi filter, Anda akan mulai melihat analisis data berdasarkan filter di konsol Amazon S3 dalam 24 hingga 48 jam. Anda dapat melihat analisis akses objek yang dipecah berdasarkan kelompok umur objek di konsol Amazon S3 setiap hari.

Berapa Banyak Penyimpanan Saya yang Jarang Diakses?

Konsol Amazon S3 menunjukkan pola akses yang dikelompokkan berdasarkan kelompok umur objek yang telah ditentukan sebelumnya. Keterangan Sering diakses atau Jarang diakses yang ditampilkan bertujuan untuk membantu Anda dalam proses pembuatan siklus hidup.

Bagaimana cara mengekspor data analisis kelas penyimpanan?

Anda dapat memilih agar analisis kelas penyimpanan mengekspor laporan analisis ke dalam file datar yang nilainya dipisahkan koma (CSV). Laporan diperbarui setiap hari dan didasarkan pada filter kelompok umur objek yang Anda konfigurasi. Saat menggunakan konsol Amazon S3, Anda dapat memilih opsi laporan ekspor ketika membuat filter. Saat memilih ekspor data, Anda harus menentukan bucket tujuan dan prefiks tujuan opsional tempat file ditulis. Anda dapat mengekspor data ke bucket tujuan di akun lain. Bucket tujuan harus berada di wilayah yang sama dengan bucket yang Anda konfigurasikan untuk dianalisis.

Anda harus membuat kebijakan bucket di bucket tujuan untuk memberikan izin ke Amazon S3 guna memverifikasi Akun AWS apa yang memiliki bucket dan menulis objek ke bucket di lokasi yang ditentukan. Untuk contoh kebijakan, lihat Berikan izin untuk Inventaris S3 dan analitik S3.

Setelah mengonfigurasi laporan analisis kelas penyimpanan, Anda akan mulai mendapatkan laporan yang diekspor setiap hari setelah 24 jam. Setelah itu, Amazon S3 akan terus memantau dan menyediakan ekspor harian.

Anda dapat membuka file CSV dalam aplikasi spreadsheet atau mengimpor file ke aplikasi lain seperti Amazon. QuickSight Untuk informasi tentang penggunaan file Amazon S3 dengan Amazon QuickSight, lihat Membuat Kumpulan Data Menggunakan File Amazon S3 di Panduan Pengguna Amazon QuickSight .

Data dalam file yang diekspor diurutkan berdasarkan tanggal dalam kelompok umur objek seperti yang ditunjukkan dalam contoh berikut. Jika kelas penyimpanannya STANDAR, baris juga berisi data untuk kolom `ObjectAgeForSIATransition` dan `RecommendedObjectAgeForSIATransition`.

Date	ConfigId	Filter	StorageClass	ObjectAge	ObjectCount	DataUploaded_MB	Storage_MB	DataRetrieved_MB	GetRequestCount	CumulativeAccessRatio	ObjectAgeForSIATransition	RecommendedObjectAgeForSIATransition
8/17/2021	SalesMaterial	SalesMaterial	STANDARD	000-014			0.4313			0		
9/2/2021	SalesMaterial	SalesMaterial	STANDARD	000-014						0.04096734		
8/22/2021	SalesMaterial	SalesMaterial	STANDARD	000-014			0.4313			0		
8/25/2021	SalesMaterial	SalesMaterial	STANDARD	000-014			0.4313			0		
9/6/2021	SalesMaterial	SalesMaterial	STANDARD	000-014						0.04096734		
8/30/2021	SalesMaterial	SalesMaterial	STANDARD	000-014						0.04096734		
8/28/2021	SalesMaterial	SalesMaterial	STANDARD	000-014						0.04096734		
8/21/2021	SalesMaterial	SalesMaterial	STANDARD	000-014			0.4313			0		
9/5/2021	SalesMaterial	SalesMaterial	STANDARD	000-014						0.04096734		

Di akhir laporan, kelompok umur objek diatur sebagai ALL. Baris ALL berisi total kumulatif, termasuk objek yang lebih kecil dari 128 KB, untuk semua kelompok umur untuk hari itu.

8/24/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0	000-014
9/3/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0.02426125	015-029
8/28/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0.03545875	015-029
8/17/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0	000-014
8/25/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0	000-014
9/6/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0.0209529	015-029
9/4/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0.02304819	015-029
8/22/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0	000-014
8/21/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0	000-014
8/30/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0.03073092	015-029
8/20/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0	000-014

Bagian selanjutnya menjelaskan kolom yang digunakan dalam laporan.

Tata letak file yang diekspor

Tabel berikut menjelaskan tata letak file ekspor analisis kelas penyimpanan Amazon S3.

Mengonfigurasi analisis kelas penyimpanan

Dengan menggunakan alat analisis kelas penyimpanan analitik Amazon S3, Anda dapat menganalisis pola akses penyimpanan untuk membantu memutuskan waktu yang tepat agar dapat memindahkan data yang tepat ke kelas penyimpanan yang tepat. Analisis kelas penyimpanan mengamati pola akses data untuk membantu Anda menentukan waktu untuk memindahkan penyimpanan STANDARD yang jarang diakses ke kelas penyimpanan STANDARD_IA (IA, untuk akses yang jarang). Untuk informasi selengkapnya tentang STANDARD_IA, lihat FAQ Amazon S3 dan Memahami dan mengelola kelas penyimpanan Amazon S3.

Anda mengatur analisis kelas penyimpanan dengan mengonfigurasi data objek yang ingin dianalisis. Anda dapat mengonfigurasi analisis kelas penyimpanan untuk melakukan hal berikut:

- Menganalisis seluruh konten bucket.

  Anda akan menerima analisis untuk semua objek dalam bucket.
- Menganalisis objek yang dikelompokkan bersama berdasarkan prefiks dan tanda.

Anda dapat mengonfigurasi filter yang mengelompokkan objek untuk analisis berdasarkan prefiks, tanda objek, atau kombinasi prefiks dan tanda. Anda akan menerima analisis terpisah untuk setiap filter yang dikonfigurasi. Anda dapat memiliki banyak konfigurasi filter per bucket, hingga 1.000.

- Mengekspor data analisis.

  Saat mengonfigurasi analisis kelas penyimpanan untuk bucket atau filter, Anda dapat memilih untuk mengekspor data analisis ke file setiap hari. Analisis untuk hari itu ditambahkan ke file agar membentuk log analisis historis untuk filter yang dikonfigurasi. File ini diperbarui setiap hari di tujuan pilihan Anda. Saat memilih data yang akan diekspor, Anda menentukan bucket tujuan dan prefiks tujuan opsional tempat file akan ditulis.

Anda dapat menggunakan konsol Amazon S3, REST API, atau AWS CLI atau AWS SDKs untuk mengonfigurasi analisis kelas penyimpanan.

> **Important**
>
> Analisis kelas penyimpanan tidak memberikan rekomendasi untuk pemindahan ke kelas penyimpanan ONEZONE_IA atau S3 Glacier Flexible Retrieval.
> Jika Anda ingin mengonfigurasi analisis kelas penyimpanan untuk mengekspor temuan Anda sebagai file.csv dan bucket tujuan menggunakan enkripsi bucket default dengan a AWS KMS key, Anda harus memperbarui kebijakan AWS KMS kunci untuk memberikan izin Amazon S3 untuk mengenkripsi file.csv. Untuk petunjuk, lihat Memberikan izin kepada Amazon S3 untuk menggunakan CMK untuk enkripsi.

Untuk informasi selengkapnya tentang analitik, lihat Analitik Amazon S3–Analisis Kelas Penyimpanan.

Menggunakan konsol S3

Untuk mengonfigurasi analisis kelas penyimpanan

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum atau ember Direktori.
3. Dalam daftar bucket, pilih nama bucket yang ingin Anda konfigurasikan analisis kelas penyimpanan.

4. Pilih tab Metrik.
5. Di Bawah Analisis Kelas Penyimpanan, pilih Buat konfigurasi analitik.
6. Ketik nama untuk filter. Jika Anda ingin menganalisis seluruh bucket, biarkan kolom Prefiks kosong.
7. Di kolom Prefiks, ketik teks untuk prefiks objek yang ingin Anda analisis.
8. Untuk menambahkan tanda, pilih Tambahkan tanda. Masukkan kunci dan nilai untuk tanda. Anda dapat memasukkan satu prefiks dan beberapa tanda.
9. Atau, Anda dapat memilih Aktifkan di bawah Ekspor CSV untuk mengekspor laporan analisis ke dalam file datar yang nilainya dipisahkan koma (.csv). Pilih bucket tujuan tempat file dapat disimpan. Anda dapat mengetikkan prefiks untuk bucket tujuan. Bucket tujuan harus Wilayah AWS sama dengan ember tempat Anda menyiapkan analisis. Bucket tujuan dapat berada di Akun AWS yang berbeda.

   Jika bucket tujuan untuk file.csv menggunakan enkripsi bucket default dengan kunci KMS, Anda harus memperbarui kebijakan AWS KMS kunci untuk memberikan izin Amazon S3 untuk mengenkripsi file.csv. Untuk petunjuk, lihat Memberikan izin kepada Amazon S3 untuk menggunakan CMK untuk enkripsi.
10. Pilih Buat Konfigurasi.

Amazon S3 membuat kebijakan bucket pada bucket tujuan yang memberikan izin penulisan kepada Amazon S3. Ini akan memungkinkannya untuk menulis data ekspor ke ember.

Apabila terjadi kesalahan saat mencoba membuat kebijakan bucket, Anda akan diberikan petunjuk untuk memperbaikinya. Misalnya, jika Anda memilih bucket tujuan di Akun AWS lain dan tidak memiliki izin untuk membaca dan menulis kebijakan bucket, Anda akan melihat pesan berikut. Anda harus meminta pemilik bucket tujuan menambahkan kebijakan bucket yang ditampilkan ke bucket tujuan. Jika kebijakan tersebut tidak ditambahkan ke bucket tujuan, Anda tidak akan mendapatkan data ekspor karena Amazon S3 tidak memiliki izin untuk menulis ke bucket tujuan. Jika bucket sumber dimiliki oleh akun yang berbeda dengan akun pengguna saat ini, ID akun yang benar dari bucket sumber harus diganti dalam kebijakan.

Untuk informasi tentang data yang diekspor dan cara kerja filter, lihat Analitik Amazon S3–Analisis Kelas Penyimpanan.

Penggunaan API REST

Untuk mengonfigurasi Analisis Kelas Penyimpanan menggunakan REST API, gunakan file PutBucketAnalyticsConfiguration. Anda juga dapat menggunakan operasi yang setara dengan AWS CLI atau AWS SDKs.

Anda dapat menggunakan REST berikut APIs untuk bekerja dengan Analisis Kelas Penyimpanan:

- Konfigurasi DELETE Bucket Analytics
- Kofigurasi GET Bucket Analytics
- Konfigurasi List Bucket Analytics

Mengelola biaya penyimpanan dengan Amazon S3 Intelligent-Tiering

Kelas penyimpanan S3 Intelligent-Tiering dirancang untuk mengoptimalkan biaya penyimpanan dengan secara otomatis memindahkan data ke jenjang akses yang paling hemat biaya saat pola akses berubah, tanpa biaya operasional atau berdampak pada kinerja. Untuk biaya pemantauan dan otomatisasi objek bulanan yang murah, S3 Intelligent-Tiering memantau pola akses dan secara otomatis memindahkan objek yang belum diakses ke jenjang akses berbiaya rendah.

S3 Intelligent-Tiering memberikan penghematan biaya penyimpanan otomatis dalam tiga latensi rendah dan tingkat akses throughput tinggi. Untuk data yang dapat diakses secara asinkron, Anda dapat memilih untuk mengaktifkan kemampuan pengarsipan otomatis di dalam kelas penyimpanan S3 Intelligent-Tiering. Tidak ada biaya pengambilan di S3 Intelligent-Tiering. Jika objek di tingkat Infrequent Access atau tingkat Archive Instant Access diakses nanti, objek tersebut secara otomatis dipindahkan kembali ke tingkat Frequent Access. Tidak ada biaya tingkat tambahan yang berlaku saat objek dipindahkan di antara tingkat akses dalam kelas penyimpanan S3 Intelligent-Tiering.

S3 Intelligent-Tiering adalah kelas penyimpanan yang direkomendasikan untuk data dengan pola akses yang tidak diketahui, berubah, atau tidak dapat diprediksi, terlepas dari ukuran objek atau periode retensi, seperti danau data, analisis data, dan aplikasi baru.

Kelas penyimpanan S3 Intelligent-Tiering mendukung semua fitur Amazon S3, termasuk yang berikut ini:

- Inventaris S3, untuk memverifikasi tingkat akses objek
- Replikasi S3, untuk mereplikasi data ke apa pun Wilayah AWS
- Lensa Penyimpanan S3, untuk melihat metrik penggunaan dan aktivitas penyimpanan

- Enkripsi di sisi server, untuk melindungi data objek
- Kunci Objek S3, untuk mencegah penghapusan data yang tidak disengaja
- AWS PrivateLink, untuk mengakses Amazon S3 melalui titik akhir pribadi di cloud privat virtual (VPC)

Untuk informasi tentang menggunakan S3 Intelligent-Tiering, lihat bagian berikut:

Topik

- Cara kerja S3 Intelligent-Tiering
- Menggunakan S3 Intelligent-Tiering
- Mengelola S3 Intelligent-Tiering

Cara kerja S3 Intelligent-Tiering

Amazon S3 Intelligent-Tiering secara otomatis menyimpan objek ke dalam tiga tingkatan akses. Satu tingkat dioptimalkan untuk akses yang sering, satu tingkat biaya lebih rendah dioptimalkan untuk akses yang tidak sering, dan tingkat biaya sangat rendah lainnya dioptimalkan untuk data yang jarang diakses. Untuk biaya pemantauan dan otomatisasi objek bulanan yang rendah, S3 Intelligent-Tiering memantau pola akses dan secara otomatis memindahkan objek ke tingkat Infrequent Access ketika objek belum diakses selama 30 hari berturut-turut. Setelah 90 hari tanpa akses, objek dipindahkan ke tingkat Archive Instant Access tanpa dampak kinerja atau overhead operasional.

Untuk mendapatkan biaya penyimpanan terendah untuk data yang dapat diakses dalam hitungan menit hingga jam, aktifkan kemampuan pengarsipan untuk menambahkan dua tingkatan akses tambahan. Anda dapat meratakan objek ke tingkat Archive Access, tingkat Deep Archive Access, atau keduanya. Dengan Archive Access, S3 Intelligent-Tiering memindahkan objek yang belum diakses selama minimal 90 hari berturut-turut ke tingkat Archive Access. Dengan Deep Archive Access, S3 Intelligent-Tiering memindahkan objek ke tingkat Deep Archive Access setelah minimal 180 hari berturut-turut tanpa akses. Untuk kedua tingkatan, Anda dapat mengonfigurasi jumlah hari tanpa akses berdasarkan kebutuhan Anda.

Tindakan berikut merupakan akses yang mencegah penurunan tingkat objek Anda ke tingkat Archive Access atau tingkat Deep Archive Access:

- Mengunduh atau menyalin objek melalui konsol Amazon S3.
- Memohon CopyObject, UploadPartCopy, atau mereplikasi objek dengan S3 Batch Replication. Dalam kasus ini, objek sumber dari operasi salinan atau replikasi naik tingkat.

- Memohon GetObject, PutObject, RestoreObject, CompleteMultipartUpload, ListParts, atau SelectObjectContent.

Misalnya, jika objek Anda diakses melalui `SelectObjectContent` sebelum jumlah hari tanpa akses yang ditentukan (misalnya, 180 hari), tindakan tersebut akan mengatur ulang pengatur waktu. Objek Anda tidak akan berpindah ke tingkat Archive Access atau tingkat Deep Archive Access hingga waktu setelah permintaan `SelectObjectContent` terakhir mencapai jumlah hari yang ditentukan.

Jika objek di tingkat Infrequent Access atau tingkat Archive Instant Access diakses nanti, objek tersebut secara otomatis dipindahkan kembali ke tingkat Frequent Access.

Tindakan berikut merupakan akses yang secara otomatis memindahkan objek dari tingkat Infrequent Access atau tingkat Archive Instant Access kembali ke tingkat Frequent Access:

- Mengunduh atau menyalin objek melalui konsol Amazon S3.
- Memohon CopyObject, UploadPartCopy, atau mereplikasi objek dengan Batch Replication. Dalam kasus ini, objek sumber dari operasi salinan atau replikasi naik tingkat.
- Memohon GetObject, PutObject, RestoreObject, CompleteMultipartUpload, atau ListParts.

Tindakan lain bukan merupakan akses yang secara otomatis memindahkan objek dari tingkat Infrequent Access atau tingkat Archive Instant Access kembali ke tingkat Frequent Access. Berikut ini adalah contoh, bukan daftar definitif, dari tindakan tersebut:

- Memohon HeadObject, GetObjectTagging, PutObjectTagging, ListObjects, ListObjectsV2, atau ListObjectVersions.
- Memohon SelectObjectContentbukan merupakan akses yang meningkatan objek hingga tingkat Akses Sering. Selain itu, ini tidak mencegah penurunan tingkat objek dari tingkat Frequent Access ke tingkat Infrequent Access, dan kemudian ke tingkat Archive Instant Access.

Anda dapat menggunakan S3 Intelligent-Tiering sebagai kelas penyimpanan default untuk data yang baru dibuat dengan menentukan `INTELLIGENT-TIERING` di header `x-amz-storage-class` permintaan saat memanggil,, atau operasi. `PutObject` `CopyObject` `CreateMultipartUpload` S3 Intelligent-Tiering dirancang untuk ketersediaan 99,9% dan daya tahan 99,999999999%.

> **Note**
>
> Jika ukuran suatu objek kurang dari 128 KB, objek tersebut tidak dipantau dan tidak memenuhi syarat untuk penetapan tingkat otomatis. Objek yang lebih kecil akan selalu disimpan di tingkat Frequent Access.

Tingkat akses S3 Intelligent-Tiering

Bagian berikut menjelaskan tingkat akses otomatis dan opsional yang berbeda. Ketika objek bergerak di antara tingkatan akses, kelas penyimpanan tetap sama (S3 Intelligent-Tiering).

Tingkat Frequent Access (otomatis)

Ini adalah tingkat akses default tempat objek apa pun yang dibuat atau ditransisikan ke S3 Intelligent-Tiering memulai siklus hidupnya. Sebuah objek tetap berada di tingkat ini selama sedang diakses. Tingkat Frequent Access memberikan latensi rendah dan kinerja dengan throughput tinggi.

Tingkat Infrequent Access (otomatis)

Jika sebuah objek tidak diakses selama 30 hari berturut-turut, objek berpindah ke tingkat Infrequent Access. Tingkat Infrequent Access memberikan latensi rendah dan kinerja dengan throughput tinggi.

Tingkat Archive Instant Access (otomatis)

Jika objek tidak diakses selama 90 hari berturut-turut, objek akan berpindah ke tingkat Archive Instant Access. Tingkat Archive Instant Access memberikan latensi rendah dan kinerja dengan throughput tinggi.

Tingkat Archive Access (opsional)

S3 Intelligent-Tiering memberi Anda opsi untuk mengaktifkan tingkat Archive Access untuk data yang dapat diakses secara asinkron. Setelah aktivasi, tingkat Archive Access secara otomatis akan mengarsipkan objek yang belum diakses selama minimal 90 hari berturut-turut. Anda dapat memperpanjang waktu akses terakhir untuk pengarsipan hingga maksimum 730 hari. Tingkat Archive Access memiliki kinerja yang sama dengan kelas penyimpanan S3 Glacier Flexible Retrieval.

Waktu pengambilan standar untuk tingkat akses ini dapat berkisar antara 3-5 jam. Jika Anda memulai permintaan pemulihan dengan menggunakan Operasi Batch S3, pemulihan dimulai

dalam beberapa menit. Untuk informasi selengkapnya tentang opsi dan waktu pengambilan, lihat the section called “Memulihkan objek dari tingkat S3 Intelligent-Tiering Archive Access dan Deep Archive Access”.

> **Note**
>
> Hanya aktifkan tingkat Archive Access selama 90 hari jika Anda ingin melewati tingkat Archive Instant Access. Tingkat Akses Arsip memberikan biaya penyimpanan yang sedikit lebih rendah, dengan waktu minute-to-hour pengambilan. Tingkat Archive Instant Access memberikan akses milidetik dan kinerja dengan throughput tinggi.

Tingkat Deep Archive Access (opsional)

S3 Intelligent-Tiering memberi Anda opsi untuk mengaktifkan tingkat Deep Archive Access untuk data yang dapat diakses secara asinkron. Setelah aktivasi, tingkat Deep Archive Access secara otomatis akan mengarsipkan objek yang belum diakses selama minimal 180 hari berturut-turut. Anda dapat memperpanjang waktu akses terakhir untuk pengarsipan hingga maksimum 730 hari. Tingkat Deep Archive Access memiliki kinerja yang sama dengan kelas penyimpanan S3 Glacier Deep Archive.

Pengambilan standar objek dalam tingkat akses ini terjadi dalam waktu 12 jam. Jika Anda memulai permintaan pemulihan dengan menggunakan Operasi Batch S3, pemulihan dimulai dalam 9 jam. Untuk informasi selengkapnya tentang opsi dan waktu pengambilan, lihat the section called “Memulihkan objek dari tingkat S3 Intelligent-Tiering Archive Access dan Deep Archive Access”.

> **Note**
>
> Aktifkan tingkatan Archive Access dan Deep Archive Access hanya jika objek dapat diakses secara asinkron oleh aplikasi Anda. Jika objek yang Anda ambil disimpan di tingkat Archive Access atau Deep Archive Access, Anda harus terlebih dahulu memulihkan objek dengan menggunakan operasi `RestoreObject`.

Menggunakan S3 Intelligent-Tiering

Anda dapat menggunakan kelas penyimpanan S3 Intelligent-Tiering untuk mengoptimalkan biaya penyimpanan secara otomatis. S3 Intelligent-Tiering memberikan penghematan biaya otomatis dengan memindahkan data pada tingkat objek granular antar tingkat akses ketika pola akses berubah. Untuk data yang dapat diakses secara asinkron, Anda dapat memilih untuk mengaktifkan pengarsipan otomatis dalam kelas penyimpanan S3 Intelligent-Tiering menggunakan,, atau Amazon S3 API. AWS Management Console AWS CLI

Memindahkan data ke S3 Intelligent-Tiering

Ada dua cara untuk memindahkan data ke S3 Intelligent-Tiering. Anda dapat mengunggah objek langsung ke S3 Intelligent-Tiering dari konsol atau secara terprogram menggunakan operasi. PUT Untuk informasi selengkapnya, lihat Mengatur kelas penyimpanan objek. Anda juga dapat mengonfigurasi konfigurasi Siklus Hidup S3 ke objek transisi dari Standar S3 atau Akses Jarang Standar S3 ke S3 Intelligent-Tiering.

Mengunggah data ke S3 Intelligent-Tiering menggunakan Direct PUT

Saat Anda mengunggah objek ke kelas penyimpanan S3 Intelligent-Tiering menggunakan operasi API PUT, Anda menentukan S3 Intelligent-Tiering di header permintaan `x-amz-storage-class`.

Permintaan berikut menyimpan citra, `my-image.jpg`, di bucket `myBucket`. Permintaan menggunakan header `x-amz-storage-class` untuk meminta objek disimpan menggunakan kelas penyimpanan S3 Intelligent-Tiering.

Example

```
PUT /my-image.jpg HTTP/1.1
Host: myBucket.s3.<Region>.amazonaws.com (http://amazonaws.com/)
Date: Wed, 1 Sep 2021 17:50:00 GMT
Authorization: authorization string
Content-Type: image/jpeg
Content-Length: 11434
Expect: 100-continue
x-amz-storage-class: INTELLIGENT_TIERING
```

Transisi data ke S3 Intelligent-Tiering dari S3 Standard atau S3 Standard-Infrequent Access menggunakan Siklus Hidup S3

Anda dapat menambahkan aturan ke konfigurasi Siklus Hidup S3 untuk memberi tahu Amazon S3 untuk mentransisi objek satu kelas penyimpanan ke kelas penyimpanan lainnya. Untuk informasi tentang transisi yang didukung dan batasan terkait, lihat Mentransisikan objek menggunakan Siklus Hidup S3.

Anda dapat menentukan konfigurasi Siklus Hidup S3 pada tingkat bucket atau awalan. Dalam aturan konfigurasi Siklus Hidup S3 ini, filter menentukan awalan kunci (`documents/`). Oleh karena itu, aturan berlaku untuk objek dengan awalan nama kunci `documents/`, seperti `documents/doc1.txt` dan `documents/doc2.txt`. Aturan menentukan tindakan `Transition` yang mengarahkan Amazon S3 untuk mentransisi objek ke kelas penyimpanan S3 Intelligent-Tiering 0 hari setelah pembuatan. Dalam hal ini, objek memenuhi syarat untuk transisi ke S3 Intelligent-Tiering pada tengah malam UTC setelah pembuatan.

Example

```
<LifecycleConfiguration>
  <Rule>
    <ID>ExampleRule</ID>
    <Filter>
       <Prefix>documents/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>0</Days>
      <StorageClass>INTELLIGENT_TIERING</StorageClass>
    </Transition>
 </Rule>
</LifecycleConfiguration>
```

Bucket yang dengan dukungan Penentuan Versi mempertahankan satu versi objek saat ini, dan nol atau lebih versi objek yang lama. Anda dapat menentukan aturan Siklus Hidup terpisah untuk versi objek saat ini dan yang lama.

Untuk informasi selengkapnya, lihat Elemen konfigurasi Siklus Hidup.

Mengaktifkan tingkatan S3 Intelligent-Tiering Archive Access dan Deep Archive Access

Untuk mendapatkan biaya penyimpanan terendah pada data yang dapat diakses dalam hitungan menit hingga jam, Anda dapat mengaktifkan salah satu atau kedua tingkatan akses arsip dengan

membuat konfigurasi tingkat bucket, awalan, atau tag objek menggunakan AWS Management Console, AWS CLI, atau Amazon S3 API.

Menggunakan konsol S3

Untuk mengaktifkan pengarsipan otomatis S3 Intelligent-Tiering

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Dari daftar Bucket, pilih nama bucket yang Anda inginkan.
3. Pilih Properti.
4. Buka bagian Konfigurasi S3 Intelligent-Tiering Archive dan pilih Buat configuration.
5. Di bagian Pengaturan konfigurasi arsip, tentukan nama konfigurasi deskriptif untuk konfigurasi S3 Intelligent-Tiering Archive Anda.
6. Di bawah Pilih cakupan konfigurasi, pilih cakupankonfigurasi yang akan digunakan. Secara opsional, Anda dapat membatasi cakupan konfigurasi ke objek tertentu dalam bucket menggunakan awalan bersama, tanda objek, atau kombinasi keduanya.
   a. Untuk membatasi cakupan konfigurasi, pilih Batasi cakupan konfigurasi ini menggunakan satu atau beberapa filter.
   b. Untuk membatasi cakupan konfigurasi menggunakan awalan tunggal, masukkan awalan di bawah Awalan.
   c. Untuk membatasi cakupan konfigurasi menggunakan tanda objek, pilih Tambahkan tanda dan masukkan nilai untuk Kunci.
7. Di bawah Status, pilih Aktifkan.
8. Di bagian Pengaturan arsip, pilih salah satu atau kedua tingkatan Archive Access yang akan diaktifkan.
9. Pilih Buat.

Menggunakan AWS CLI

Anda dapat menggunakan AWS CLI perintah berikut untuk mengelola konfigurasi S3 Intelligent-Tiering:

- `delete-bucket-intelligent-tiering-configuration`
- `get-bucket-intelligent-tiering-configuration`

- `list-bucket-intelligent-tiering-configurations`
- `put-bucket-intelligent-tiering-configuration`

Untuk petunjuk cara menyiapkan AWS CLI, lihat Mengembangkan dengan Amazon S3 menggunakan AWS CLI di Referensi API Amazon S3.

Saat menggunakan AWS CLI, Anda tidak dapat menentukan konfigurasi sebagai file XHTML. Anda harus menentukan JSON sebagai gantinya. Berikut ini adalah contoh konfigurasi XML S3 Intelligent-Tiering dan JSON setara yang dapat Anda tentukan dalam perintah AWS CLI .

Contoh berikut menempatkan konfigurasi S3 Intelligent-Tiering ke bucket tertentu.

Example **`put-bucket-intelligent-tiering-configuration`**

JSON

```
{
  "Id": "string",
  "Filter": {
    "Prefix": "string",
    "Tag": {
      "Key": "string",
      "Value": "string"
    },
    "And": {
      "Prefix": "string",
      "Tags": [
        {
          "Key": "string",
          "Value": "string"
        }
        ...
      ]
    }
  },
  "Status": "Enabled"|"Disabled",
  "Tierings": [
    {
      "Days": integer,
      "AccessTier": "ARCHIVE_ACCESS"|"DEEP_ARCHIVE_ACCESS"
    }
    ...
```

```
  ]
}
```

XML

```
PUT /?intelligent-tiering&id=Id HTTP/1.1
Host: Bucket.s3.amazonaws.com
<?xml version="1.0" encoding="UTF-8"?>
<IntelligentTieringConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
   <Id>string</Id>
   <Filter>
      <And>
         <Prefix>string</Prefix>
         <Tag>
            <Key>string</Key>
            <Value>string</Value>
         </Tag>
         ...
      </And>
      <Prefix>string</Prefix>
      <Tag>
         <Key>string</Key>
         <Value>string</Value>
      </Tag>
   </Filter>
   <Status>string</Status>
   <Tiering>
      <AccessTier>string</AccessTier>
      <Days>integer</Days>
   </Tiering>
   ...
</IntelligentTieringConfiguration>
```

Menggunakan operasi API PUT

Anda dapat menggunakan operasi PutBucketIntelligentTieringConfiguration untuk bucket tertentu dan hingga 1.000 konfigurasi S3 Intelligent-Tiering per bucket. Anda dapat menentukan objek di dalam bucket yang memenuhi syarat untuk tingkat Archive Access menggunakan awalan bersama atau tanda objek. Menggunakan awalan bersama atau tanda objek memungkinkan Anda untuk menyelaraskan ke aplikasi bisnis, alur kerja, atau organisasi internal

tertentu. Anda juga memiliki fleksibilitas untuk mengaktifkan tingkat Archive Access, tingkat Deep Archive Access, atau keduanya.

Memulai S3 Intelligent-Tiering

Untuk mempelajari cara menggunakan S3 Intelligent-Tiering lebih lanjut, lihat Tutorial: Mulai menggunakan S3 Intelligent-Tiering.

Mengelola S3 Intelligent-Tiering

Kelas penyimpanan S3 Intelligent-Tiering memberikan penghematan biaya penyimpanan otomatis dalam tiga tingkat akses latensi rendah dan throughput tinggi. Ini juga menawarkan kemampuan arsip opsional untuk membantu Anda mendapatkan biaya penyimpanan terendah di cloud untuk data yang dapat diakses dalam hitungan menit hingga jam.

Mengidentifikasi objek tingkat akses S3 Intelligent-Tiering yang disimpan di dalamnya

Untuk mendapatkan daftar objek dan metadata yang sesuai, termasuk tingkat akses S3 Intelligent-Tiering, Anda dapat menggunakan Inventaris Amazon S3. S3 Inventory menyediakan CSV, ORC, atau Parquet output file yang mencantumkan objek Anda dan metadata yang sesuai. Anda dapat menerima laporan inventaris ini setiap hari atau setiap pekan untuk bucket Amazon S3 atau awalan bersama. (Awalan bersama mengacu pada objek yang memiliki nama yang dimulai dengan string umum.)

Melihat status arsip objek dalam S3 Intelligent-Tiering

Untuk menerima pemberitahuan ketika objek dalam kelas penyimpanan S3 Intelligent-Tiering telah berpindah ke tingkat Archive Access atau tingkat Deep Archive Access, Anda dapat mengatur Notifikasi Peristiwa S3. Untuk informasi selengkapnya, lihat Mengaktifkan pemberitahuan peristiwa.

Amazon S3 dapat menerbitkan pemberitahuan kejadian ke topik Amazon Simple Notification Service (Amazon SNS), atau fungsi Amazon Simple Queue Service (Amazon SQS), atau fungsi AWS Lambda . Untuk informasi selengkapnya, lihat Notifikasi Peristiwa Amazon S3.

Berikut ini adalah contoh pesan yang dikirimkan Amazon S3 untuk menerbitkan peristiwa `s3:IntelligentTiering`. Untuk informasi selengkapnya, lihat the section called “Struktur pesan peristiwa”.

```
{
   "Records":[
      {
```

```
         "eventVersion":"2.3",
         "eventSource":"aws:s3",
         "awsRegion":"us-west-2",
         "eventTime":"1970-01-01T00:00:00.000Z",
         "eventName":"IntelligentTiering",
         "userIdentity":{
            "principalId":"s3.amazonaws.com"
         },
         "requestParameters":{
            "sourceIPAddress":"s3.amazonaws.com"
         },
         "responseElements":{
            "x-amz-request-id":"C3D13FE58DE4C810",
            "x-amz-id-2":"FMyUVURIY8/IgAtTv8xRjskZQpcIZ9KG4V5Wp6S7S/
JRWeUWerMUE5JgHvANOjpD"
         },
         "s3":{
            "s3SchemaVersion":"1.0",
            "configurationId":"testConfigRule",
            "bucket":{
               "name":"amzn-s3-demo-bucket",
               "ownerIdentity":{
                  "principalId":"A3NL1KOZZKExample"
               },
               "arn":"arn:aws:s3:::amzn-s3-demo-bucket"
            },
            "object":{
               "key":"HappyFace.jpg",
               "size":1024,
               "eTag":"d41d8cd98f00b204e9800998ecf8427e",
            }
         },
         "intelligentTieringEventData":{
            "destinationAccessTier": "ARCHIVE_ACCESS"
            }
      }
   ]
}
```

Anda juga dapat menggunakan Permintaan objek HEAD untuk melihat status arsip objek. Jika objek disimpan di kelas penyimpanan S3 Intelligent-Tiering dan berada di salah satu tingkatan arsip, respons HEAD objek menunjukkan tingkat arsip saat ini. Untuk menampilkan tingkat arsip, permintaan menggunakan x-amz-archive-statussundulan.

Permintaan objek HEAD berikut mengembalikan metadata objek (dalam hal ini, *my-image.jpg*).

Example

```
HEAD /my-image.jpg HTTP/1.1
Host: bucket.s3.region.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:02236Q3V0RonhpaBX5sCYVf1bNRuU=
```

Anda juga dapat menggunakan permintaan objek HEAD untuk memantau status permintaan `restore-object`. Jika restorasi arsip sedang berlangsung, respons HEAD objek mencakup x-amz-restoresundulan.

Respons objek HEAD sampel berikut menunjukkan objek yang diarsipkan menggunakan S3 Intelligent-Tiering dengan permintaan pemulihan yang sedang berlangsung.

Example

```
HTTP/1.1 200 OK
x-amz-id-2: FSVaTMjrmBp3Izs1NnwBZeu7M19iI8UbxMbi0A8AirHANJBo+hEftBuiESACOMJp
x-amz-request-id: E5CEFCB143EB505A
Date: Fri, 13 Nov 2020 00:28:38 GMT
Last-Modified: Mon, 15 Oct 2012 21:58:07 GMT
ETag: "1accb31fcf202eba0c0f41fa2f09b4d7"
x-amz-storage-class: 'INTELLIGENT_TIERING'
x-amz-archive-status: 'ARCHIVE_ACCESS'
x-amz-restore: 'ongoing-request="true"'
x-amz-restore-request-date: 'Fri, 13 Nov 2020 00:20:00 GMT'
Accept-Ranges: bytes
Content-Type: binary/octet-stream
Content-Length: 300
Server: AmazonS3
```

Memulihkan objek dari tingkat S3 Intelligent-Tiering Archive Access dan Deep Archive Access

Untuk mengakses objek di tingkat S3 Intelligent-Tiering Archive Access dan Deep Archive Access, Anda harus memulai permintaan pemulihan, dan kemudian menunggu hingga objek dipindahkan ke tingkat Frequent Access. Untuk informasi selengkapnya tentang objek yang diarsipkan, lihatthe section called “Bekerja dengan objek yang diarsipkan”.

Saat memulihkan objek dari tingkat Akses Arsip atau tingkat Akses Arsip Dalam, objek akan kembali ke tingkat Akses Sering. Setelah itu, jika objek tidak diakses selama 30 hari berturut-turut, objek akan

secara otomatis berpindah ke tingkat Infrequent Access. Kemudian, setelah minimal 90 hari berturut-turut tanpa akses, objek berpindah ke tingkat Archive Access. Setelah minimal 180 hari berturut-turut tanpa akses, objek berpindah ke tingkat Deep Archive Access. Untuk informasi selengkapnya, lihat the section called “Cara kerja S3 Intelligent-Tiering”.

Anda dapat memulihkan objek yang diarsipkan menggunakan konsol Amazon S3, Operasi Batch S3, API REST Amazon S3, AWS SDKs the, atau (). AWS Command Line Interface AWS CLI Untuk informasi selengkapnya, lihat the section called “Bekerja dengan objek yang diarsipkan”.

Memahami kelas penyimpanan S3 Glacier untuk penyimpanan data jangka panjang

Anda dapat menggunakan kelas penyimpanan Amazon S3 S3 Glacier untuk memberikan solusi hemat biaya untuk menyimpan data jangka panjang yang tidak sering diakses. Kelas penyimpanan S3 Glacier adalah:

- S3 Glacier Instant Retrieval
- S3 Glacier Flexible Retrieval
- S3 Glacier Deep Archive

Anda memilih salah satu kelas penyimpanan ini berdasarkan seberapa sering Anda mengakses data Anda dan seberapa cepat Anda perlu mengambilnya. Masing-masing kelas penyimpanan ini menawarkan daya tahan dan ketahanan yang sama dengan kelas penyimpanan Standar S3, tetapi dengan biaya penyimpanan yang lebih rendah. Untuk informasi lebih lanjut tentang kelas penyimpanan S3 Glacier, lihat kelas penyimpanan/gletser/. https://aws.amazon.com/s3/

Topik

- Membandingkan kelas penyimpanan S3 Glacier
- S3 Glacier Instant Retrieval
- S3 Glacier Flexible Retrieval
- S3 Glacier Deep Archive
- Memahami penyimpanan arsip di S3 Glacier Flexible Retrieval dan S3 Glacier Deep Archive
- Bagaimana kelas penyimpanan ini berbeda dari layanan S3 Glacier

Membandingkan kelas penyimpanan S3 Glacier

Setiap kelas penyimpanan S3 Glacier memiliki durasi penyimpanan minimum untuk semua objek. Jika Anda menghapus, menimpa, atau mentransisikan objek ke kelas penyimpanan yang berbeda sebelum minimum, Anda akan dikenakan biaya untuk sisa durasi tersebut.

Beberapa kelas penyimpanan S3 Glacier adalah arsip, yang berarti objek yang disimpan di kelas tersebut diarsipkan dan tidak tersedia untuk akses real-time. Untuk informasi selengkapnya, lihat Memahami penyimpanan arsip di S3 Glacier Flexible Retrieval dan S3 Glacier Deep Archive.

Kelas penyimpanan yang dirancang untuk pola akses yang lebih jarang dengan waktu pengambilan yang lebih lama menawarkan biaya penyimpanan yang lebih rendah. Untuk informasi harga, lihathttps://aws.amazon.com/s3/pricing/.

Tabel berikut merangkum poin-poin penting yang perlu dipertimbangkan ketika memilih kelas penyimpanan S3 Glacier:

S3 Glacier Instant Retrieval

Kami merekomendasikan penggunaan S3 Glacier Instant Retrieval untuk data jangka panjang yang diakses sekali per kuartal dan membutuhkan waktu pengambilan milidetik. Kelas penyimpanan ini sangat ideal untuk kasus penggunaan yang sensitif terhadap kinerja seperti hosting gambar, aplikasi berbagi file, dan menyimpan catatan medis untuk akses selama janji temu.

Kelas penyimpanan S3 Glacier Instant Retrieval menawarkan akses real-time ke objek Anda dengan kinerja latensi dan throughput yang sama dengan kelas penyimpanan IA Standar S3. Jika dibandingkan dengan S3 Standard-IA, S3 Glacier Instant Retrieval memiliki biaya penyimpanan yang lebih rendah tetapi biaya akses data yang lebih tinggi.

Ada ukuran objek minimum 128 KB untuk data yang disimpan di kelas penyimpanan S3 Glacier Instant Retrieval. Kelas penyimpanan ini juga memiliki jangka waktu penyimpanan minimum 90 hari.

S3 Glacier Flexible Retrieval

Kami merekomendasikan penggunaan S3 Glacier Flexible Retrieval untuk data arsip yang diakses satu hingga dua kali setahun dan tidak memerlukan akses langsung. S3 Glacier Flexible Retrieval menawarkan waktu pengambilan yang fleksibel untuk membantu Anda menyeimbangkan biaya, dengan waktu akses mulai dari beberapa menit hingga jam, dan pengambilan massal gratis. Kelas penyimpanan ini sangat ideal untuk cadangan dan pemulihan bencana.

Objek yang disimpan di S3 Glacier Flexible Retrieval diarsipkan dan tidak tersedia untuk akses real-time. Untuk informasi selengkapnya, lihat Memahami penyimpanan arsip di S3 Glacier Flexible Retrieval dan S3 Glacier Deep Archive. Untuk mengakses objek ini, pertama-tama Anda memulai permintaan pemulihan yang membuat salinan sementara objek yang dapat Anda akses saat permintaan selesai. Untuk informasi, lihat Bekerja dengan objek yang diarsipkan. Saat memulihkan objek, Anda dapat memilih tingkat pengambilan untuk memenuhi kasus penggunaan Anda, dengan biaya lebih rendah untuk waktu pemulihan yang lebih lama.

Tingkatan pengambilan berikut tersedia untuk S3 Glacier Flexible Retrieval:

- Pengambilan dipercepat — Biasanya mengembalikan objek dalam 1—5 menit. Pengambilan yang dipercepat tunduk pada permintaan, jadi untuk memastikan Anda memiliki waktu pemulihan yang andal dan dapat diprediksi, kami sarankan Anda membeli kapasitas pengambilan yang disediakan. Untuk informasi selengkapnya, lihat Kapasitas yang disediakan.
- Pengambilan standar — Biasanya mengembalikan objek dalam 3-5 jam, atau dalam 1 menit hingga 5 jam saat Anda menggunakan Operasi Batch S3. Untuk informasi selengkapnya, lihat Memulihkan objek dengan Operasi Batch.
- Pengambilan massal — Biasanya mengembalikan objek dalam waktu 5-12 jam. Pengambilan massal gratis.

Durasi penyimpanan minimum untuk objek di kelas penyimpanan S3 Glacier Flexible Retrieval adalah 90 hari.

Pengambilan Fleksibel Gletser S3 membutuhkan 40 KB metadata tambahan untuk setiap objek. Ini termasuk 32 KB metadata yang diperlukan untuk mengidentifikasi dan mengambil data Anda, yang dibebankan pada tingkat default untuk S3 Glacier Flexible Retrieval. Data 8 KB tambahan diperlukan untuk mempertahankan nama dan metadata yang ditentukan pengguna untuk objek yang diarsipkan, dan dibebankan pada tarif Standar S3.

S3 Glacier Deep Archive

Sebaiknya gunakan S3 Glacier Deep Archive untuk arsip data yang diakses kurang dari setahun sekali. Kelas penyimpanan ini dirancang untuk mempertahankan kumpulan data selama beberapa tahun untuk memenuhi persyaratan kepatuhan dan juga dapat digunakan untuk cadangan atau pemulihan bencana atau data yang jarang diakses yang dapat Anda tunggu hingga 72 jam untuk diambil. S3 Glacier Deep Archive adalah opsi penyimpanan dengan biaya terendah di AWS.

Objek yang disimpan di S3 Glacier Deep Archive diarsipkan dan tidak tersedia untuk akses real-time. Untuk informasi selengkapnya, lihat Memahami penyimpanan arsip di S3 Glacier Flexible Retrieval dan S3 Glacier Deep Archive. Untuk mengakses objek ini, pertama-tama Anda memulai permintaan pemulihan yang membuat salinan sementara objek yang dapat Anda akses saat permintaan selesai. Untuk informasi, lihat Bekerja dengan objek yang diarsipkan. Saat memulihkan objek, Anda dapat memilih tingkat pengambilan untuk memenuhi kasus penggunaan Anda, dengan biaya lebih rendah untuk waktu pemulihan yang lebih lama.

Tingkatan pengambilan berikut tersedia untuk S3 Glacier Deep Archive:

- Pengambilan standar — Biasanya mengembalikan objek dalam waktu 12 jam, atau dalam waktu 9-12 jam saat Anda menggunakan Operasi Batch S3. Untuk informasi selengkapnya, lihat Memulihkan objek dengan Operasi Batch.
- Pengambilan massal — Biasanya mengembalikan objek dalam waktu 48 jam dengan sebagian kecil dari biaya tingkat pengambilan Standar.

Durasi penyimpanan minimum untuk objek di kelas penyimpanan S3 Glacier Deep Archive adalah 180 hari.

S3 Glacier Deep Archive membutuhkan 40 KB metadata tambahan untuk setiap objek. Ini termasuk 32 KB metadata yang diperlukan untuk mengidentifikasi dan mengambil data Anda, yang dibebankan pada tarif default untuk S3 Glacier Deep Archive. Data 8 KB tambahan diperlukan untuk mempertahankan nama dan metadata yang ditentukan pengguna untuk objek yang diarsipkan, dan dibebankan pada tarif Standar S3.

Memahami penyimpanan arsip di S3 Glacier Flexible Retrieval dan S3 Glacier Deep Archive

S3 Glacier Flexible Retrieval dan S3 Glacier Deep Archive adalah kelas penyimpanan arsip. Ini berarti bahwa ketika Anda menyimpan objek di kelas penyimpanan ini objek tersebut diarsipkan, dan tidak dapat diakses secara langsung. Untuk mengakses objek yang diarsipkan, Anda mengirimkan permintaan pemulihan untuk itu, dan kemudian menunggu layanan memulihkan objek. Permintaan pemulihan mengembalikan salinan sementara objek, dan salinan itu dihapus ketika durasi yang Anda tentukan dalam permintaan berakhir. Untuk informasi selengkapnya, lihat Bekerja dengan objek yang diarsipkan.

Transisi objek ke kelas penyimpanan S3 Glacier Deep Archive hanya dapat menggunakan satu cara.

Jika Anda ingin mengubah kelas penyimpanan dari objek yang diarsipkan ke kelas penyimpanan lain, Anda harus menggunakan operasi pemulihan untuk membuat salinan sementara dari objek terlebih dahulu. Kemudian gunakan operasi penyalinan untuk menimpa objek yang menentukan S3 Standard, S3 Intelligent-Tiering, S3 Standard-IA, S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval, atau Reduced Redundancy Storage sebagai kelas penyimpanan.

> **Note**
>
> Operasi Salin untuk objek yang dipulihkan tidak didukung di konsol S3 Amazon untuk objek di kelas penyimpanan S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive. Untuk jenis operasi Copy ini, gunakan AWS Command Line Interface (AWS CLI), the AWS SDKs, atau REST API.

Anda dapat memulihkan objek yang diarsipkan di kelas penyimpanan ini dengan hingga 1.000 transaksi per detik (TPS) permintaan pemulihan objek per akun per akun per. Wilayah AWS

Pertimbangan biaya

Jika Anda berencana untuk mengarsipkan data yang jarang diakses selama beberapa bulan atau tahun, kelas penyimpanan S3 Glacier Flexible Retrieval dan S3 Glacier Deep Archive dapat mengurangi biaya penyimpanan Anda. Namun, untuk memastikan bahwa kelas penyimpanan S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive sesuai bagi Anda, pertimbangkan hal berikut ini:

- Biaya overhead penyimpanan — Setiap objek yang diarsipkan membutuhkan 40 KB metadata tambahan. Ini termasuk 32 KB metadata yang diperlukan untuk mengidentifikasi dan mengambil data Anda, yang dibebankan pada tingkat default untuk kelas penyimpanan tersebut. Data 8 KB tambahan diperlukan untuk mempertahankan nama dan metadata yang ditentukan pengguna untuk objek yang diarsipkan, dan dibebankan pada tarif Standar S3.

  Jika Anda mengarsipkan objek kecil, pertimbangkan biaya penyimpanan ini. Pertimbangkan juga untuk menggabungkan banyak objek kecil menjadi sejumlah kecil objek besar untuk mengurangi biaya overhead.
- Harga unggahan multipart — Objek di S3-storage-class-glacier; dan S3 Glacier Deep Archive ditagih dengan tarif kelas penyimpanan Standar S3 saat Anda mengunggahnya menggunakan unggahan multipart. Untuk informasi selengkapnya, lihat Unggahan multibagian dan harga.

- Biaya penyimpanan minimum 30 hari — Pengambilan Fleksibel Gletser S3 dan S3 Glacier Deep Archive adalah solusi arsip jangka panjang. Periode durasi penyimpanan minimal adalah 90 hari untuk kelas penyimpanan S3 Glacier Flexible Retrieval dan 180 hari untuk S3 Glacier Deep Archive. Menghapus data yang diarsipkan ke kelas penyimpanan ini tidak dikenakan biaya jika objek yang Anda hapus diarsipkan selama lebih dari periode durasi penyimpanan minimal. Jika Anda menghapus atau menimpa objek yang diarsipkan dalam periode durasi minimal, Amazon S3 mengenakan biaya untuk sisa durasi tersebut.
- Biaya pengambilan data — Saat Anda memulihkan objek yang diarsipkan ke S3 Glacier Flexible Retrieval dan S3 Glacier Deep Archive, ada biaya pengambilan data per permintaan. Biaya ini bervariasi berdasarkan tingkat pengambilan yang Anda pilih saat memulai pemulihan. Untuk informasi harga, lihat Harga Amazon S3.
- Siklus Hidup S3 — Saat Anda memulihkan objek yang diarsipkan ke S3 Glacier Flexible Retrieval dan S3 Glacier Deep Archive, ada biaya pengambilan data per permintaan. Biaya ini bervariasi berdasarkan tingkat pengambilan yang Anda pilih saat memulai pemulihan. Untuk informasi harga, lihat Harga Amazon S3.

Memulihkan objek yang diarsipkan

Objek yang diarsipkan tidak dapat diakses secara real time. Anda harus memulai permintaan pemulihan terlebih dahulu, kemudian menunggu hingga salinan sementara objek tersedia selama durasi yang Anda tentukan dalam permintaan. Setelah Anda menerima salinan sementara dari objek yang dipulihkan, kelas penyimpanan objek tetaplah S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive. (Permintaan operasi A HeadObjectatau GetObjectAPI akan mengembalikan S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive sebagai kelas penyimpanan.)

> **Note**
>
> Saat memulihkan arsip, Anda membayar untuk arsip tersebut (S3 Glacier Flexible Retrieval atau tingkat S3 Glacier Deep Archive) dan salinan yang Anda pulihkan sementara (tingkat penyimpanan S3 Standard). Untuk informasi selengkapnya tentang harga, silakan lihat Harga Amazon S3.

Anda dapat memulihkan salinan objek secara terprogram atau dengan menggunakan konsol Amazon S3. Amazon S3 hanya memproses satu permintaan pemulihan pada satu waktu per objek. Untuk informasi selengkapnya, lihat Memulihkan objek yang diarsipkan.

Bagaimana kelas penyimpanan ini berbeda dari layanan S3 Glacier

Kelas penyimpanan S3 Glacier adalah bagian dari layanan Amazon S3 dan menyimpan data sebagai objek dalam ember S3. Anda dapat mengelola objek di kelas penyimpanan ini menggunakan konsol S3 atau secara terprogram menggunakan S3 atau. APIs SDKs Saat menyimpan objek di kelas penyimpanan S3 Glacier, Anda dapat menggunakan fitur S3 seperti enkripsi lanjutan, penandaan objek, dan konfigurasi Siklus Hidup S3 untuk membantu mengelola aksesibilitas dan biaya data.

> **Important**
>
> Sebaiknya gunakan kelas penyimpanan S3 Glacier dalam layanan Amazon S3 untuk semua data jangka panjang Anda.

Layanan Amazon S3 Glacier (S3 Glacier) adalah layanan terpisah yang menyimpan data sebagai arsip di dalam brankas. Layanan ini tidak mendukung fitur Amazon S3 dan tidak menyediakan dukungan konsol untuk operasi pengunggahan dan pengunduhan data. Kami tidak menyarankan menggunakan layanan S3 Glacier untuk data jangka panjang Anda. Data yang disimpan dalam layanan ini tidak dapat diakses dari layanan Amazon S3. Jika Anda mencari informasi tentang layanan S3 Glacier, lihat Panduan Pengembang Amazon S3 Glacier. Untuk mentransfer data dari layanan Amazon S3 Glacier ke kelas penyimpanan di Amazon S3 lihat Transfer Data dari Amazon S3 Glacier Vaults ke Amazon S3 di perpustakaan solusi. AWS

Bekerja dengan objek yang diarsipkan

Untuk mengurangi biaya penyimpanan objek yang jarang diakses, Anda dapat mengarsipkan objek tersebut. Ketika Anda mengarsipkan objek, objek tersebut dipindahkan ke penyimpanan berbiaya rendah, yang berarti Anda tidak dapat mengaksesnya secara real time.

Meskipun objek yang diarsipkan tidak dapat diakses secara waktu nyata, Anda dapat mengembalikannya dalam hitungan menit atau jam, tergantung pada kelas penyimpanan. Anda dapat memulihkan objek yang diarsipkan menggunakan konsol Amazon S3, Operasi Batch S3, REST API, AWS SDKs the, dan AWS Command Line Interface ().AWS CLI Untuk petunjuk, silakan lihat Memulihkan objek yang diarsipkan.

Objek Amazon S3 di kelas atau tingkatan penyimpanan berikut diarsipkan dan tidak dapat diakses secara waktu nyata:

- Kelas penyimpanan S3 Glacier Flexible Retrieval

- Kelas penyimpanan S3 Glacier Deep Archive
- Tingkat Akses Arsip Intelligent-Tiering S3
- Tingkat Akses Arsip Deep Intelligent-Tiering S3

Untuk memulihkan objek yang diarsipkan, lakukan hal berikut:

- Untuk objek di kelas penyimpanan S3 Glacier Flexible Retrieval dan S3 Glacier Deep Archive, Anda harus memulai permintaan pemulihan dan menunggu hingga salinan sementara objek tersedia. Ketika salinan sementara dari objek yang dipulihkan dibuat, kelas penyimpanan objek akan tetap sama. (A HeadObject atau GetObjectPermintaan operasi API mengembalikan S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive sebagai kelas penyimpanan.)
- Untuk objek di tingkat Akses Arsip Intelligent-Tiering S3 dan tingkat Arsip Deep Glacier Intelligent-Tiering S3, Anda harus memulai permintaan pemulihan dan menunggu hingga objek dipindahkan ke tingkat Akses Sering.

Untuk informasi selengkapnya tentang perbandingan semua kelas penyimpanan Amazon S3, lihat Memahami dan mengelola kelas penyimpanan Amazon S3. Untuk informasi selengkapnya tentang S3 Intelligent-Tiering, lihat. the section called “Cara kerja S3 Intelligent-Tiering”

Waktu yang dibutuhkan untuk menyelesaikan pekerjaan pemulihan tergantung pada kelas penyimpanan arsip atau tingkat penyimpanan yang Anda gunakan dan opsi pengambilan yang Anda tentukan: Dipercepat (hanya tersedia untuk S3 Glacier Flexible Retrieval dan Akses Arsip Intelligent-Tiering S3), Standar, atau Massal. Untuk informasi selengkapnya, lihat Memahami opsi pengambilan arsip.

Anda dapat menerima pemberitahuan saat pemulihan telah selesai dengan menggunakan Amazon S3 Event Notifications. Untuk informasi selengkapnya, lihat Notifikasi Peristiwa Amazon S3.

Memulihkan objek dari S3 Glacier

Saat Anda menggunakan S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive, Amazon S3 akan memulihkan salinan sementara objek hanya selama durasi tertentu saja. Setelah itu, layanan akan menghapus salinan objek yang dipulihkan. Anda dapat mengubah masa berlaku salinan yang dipulihkan dengan menerbitkan kembali permintaan pemulihan. Dalam hal ini, Amazon S3 akan memperbarui periode kedaluwarsa relatif terhadap waktu saat ini.

> **Note**
>
> Saat memulihkan objek yang diarsipkan dari S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive, Anda perlu membayar objek yang diarsipkan dan salinan yang Anda pulihkan sementara. Untuk informasi tentang harga, lihat Harga Amazon S3.

Amazon S3 menghitung waktu kedaluwarsa salinan objek yang dipulihkan dengan menambahkan jumlah hari yang ditentukan dalam permintaan pemulihan pada saat pemulihan yang diminta telah selesai. Amazon S3 kemudian akan membulatkan waktu yang dihasilkan ke hari berikutnya pada tengah malam Waktu Koordinat Universal (UTC). Sebagai contoh, sebuah salinan objek yang dipulihkan dibuat pada tanggal 15 Oktober 2012, pada pukul 10:30 AM UTC, dan periode pemulihan ditentukan selama 3 hari. Dalam hal ini, salinan yang dipulihkan akan kedaluwarsa pada tanggal 19 Oktober 2012, pukul 00:00 UTC, pada saat itu Amazon S3 akan menghapus salinan objek.

Memulihkan objek dari S3 Intelligent-Tiering

Saat memulihkan objek dari tingkat Akses Arsip Intelligent-Tiering S3 atau tingkat Akses Arsip Deep Intelligent-Tiering S3, objek akan kembali ke tingkat Akses Sering Intelligent-Tiering S3. Jika objek tidak diakses setelah 30 hari berturut-turut, objek akan secara otomatis berpindah ke tingkat Akses Jarang. Setelah minimal 90 hari berturut-turut tanpa akses, objek akan berpindah ke tingkat Akses Arsip Intelligent-Tiering S3. Jika objek tidak diakses setelah minimal 180 hari berturut-turut, objek akan berpindah ke tingkat Akses Arsip Dalam.

> **Note**
>
> Tidak seperti di kelas penyimpanan S3 Glacier Flexible Retrieval dan S3 Glacier Deep Archive, permintaan pemulihan untuk objek S3 Intelligent-Tiering tidak menerima nilai `Days`.

Menggunakan Operasi Batch S3 dengan permintaan pemulihan

Untuk memulihkan lebih dari satu objek Amazon S3 dengan satu permintaan, Anda dapat menggunakan Operasi Batch S3. Anda menyediakan daftar objek yang akan dioperasikan kepada Operasi Batch S3. Operasi Batch S3 akan memanggil masing-masing operasi API untuk melakukan operasi tertentu. Satu tugas Operasi Batch dapat melakukan operasi tertentu pada miliaran objek yang berisi data sebesar eksabita.

Topik

- Memahami opsi pengambilan arsip
- Memulihkan objek yang diarsipkan

Memahami opsi pengambilan arsip

Berikut ini adalah opsi pengambilan yang tersedia saat memulihkan objek yang diarsipkan dalam Amazon S3:

- Dipercepat-Akses data Anda dengan cepat melalui penyimpanan kelas S3 Glacier Flexible Retrieval atau Akses Arsip Intelligent-Tiering S3. Anda dapat menggunakan opsi ini saat ada permintaan mendesak untuk subset arsip. Untuk semua objek arsip kecuali yang paling besar (250 MB+), data yang diakses dengan menggunakan pengambilan yang dipercepat biasanya akan tersedia dalam waktu 1-5 menit.

  > **Note**
  >
  > Pengambilan yang dipercepat adalah fitur premium dan dibebankan pada tingkat permintaan dan pengambilan yang dipercepat.
  > Untuk informasi tentang harga Amazon S3, lihat Harga Amazon S3.

  Kapasitas yang disediakan membantu memastikan bahwa kapasitas pengambilan untuk proses pengambilan yang dipercepat dari S3 Glacier Flexible Retrieval selalu tersedia ketika Anda memerlukannya. Untuk informasi selengkapnya, lihat Kapasitas yang disediakan.
- Standar—Akses semua objek yang diarsipkan dalam beberapa jam. Standar adalah opsi default untuk permintaan pengambilan yang tidak menentukan opsi pengambilan. Pengambilan standar biasanya selesai dalam 3–5 jam untuk objek yang disimpan di kelas penyimpanan S3 Glacier Flexible Retrieval atau tingkat Akses Arsip Intelligent-Tiering S3. Pengambilan ini biasanya selesai dalam waktu 12 jam untuk objek yang disimpan di kelas penyimpanan S3 Glacier Deep Archive atau tingkat Akses Arsip Intelligent-Tiering S3. Pengambilan standar gratis untuk objek yang disimpan di S3 Intelligent-Tiering.

  > **Note**
  >
  > - Untuk objek yang disimpan di kelas penyimpanan S3 Glacier Flexible Retrieval atau tingkat S3 Intelligent-Tiering Archive Access, pengambilan Standar yang dimulai dengan

menggunakan operasi pemulihan Operasi Batch S3 biasanya dimulai dalam beberapa menit dan selesai dalam 3-5 jam.

- Untuk objek di kelas penyimpanan S3 Glacier Deep Archive atau tingkat S3 Intelligent-Tiering Deep Archive Access, pengambilan Standar yang dimulai dengan menggunakan operasi pemulihan Operasi Batch biasanya dimulai dalam waktu 9 jam dan selesai dalam 12 jam.

- Massal—Akses data Anda dengan opsi pengambilan biaya terendah di Amazon S3 Glacier. Dengan pengambilan massal, Anda dapat mengambil data dalam jumlah besar, bahkan yang berukuran petabyte, dengan biaya yang murah.

  Untuk objek yang disimpan di kelas penyimpanan S3 Glacier Flexible Retrieval atau tingkat S3 Intelligent-Tiering Archive Access, pengambilan massal biasanya selesai dalam waktu 5—12 jam. Untuk objek yang disimpan di kelas penyimpanan S3 Glacier Deep Archive atau tingkat S3 Intelligent-Tiering Deep Archive Access, pengambilan ini biasanya selesai dalam waktu 48 jam.

  Pengambilan massal gratis untuk objek yang disimpan di kelas penyimpanan S3 Glacier Flexible Retrieval atau S3 Intelligent-Tiering.

Tabel berikut merangkum opsi pengambilan arsip. Untuk informasi tentang harga, lihat Harga Amazon S3.

Untuk membuat`Expedited`,`Standard`, atau `Bulk` pengambilan, atur elemen `Tier` permintaan di RestoreObjectPermintaan operasi REST API ke opsi yang Anda inginkan, atau yang setara di AWS Command Line Interface (AWS CLI) atau AWS SDKs. Jika Anda sudah membeli kapasitas yang disediakan, semua pengambilan akan secara otomatis dilayani melalui kapasitas tersebut.

Kapasitas yang disediakan

Kapasitas yang disediakan membantu memastikan bahwa kapasitas pengambilan Anda untuk pengambilan Dipercepat dari S3 Glacier Flexible Retrieval tersedia saat Anda membutuhkannya. Setiap unit kapasitas menyediakan bahwa setidaknya tiga Expedited retrievals dapat dilakukan setiap 5 menit, dan menyediakan hingga 150 megabyte per detik () throughput pengambilan. MBps

Jika beban kerja Anda membutuhkan akses yang sangat andal dan dapat diprediksi ke sebuah subset data dalam hitungan menit, pertimbangkan untuk membeli kapasitas pengambilan yang disediakan. Tanpa kapasitas yang disediakan, pengambilan Dipercepat mungkin tidak akan diterima jika periode permintaan sedang tinggi. Jika Anda memerlukan akses ke pengambilan Dipercepat dalam keadaan apa pun, sebaiknya beli kapasitas pengambilan yang disediakan.

Unit kapasitas yang disediakan dialokasikan ke a. Akun AWS Dengan demikian, pemohon pengambilan data Dipercepat harus membeli unit kapasitas yang disediakan, bukan pemilik bucket.

Anda dapat membeli kapasitas yang disediakan dengan menggunakan konsol Amazon S3, konsol Amazon S3 Glacier, operasi Purchase Provisioned Capacity REST API, atau. AWS SDKs AWS CLI Untuk informasi harga kapasitas yang disediakan, lihat harga Amazon S3.

Tingkat permintaan inisiasi pemulihan S3 Glacier

Ketika Anda menginisiasi permintaan pemulihan untuk objek yang disimpan di kelas penyimpanan S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive, kuota permintaan pemulihan akan diterapkan untuk file Akun AWS. S3 Glacier mendukung permintaan pemulihan dengan kecepatan 1.000 transaksi per detik. Jika kecepatan ini terlampaui, maka permintaan yang valid akan dibatasi atau ditolak dan Amazon S3 akan mengembalikan kesalahan `ThrottlingException`.

Atau, Anda juga bisa menggunakan Operasi Batch S3 untuk mengambil sejumlah besar objek yang disimpan di S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive dengan satu permintaan. Untuk informasi selengkapnya, lihat Melakukan operasi objek secara massal dengan Operasi Batch.

Memulihkan objek yang diarsipkan

Objek Amazon S3 di kelas atau tingkatan penyimpanan berikut diarsipkan dan tidak dapat diakses secara waktu nyata:

- Kelas penyimpanan S3 Glacier Flexible Retrieval
- Kelas penyimpanan S3 Glacier Deep Archive
- Tingkat Akses Arsip Intelligent-Tiering S3
- Tingkat Akses Arsip Deep Intelligent-Tiering S3

Objek Amazon S3 yang disimpan di kelas penyimpanan S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive tidak langsung dapat diakses. Untuk mengakses objek dalam kelas penyimpanan ini, Anda harus mengembalikan salinan sementara objek ke bucket S3 selama durasi tertentu (jumlah hari). Jika Anda menginginkan salinan permanen dari objek tersebut, pulihkan objek, lalu buat salinannya di bucket Amazon S3 Anda. Menyalin objek yang dipulihkan tidak didukung di konsol Amazon S3. Untuk jenis operasi penyalinan ini, gunakan AWS Command Line Interface (AWS CLI), the AWS SDKs, atau REST API. Kecuali jika Anda membuat salinan dan mengubah kelas penyimpanannya, objek akan tetap disimpan di kelas penyimpanan S3 Glacier Flexible Retrieval atau

S3 Glacier Deep Archive. Untuk informasi tentang menggunakan kelas penyimpanan, ini lihat Kelas penyimpanan untuk objek yang jarang diakses.

Untuk mengakses objek di tingkat Akses Arsip Intelligent-Tiering S3 dan Deep Archive Access, Anda harus memulai permintaan pemulihan dan menunggu hingga objek dipindahkan ke tingkat Akses Sering. Saat memulihkan objek dari tingkat Akses Arsip atau tingkat Akses Arsip Dalam, objek akan kembali ke tingkat Akses Sering. Untuk informasi tentang menggunakan kelas penyimpanan, ini lihatKelas penyimpanan untuk mengoptimalkan data secara otomatis dengan pola akses yang berubah atau tidak diketahui.

Untuk informasi umum tentang objek yang diarsipkan, lihat Bekerja dengan objek yang diarsipkan.

> **Note**
>
> - Saat memulihkan objek yang diarsipkan dari kelas penyimpanan S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive, Anda membayar objek yang diarsipkan dan salinan yang dipulihkan sementara.
> - Saat Anda memulihkan objek dari S3 Intelligent-Tiering, tidak ada biaya pengambilan untuk pengambilan Standar atau Massal.
> - Permintaan pemulihan berikutnya yang dipanggil pada objek yang diarsipkan yang telah dipulihkan ditagih sebagai `GET` permintaan. Untuk informasi tentang harga, lihat Harga Amazon S3.

Memulihkan objek yang diarsipkan

Anda dapat memulihkan objek yang diarsipkan menggunakan konsol Amazon S3, Amazon S3 REST API, AWS SDKs Operasi Batch AWS CLI(), AWS Command Line Interface atau S3.

Menggunakan konsol S3

Pemulihan objek menggunakan konsol Amazon S3

Gunakan prosedur berikut untuk memulihkan objek yang telah diarsipkan ke kelas penyimpanan S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive Access, atau tingkat penyimpanan Akses Arsip Intelligent-Tiering S3 atau Akses Arsip Dalam.

Untuk memulihkan objek yang diarsipkan

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Dalam daftar bucket, pilih nama bucket yang berisi objek yang ingin Anda pulihkan.
4. Di daftar Objek, pilih objek yang ingin Anda pulihkan, pilih Tindakan, lalu pilih Memulai pemulihan.
5. Jika Anda memulihkan dari S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive, masukkan jumlah hari yang Anda inginkan agar data yang diarsipkan dapat diakses di kotak Jumlah hari di mana salinan yang telah dipulihkan tetap tersedia.
6. Untuk Tingkat pengambilan, lakukan salah satu hal berikut:

   - Pilih Pengambilan massal atau Pengambilan standar, lalu pilih Memulai pemulihan.
   - Pilih Pengambilan dipercepat (hanya tersedia untuk S3 Glacier Flexible Retrieval atau Akses Arsip Intelligent-Tiering S3). Jika Anda memulihkan objek di S3 Glacier Flexible Retrieval, Anda dapat memilih apakah akan membeli kapasitas yang disediakan untuk pengambilan Dipercepat. Jika Anda ingin membeli kapasitas yang disediakan, lanjutkan ke langkah berikutnya. Jika tidak, pilih Memulai pemulihan.

   > **Note**
   >
   > Objek dari tingkat S3 Intelligent-Tiering Archive Access dan Deep Archive Access secara otomatis dikembalikan ke tingkat Frequent Access.

7. (Opsional) Jika Anda memulihkan objek di S3 Glacier Flexible Retrieval dan memilih Pengambilan yang dipercepat, Anda dapat memilih apakah akan membeli kapasitas yang disediakan. Kapasitas yang disediakan hanya tersedia untuk objek di S3 Glacier Flexible Retrieval. Jika Anda memiliki kapasitas yang disediakan, pilih Memulai pemulihan untuk memulai pengambilan yang disediakan.

   Jika Anda memiliki kapasitas yang disediakan, semua pengambilan yang Dipercepat dilayani oleh kapasitas yang disediakan. Untuk informasi selengkapnya, lihat Kapasitas yang disediakan.

   - Jika Anda tidak memiliki kapasitas yang disediakan dan tidak ingin membelinya, pilih Memulai pemulihan.

- Jika Anda tidak memiliki kapasitas yang disediakan, tetapi Anda ingin membeli unit kapasitas yang disediakan (PCUs), pilih Purchase. PCUs Di kotak PCUs dialog Pembelian, pilih berapa banyak yang ingin PCUs Anda beli, konfirmasi pembelian Anda, lalu pilih Pembelian PCUs. Saat Anda mendapatkan pesan Pembelian berhasil, pilih Memulai pemulihan untuk memulai pengambilan yang disediakan.

Menggunakan AWS CLI

Memulihkan objek dari S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive

Contoh berikut menggunakan perintah `restore-object` untuk mengembalikan objek *`dir1/example.obj`* dalam bucket *`amzn-s3-demo-bucket`* selama 25 hari.

```
aws s3api restore-object --bucket amzn-s3-demo-bucket --key dir1/example.obj --restore-request '{"Days":25,"GlacierJobParameters":{"Tier":"Standard"}}'
```

Jika sintaks JSON digunakan dalam contoh hasil dalam kesalahan pada klien Windows, gantikan permintaan pemulihan dengan sintaks berikut:

```
--restore-request Days=25,GlacierJobParameters={"Tier"="Standard"}
```

Memulihkan objek dari S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive

Contoh berikut menggunakan perintah `restore-object` untuk mengembalikan objek *`dir1/example.obj`* dalam bucket *`amzn-s3-demo-bucket`* ke tingkat Akses Sering.

```
aws s3api restore-object --bucket amzn-s3-demo-bucket --key dir1/example.obj --restore-request '{}'
```

> **Note**
>
> Tidak seperti di kelas penyimpanan S3 Glacier Flexible Retrieval dan S3 Glacier Deep Archive, permintaan pemulihan untuk objek S3 Intelligent-Tiering tidak menerima nilai `Days`.

Memantau status pemulihan

Untuk memantau status permintaan `restore-object` Anda, gunakan perintah `head-object` berikut:

```
aws s3api head-object --bucket amzn-s3-demo-bucket --key dir1/example.obj
```

Untuk informasi selengkapnya, silakan lihat restore-objectdalam AWS CLI Command Reference.

Penggunaan API REST

Amazon S3 menyediakan operasi API bagi Anda untuk memulai pemulihan objek yang diarsipkan. Untuk informasi selengkapnya, silakan lihat RestoreObject dalam Referensi API Layanan Penyimpanan Sederhana Amazon.

Menggunakan AWS SDKs

Untuk contoh cara memulihkan objek yang diarsipkan di S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive dengan AWS SDKs, lihat Contoh kode di Referensi API Amazon S3.

Menggunakan Operasi Batch S3

Untuk memulihkan lebih dari satu objek yang diarsipkan dengan satu permintaan, Anda dapat menggunakan Operasi Batch S3. Anda menyediakan daftar objek yang akan dioperasikan kepada Operasi Batch S3. Operasi Batch S3 akan memanggil masing-masing operasi API untuk melakukan operasi tertentu. Satu tugas Operasi Batch dapat melakukan operasi tertentu pada miliaran objek yang berisi data sebesar eksabita.

Untuk membuat tugas Operasi Batch, Anda harus memiliki manifes yang hanya berisi objek yang ingin dipulihkan. Anda dapat membuat manifes dengan menggunakan Inventaris S3, atau Anda dapat memasok file CSV dengan informasi yang diperlukan. Untuk informasi selengkapnya, lihat the section called “Menentukan manifes”.

Sebelum membuat dan menjalankan tugas Operasi Batch S3, Anda harus memberikan izin ke Amazon S3 untuk menjalankan Operasi Batch S3 atas nama Anda. Untuk izin yang diperlukan, lihat the section called “Memberikan izin”.

> **Note**
>
> Tugas Operasi Batch dapat beroperasi baik pada objek kelas penyimpanan S3 Glacier Flexible Retrieval dan S3 Glacier Deep Archive atau di objek tingkatan penyimpanan S3 Intelligent-Tiering Archive Access dan Deep Archive Access. Operasi Batch tidak dapat beroperasi pada kedua jenis objek yang diarsipkan dalam tugas yang sama. Untuk memulihkan kedua jenis objek tersebut, Anda harus membuat pekerjaan Operasi Batch terpisah.

Untuk informasi selengkapnya tentang menggunakan Operasi Batch untuk memulihkan objek arsip, lihat the section called “Memulihkan objek”.

Untuk membuat tugas Memulai Operasi Batch Pemulihan Objek S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Operasi Batch.
3. Pilih Buat tugas.
4. Untuk Wilayah AWS, pilih Wilayah tempat Anda ingin membuat tugas Anda.
5. Pada Format manifes, pilih jenis manifes yang akan digunakan.
   - Jika Anda memilih Laporan inventaris S3, masukkan jalur ke objek `manifest.json` yang dihasilkan Amazon S3 sebagai bagian dari laporan inventaris yang diformat CSV. Jika Anda ingin menggunakan versi manifes selain versi terbaru, masukkan ID versi untuk objek `manifest.json`.
   - Jika Anda memilih CSV, masukkan jalur ke objek manifes yang diformat CSV. Objek manifes harus mengikuti format yang dijelaskan di konsol. Jika Anda ingin menggunakan versi selain yang terbaru, Anda dapat secara opsional memasukkan ID versi untuk objek manifes.
6. Pilih Selanjutnya.
7. Di bagian Operasi, pilih Pulihkan.
8. Di bagian Pulihkan, untuk Pulihkan sumber, pilih Glacier Flexible Retrieval atau Glacier Deep Archive atau tingkat Akses Arsip Intelligent-Tiering atau tingkat Akses Arsip Dalam.

   Jika Anda memilih Glacier Flexible Retrieval atau Glacier Deep Archive, masukkan jumlah Jumlah hari tersedianya salinan yang dipulihkan.

   Untuk Tingkat pengambilan, pilih tingkat yang ingin Anda gunakan.
9. Pilih Selanjutnya.
10. Di halaman Konfigurasi opsi tambahan, isilah bagian berikut:
    - Di bagian Opsi tambahan, berikan deskripsi untuk tugas tersebut dan tentukan nomor prioritas untuk tugas tersebut. Angka yang lebih tinggi menunjukkan prioritas yang lebih tinggi. Untuk informasi selengkapnya, lihat the section called “Menetapkan prioritas tugas”.

- Di bagian Laporan penyelesaian, pilih apakah Operasi Batch harus membuat laporan penyelesaian. Untuk informasi selengkapnya tentang laporan penyelesaian, lihat the section called “Laporan penyelesaian”.
- Di bagian Izin, Anda harus memberikan izin ke Amazon S3 untuk melakukan Operasi Batch atas nama Anda. Untuk izin yang diperlukan, lihat the section called “Memberikan izin”.
- (Opsional) Di bagian Tanda tugas, tambahkan tanda dalam pasangan nilai kunci. Untuk informasi selengkapnya, lihat the section called “Menggunakan tanda”.

Setelah selesai, pilih Berikutnya.

11. Pada halaman Tinjau, verifikasi pengaturan. Jika Anda perlu membuat perubahan, pilih Sebelumnya. Atau, pilih Buat tugas.

Untuk informasi selengkapnya tentang Operasi Batch, lihat Memulihkan objek dengan Operasi Batch dan Membuat pekerjaan Operasi Batch S3.

Memeriksa status pemulihan dan tanggal kedaluwarsa

Anda dapat memeriksa status permintaan pemulihan atau tanggal kedaluwarsa dengan menggunakan konsol Amazon S3, Pemberitahuan Acara Amazon S3, API REST Amazon S3, AWS CLI atau Amazon S3 REST API.

> **Note**
>
> Objek yang dipulihkan dari kelas penyimpanan S3 Glacier Flexible Retrieval dan S3 Glacier Deep Archive disimpan hanya untuk jumlah hari yang Anda tentukan. Prosedur berikut mengembalikan tanggal kedaluwarsa untuk salinan ini.
> Objek yang dipulihkan dari S3 Intelligent-Tiering Archive Access dan tingkat penyimpanan Deep Archive Access tidak memiliki tanggal kedaluwarsa dan sebagai gantinya dipindahkan kembali ke tingkat Frequent Access.

Menggunakan konsol S3

Untuk memeriksa status pemulihan dan tanggal kedaluwarsa sebuah objek di konsol Amazon S3

1. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.

3. Dalam daftar bucket, pilih nama bucket yang berisi objek yang sedang Anda pulihkan.
4. Dalam daftar Objek, pilih objek yang sedang Anda pulihkan. Halaman detail objek akan muncul.

   - Jika pemulihan belum selesai, di bagian atas halaman, Anda melihat bagian yang mengatakan Pemulihan sedang berlangsung.
   - Jika pemulihan selesai, di bagian atas halaman, Anda akan melihat bagian yang mengatakan Pemulihan selesai. Jika Anda memulihkan dari S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive, bagian ini juga menampilkan Tanggal kedaluwarsa pemulihan. Amazon S3 akan menghapus salinan yang dipulihkan dari objek yang diarsipkan pada tanggal ini.

Menggunakan Pemberitahuan Acara Amazon S3

Anda dapat diberi tahu tentang penyelesaian restorasi objek dengan menggunakan `s3:ObjectRestore:Completed` tindakan dengan fitur Pemberitahuan Acara Amazon S3. Untuk informasi selengkapnya tentang mengaktifkan notifikasi peristiwa, lihat Mengaktifkan notifikasi menggunakan Amazon SQS, Amazon SNS, dan. AWS Lambda Untuk informasi selengkapnya tentang berbagai jenis `ObjectRestore` acara, lihatthe section called “Jenis event yang didukung untuk SQS, SNS, dan Lambda”.

Menggunakan AWS CLI

Periksa status pemulihan objek dan tanggal kedaluwarsa dengan AWS CLI

Contoh berikut menggunakan perintah `head-object` untuk melihat metadata untuk objek *`dir1/example.obj`* di bucket *`amzn-s3-demo-bucket`*. Ketika Anda menjalankan perintah ini pada objek yang sedang dipulihkan, Amazon S3 akan menampilkan informasi jika pemulihan sedang berlangsung dan (jika ada) tanggal kedaluwarsanya.

```
aws s3api head-object --bucket amzn-s3-demo-bucket --key dir1/example.obj
```

Output yang diharapkan (pemulihan sedang berlangsung):

```
{
    "Restore": "ongoing-request=\"true\"",
    "LastModified": "2020-06-16T21:55:22+00:00",
    "ContentLength": 405,
    "ETag": "\"b662d79adeb7c8d787ea7eafb9ef6207\"",
    "VersionId": "wbYaE2vtOV0iIBXrOqGAJt3fP1cHB8Wi",
    "ContentType": "binary/octet-stream",
    "ServerSideEncryption": "AES256",
```

```
    "Metadata": {},
    "StorageClass": "GLACIER"
}
```

Output yang diharapkan (pemulihan selesai):

```
{
    "Restore": "ongoing-request=\"false\", expiry-date=\"Wed, 12 Aug 2020 00:00:00 GMT
\"",
    "LastModified": "2020-06-16T21:55:22+00:00",
    "ContentLength": 405,
    "ETag": "\"b662d79adeb7c8d787ea7eafb9ef6207\"",
    "VersionId": "wbYaE2vtOV0iIBXrOqGAJt3fP1cHB8Wi",
    "ContentType": "binary/octet-stream",
    "ServerSideEncryption": "AES256",
    "Metadata": {},
    "StorageClass": "GLACIER"
}
```

Untuk informasi selengkapnya tentang `head-object`, lihat head-objectdalam AWS CLI Command Reference.

Penggunaan API REST

Amazon S3 menyediakan operasi API bagi Anda untuk mengambil metadata objek. Untuk memeriksa status pemulihan dan tanggal kedaluwarsa objek yang diarsipkan menggunakan REST API, lihat HeadObject dalam Referensi API Layanan Penyimpanan Sederhana Amazon.

Meningkatkan kecepatan pemulihan yang sedang berlangsung

Anda dapat meningkatkan kecepatan pemulihan saat sedang dalam proses.

Untuk meningkatkan pemulihan yang sedang dalam proses ke tingkat yang lebih cepat

1. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Di daftar Bucket pilih nama bucket yang berisi objek yang ingin dipulihkan.
4. Dalam daftar Objek, pilih objek yang sedang Anda pulihkan. Halaman detail objek akan muncul. Pada halaman detail objek, pilih Tingkatkan tingkat pengambilan. Untuk informasi tentang pemeriksaan status pemulihan objek, lihat Memeriksa status pemulihan dan tanggal kedaluwarsa.

5. Pilih tingkat yang ingin Anda tingkatkan, lalu pilih Memulai pemulihan.

Mengelola siklus hidup objek

Siklus Hidup S3 membantu Anda menyimpan objek secara efektif sepanjang siklus hidupnya dengan mentransisisinya ke kelas penyimpanan berbiaya lebih rendah, atau, menghapus objek kedaluwarsa atas nama Anda. Untuk mengelola siklus hidup objek, buat konfigurasi Siklus Hidup S3 untuk bucket Anda. Konfigurasi Siklus Hidup S3 adalah serangkaian aturan yang menetapkan tindakan yang diterapkan Amazon S3 pada sekelompok objek. Ada dua jenis tindakan:

- Tindakan transisi–Tindakan ini menentukan kapan objek bertransisi ke kelas penyimpanan lainnya. Misalnya, Anda dapat memilih untuk melakukan transisi objek ke kelas penyimpanan S3 Standard-IA dalam waktu 30 hari setelah membuatnya, atau mengarsipkan objek ke kelas penyimpanan S3 Glacier Flexible Retrieval satu tahun setelah membuatnya. Untuk informasi selengkapnya, lihat Memahami dan mengelola kelas penyimpanan Amazon S3.

  Ada biaya yang terkait dengan permintaan transisi siklus hidup. Untuk informasi harga, lihat Harga Amazon S3.
- Tindakan kedaluwarsa–Tindakan ini menentukan kapan objek kedaluwarsa. Amazon S3 menghapus objek kedaluwarsa atas nama Anda. Misalnya, Anda mungkin memilih untuk kedaluwarsa objek setelah disimpan selama periode kepatuhan peraturan. Untuk informasi selengkapnya, lihat Mengakhiri objek.

  Ada potensi biaya yang terkait dengan kedaluwarsa siklus hidup hanya ketika Anda kedaluwarsa objek di kelas penyimpanan dengan durasi penyimpanan minimum. Untuk informasi selengkapnya, lihat Biaya durasi penyimpanan minimum.

> **Important**
>
> Bucket tujuan umum — Anda tidak dapat menggunakan kebijakan bucket untuk mencegah penghapusan atau transisi oleh aturan Siklus Hidup S3. Misalnya, meskipun kebijakan bucket Anda menolak semua tindakan untuk semua prinsipal, konfigurasi Siklus Hidup S3 Anda tetap berfungsi seperti biasa.

Objek yang ada dan baru

Ketika Anda menambahkan konfigurasi Siklus Hidup ke dalam bucket, aturan konfigurasi berlaku untuk objek yang ada dan objek yang Anda tambahkan kemudian. Misalnya, jika Anda menambahkan aturan konfigurasi Siklus Hidup hari ini dengan tindakan kedaluwarsa yang menyebabkan objek kedaluwarsa 30 hari setelah pembuatan, Amazon S3 akan mengantri untuk menghapus objek yang ada yang berumur lebih dari 30 hari.

Perubahan dalam penagihan

Jika ada penundaan antara saat objek memenuhi syarat untuk tindakan siklus hidup dan saat Amazon S3 mentransfer atau mengakhiri objek Anda, perubahan penagihan akan diterapkan segera setelah objek memenuhi syarat untuk tindakan siklus hidup. Misalnya, jika objek dijadwalkan kedaluwarsa dan Amazon S3 tidak segera kedaluwarsa objek, Anda tidak akan dikenakan biaya penyimpanan setelah waktu kedaluwarsa.

Satu pengecualian untuk perilaku ini adalah jika Anda memiliki aturan siklus hidup untuk bertransisi ke kelas penyimpanan S3 Intelligent-Tiering. Dalam hal ini, perubahan penagihan tidak terjadi hingga objek telah ditransisikan ke S3 Intelligent-Tiering. Untuk informasi selengkapnya tentang aturan Siklus Hidup S3, lihat Elemen konfigurasi Siklus Hidup.

> **Note**
>
> Tidak ada biaya pengambilan data untuk transisi siklus hidup. Namun, ada biaya konsumsi per permintaan saat menggunakan`PUT`,`COPY`, atau aturan siklus hidup untuk memindahkan data ke kelas penyimpanan S3 apa pun. Pertimbangkan biaya konsumsi atau transisi sebelum memindahkan objek ke kelas penyimpanan apa pun. Untuk informasi selengkapnya tentang pertimbangan biaya, lihat Harga Amazon S3.

Memantau pengaruh aturan siklus hidup

Untuk memantau efek pembaruan yang dibuat oleh aturan siklus hidup aktif, lihat. the section called “Bagaimana cara memantau tindakan yang diambil oleh aturan siklus hidup saya?”

Mengelola siklus hidup objek yang lengkap

Dengan aturan konfigurasi Siklus Hidup S3, Anda dapat memberi tahu Amazon S3 untuk mengalihkan objek ke kelas penyimpanan yang lebih murah, mengarsipkan atau menghapusnya. Misalnya:

- Jika Anda mengunggah log berkala ke bucket, aplikasi Anda mungkin memerlukannya selama seminggu atau sebulan. Setelah itu, Anda mungkin ingin menghapusnya.
- Beberapa dokumen sering diakses untuk periode waktu terbatas. Setelah itu, dokumen tersebut jarang diakses. Sewaktu-waktu, Anda mungkin tidak memerlukan akses waktu nyata, tetapi organisasi atau peraturan Anda mungkin mengharuskan Anda untuk mengarsipkannya selama periode tertentu. Setelah itu, Anda dapat menghapusnya.
- Anda dapat mengunggah beberapa jenis data ke Amazon S3 terutama untuk tujuan pengarsipan. Misalnya, Anda dapat mengarsipkan media digital, catatan keuangan, dan perawatan kesehatan, data urutan genomik mentah, cadangan basis data jangka panjang, dan data yang harus disimpan untuk kepatuhan terhadap peraturan.

Dengan menggabungkan tindakan Siklus Hidup S3 untuk mengelola siklus hidup lengkap objek. Misalnya, bayangkan objek yang Anda buat memiliki siklus hidup yang didefinisikan dengan baik. Pada awalnya, objek sering diakses selama periode 30 hari. Kemudian, objek jarang diakses hingga 90 hari. Setelah itu, objek tidak lagi diperlukan, jadi Anda dapat memilih untuk mengarsipkan atau menghapusnya.

Dalam skenario ini, Anda dapat membuat aturan Siklus Hidup S3, yaitu Anda menentukan tindakan transisi awal ke penyimpanan S3 Intelligent-Tiering, S3 Standard-IA, atau S3 One Zone-IA, tindakan transisi lain ke penyimpanan S3 Glacier Flexible Retrieval untuk pengarsipan, dan tindakan kedaluwarsa. Saat Anda memindahkan objek dari satu kelas penyimpanan ke kelas penyimpanan lainnya, Anda menghemat biaya penyimpanan. Untuk informasi selengkapnya tentang pertimbangan biaya, lihat Harga Amazon S3.

Topik

- Transisi objek menggunakan Siklus Hidup Amazon S3
- Mengakhiri objek
- Menyetel konfigurasi Siklus Hidup S3 pada bucket
- Bagaimana Siklus Hidup S3 berinteraksi dengan konfigurasi bucket lainnya
- Mengonfigurasi pemberitahuan acara Siklus Hidup S3
- Elemen konfigurasi Siklus Hidup
- Bagaimana Amazon S3 menangani konflik dalam konfigurasi siklus hidup
- Contoh konfigurasi Siklus Hidup S3
- Memecahkan masalah Siklus Hidup Amazon S3

Transisi objek menggunakan Siklus Hidup Amazon S3

Anda dapat menambahkan tindakan transisi ke konfigurasi Siklus Hidup S3 untuk memberi tahu Amazon S3 agar memindahkan objek ke kelas penyimpanan Amazon S3 lainnya. Untuk informasi selengkapnya tentang kelas penyimpanan, lihat Memahami dan mengelola kelas penyimpanan Amazon S3. Beberapa contoh kapan Anda dapat menggunakan konfigurasi Siklus Hidup S3 dengan cara ini termasuk berikut ini:

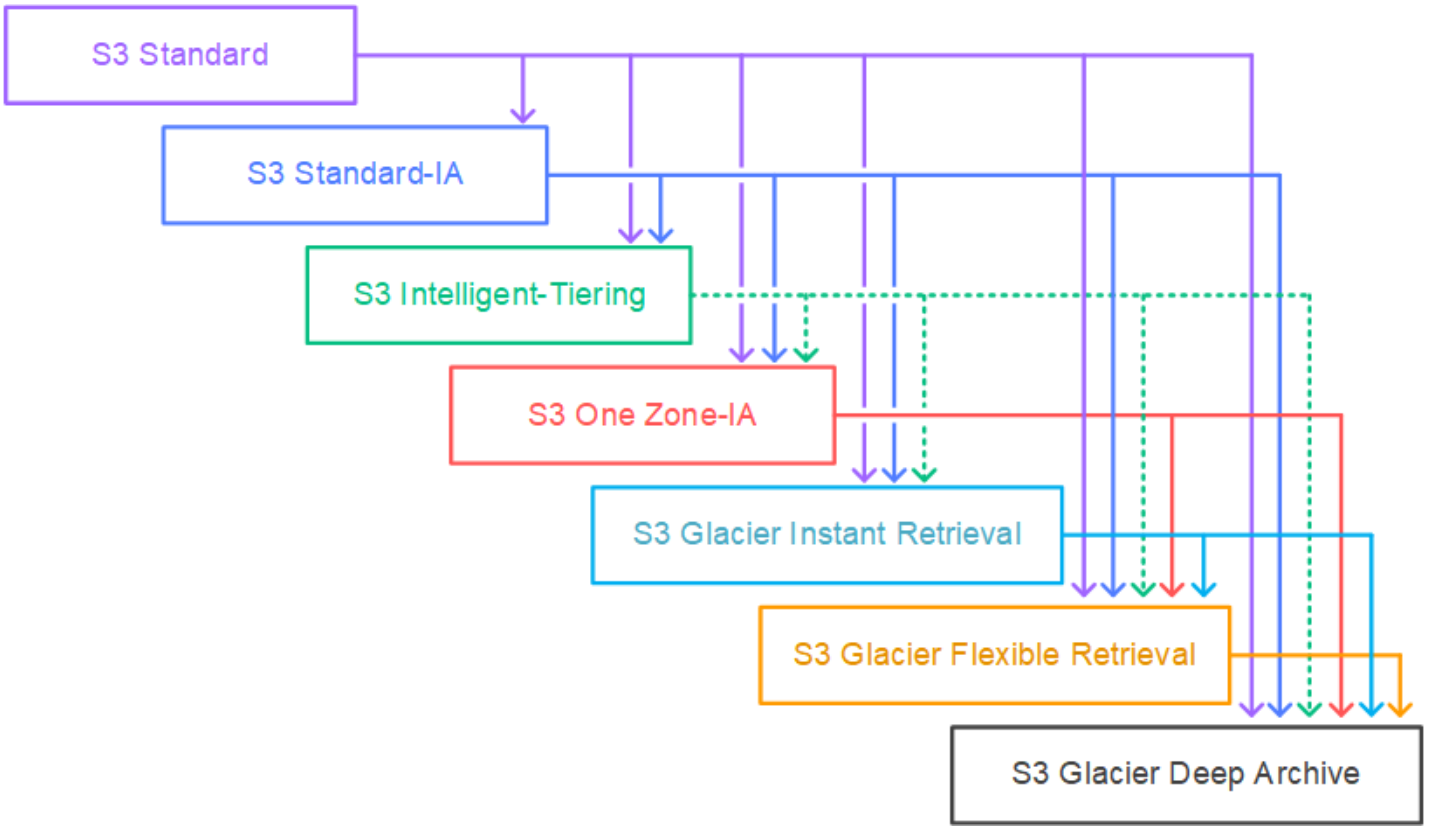
- Ketika Anda mengetahui bahwa objek jarang diakses, Anda dapat mengalihkannya ke kelas penyimpanan S3 Standard-IA.
- Anda mungkin ingin mengarsipkan objek yang tidak perlu Anda akses secara real time ke kelas penyimpanan S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive.

> **Note**
>
> Objek terenkripsi tetap dienkripsi selama proses transisi kelas penyimpanan.

Transisi yang didukung

Dalam konfigurasi Siklus Hidup S3, Anda dapat menentukan aturan untuk transisi objek dari satu kelas penyimpanan ke kelas penyimpanan lainnya guna menghemat biaya penyimpanan. Saat Anda tidak mengetahui pola akses objek Anda, atau pola akses Anda berubah seiring waktu berjalan, Anda dapat melakukan transisi objek ke kelas penyimpanan S3 Intelligent-Tiering untuk penghematan biaya otomatis. Untuk informasi tentang kelas penyimpanan, lihat Memahami dan mengelola kelas penyimpanan Amazon S3.

Amazon S3 mendukung model air terjun untuk transisi antar kelas penyimpanan, seperti yang ditunjukkan pada diagram berikut.



Transisi siklus hidup yang didukung

Amazon S3 mendukung transisi siklus hidup antara kelas penyimpanan berikut menggunakan konfigurasi Siklus Hidup S3.

- Kelas penyimpanan Standar S3 untuk kelas penyimpanan IA Standar S3-IA, S3 Intelligent-Tiering, S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval, atau S3 Glacier Deep Archive kelas penyimpanan.
- Kelas penyimpanan S3 Standard-IA ke kelas penyimpanan S3 Intelligent-Tiering, S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval, atau S3 Glacier Deep Archive.
- Kelas penyimpanan S3 Intelligent-Tiering dapat bertransisi ke kelas penyimpanan yang berbeda tergantung pada tingkat akses S3 Intelligent-Tiering. Transisi berikut dimungkinkan untuk setiap tingkat akses.
  - Tingkat Akses Sering atau tingkat Akses Jarang ke kelas penyimpanan IA S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval, atau S3 Glacier Deep Archive.
  - Arsipkan tingkat Akses Instan ke S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval, atau kelas penyimpanan S3 Glacier Deep Archive.

  - Arsipkan Tingkat Akses ke S3 Glacier Flexible Retrieval, atau kelas penyimpanan S3 Glacier Deep Archive.
  - Tingkat akses Deep Archive ke kelas penyimpanan S3 Glacier Deep Archive.
- Kelas penyimpanan S3 One Zone-IA ke kelas penyimpanan S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive.
- Kelas penyimpanan S3 Glacier Instant Retrieval ke kelas penyimpanan S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive.
- Kelas penyimpanan S3 Glacier Flexible Retrieval ke kelas penyimpanan S3 Glacier Deep Archive.

> **Note**
>
> Untuk mengaktifkan versi atau membuat versi bucket yang ditangguhkan, Anda tidak dapat mentransisikan objek dengan status replikasi. `Pending`

Kendala dan pertimbangan untuk transisi

Transisi kelas penyimpanan siklus hidup memiliki kendala berikut:

Objek yang lebih kecil dari 128 KB tidak akan bertransisi secara default ke kelas penyimpanan apa pun

Amazon S3 menerapkan perilaku default ke konfigurasi Siklus Hidup S3 yang mencegah objek yang lebih kecil dari 128 KB dialihkan ke kelas penyimpanan apa pun. Kami tidak menyarankan transisi objek kurang dari 128 KB karena Anda dikenakan biaya permintaan transisi untuk setiap objek. Ini berarti, untuk objek yang lebih kecil, biaya transisi dapat lebih besar daripada penghematan penyimpanan. Untuk informasi selengkapnya tentang biaya permintaan transisi, lihat Permintaan & pengambilan data di tab Penyimpanan & permintaan di halaman harga Amazon S3.

Untuk mengizinkan objek yang lebih kecil bertransisi, Anda dapat menambahkan filter ukuran objek ke aturan transisi Siklus Hidup yang menentukan ukuran minimum kustom (`0bjectSizeGreaterThan`) atau ukuran maksimum (). `0bjectSizeLessThan` Untuk informasi selengkapnya, lihat Contoh: Mengizinkan objek yang lebih kecil dari 128 KB untuk dialihkan.

> **Note**
>
> Pada bulan September 2024 Amazon S3 memperbarui perilaku transisi default untuk objek kecil, sebagai berikut:
>
> - Perilaku transisi default yang diperbarui — Mulai September 2024, perilaku default mencegah objek yang lebih kecil dari 128 KB dialihkan ke kelas penyimpanan apa pun.
> - Perilaku transisi default sebelumnya — Sebelum September 2024, perilaku default memungkinkan objek yang lebih kecil dari 128 KB untuk dialihkan hanya ke kelas penyimpanan S3 Glacier dan S3 Glacier Deep Archive.
>
> Konfigurasi yang dibuat sebelum September 2024 mempertahankan perilaku transisi sebelumnya kecuali Anda memodifikasinya. Artinya, jika Anda membuat, mengedit, atau menghapus aturan, perilaku transisi default untuk konfigurasi Anda berubah ke perilaku yang diperbarui. Jika kasus penggunaan diperlukan, Anda dapat mengubah perilaku transisi default sehingga objek yang lebih kecil dari 128KB akan bertransisi ke S3 Glacier dan S3 Glacier Deep Archive. Untuk melakukan ini, gunakan `x-amz-transition-default-minimum-object-size` header opsional dalam PutBucketLifecycleConfigurationpermintaan.

Objek harus disimpan setidaknya selama 30 hari sebelum beralih ke S3 Standard-IA atau S3 One Zone-IA

Sebelum Anda melakukan transisi objek ke S3 Standard-IA atau S3 One Zone-IA, Anda harus menyimpannya setidaknya 30 hari di Amazon S3. Misalnya, Anda tidak dapat membuat aturan Siklus Hidup untuk memindahkan objek ke kelas penyimpanan S3 Standard-IA satu hari setelah Anda membuatnya. Amazon S3 tidak mendukung transisi ini dalam 30 hari pertama karena objek yang lebih baru sering diakses atau dihapus lebih cepat daripada yang sesuai untuk penyimpanan S3 Standard-IA atau S3 One Zone-IA.

Demikian pula, jika Anda sedang melakukan transisi objek lama (dalam bucket berversi), Anda hanya dapat melakukan transisi objek yang setidaknya 30 hari lamanya ke penyimpanan S3 Standard-IA atau S3 One Zone-IA. Untuk daftar durasi penyimpanan minimum untuk semua kelas penyimpanan, lihatMembandingkan kelas penyimpanan Amazon S3.

Anda dikenakan biaya untuk mentransisikan objek sebelum durasi penyimpanan minimumnya

Kelas penyimpanan tertentu memiliki durasi penyimpanan objek minimum. Jika Anda mengalihkan objek dari kelas penyimpanan ini sebelum durasi minimum, Anda akan dikenakan biaya untuk sisa durasi tersebut. Untuk informasi selengkapnya tentang kelas penyimpanan mana yang memiliki durasi penyimpanan minimum, lihatMembandingkan kelas penyimpanan Amazon S3.

Anda tidak dapat membuat aturan Siklus Hidup tunggal yang mentransisikan objek dari satu kelas penyimpanan ke kelas penyimpanan lainnya sebelum periode durasi penyimpanan minimum berlalu.

Misalnya, S3 Glacier Instant Retrieval memiliki durasi penyimpanan minimum 90 hari. Anda tidak dapat menentukan aturan siklus hidup yang mentransisikan objek ke S3 Glacier Instant Retrieval setelah 4 hari, lalu mentransisikan objek ke S3 Glacier Deep Archive setelah 20 hari. Dalam hal ini transisi S3 Glacier Deep Archive harus terjadi setelah setidaknya 94 hari.

Anda dapat menentukan dua aturan untuk mencapai hal ini, tetapi Anda membayar biaya penyimpanan durasi minimum. Untuk informasi selengkapnya tentang pertimbangan biaya, lihat Harga Amazon S3.

Untuk informasi selengkapnya tentang membuat Siklus Hidup S3, lihat. Menyetel konfigurasi Siklus Hidup S3 pada bucket

Transisi ke kelas penyimpanan S3 Glacier Flexible Retrieval dan S3 Glacier Deep Archive (pengarsipan objek)

Dengan menggunakan konfigurasi Siklus Hidup S3, Anda dapat mentransisikan objek ke kelas penyimpanan S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive untuk pengarsipan.

Sebelum mengarsipkan objek, pelajari bagian berikut untuk pertimbangan yang relevan.

Pertimbangan umum

Berikut ini adalah pertimbangan umum yang perlu Anda pertimbangkan sebelum mengarsipkan objek:

- Objek terenkripsi tetap dienkripsi selama proses transisi kelas penyimpanan.
- Objek yang disimpan di kelas penyimpanan S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive tidak tersedia secara waktu nyata.

  Objek yang diarsipkan adalah objek Amazon S3, tetapi sebelum Anda dapat mengakses objek yang diarsipkan, Anda harus memulihkan salinan sementara terlebih dahulu. Salinan objek yang dipulihkan hanya tersedia selama durasi yang Anda tentukan dalam permintaan pemulihan.

Setelah itu, Amazon S3 menghapus salinan sementara, dan objek tetap diarsipkan di S3 Glacier Flexible Retrieval.

Anda dapat memulihkan objek dengan menggunakan konsol Amazon S3 atau secara terprogram dengan menggunakan pustaka pembungkus AWS SDK atau Amazon S3 REST API dalam kode Anda. Untuk informasi selengkapnya, lihat Memulihkan objek yang diarsipkan.

- Objek yang disimpan di kelas penyimpanan S3 Glacier Flexible Retrieval hanya dapat dialihkan ke kelas penyimpanan S3 Glacier Deep Archive.

  Anda dapat menggunakan aturan konfigurasi Siklus Hidup S3 untuk mengonversi kelas penyimpanan objek dari S3 Glacier Flexible Retrieval ke kelas penyimpanan S3 Glacier Deep Archive saja. Jika Anda ingin mengubah kelas penyimpanan objek yang disimpan dalam S3 Glacier Flexible Retrieval ke kelas penyimpanan selain S3 Glacier Deep Archive, Anda harus menggunakan operasi pemulihan untuk membuat salinan sementara dari objek terlebih dahulu. Kemudian gunakan operasi penyalinan untuk menimpa objek yang menentukan S3 Standard, S3 Intelligent-Tiering, S3 Standard-IA, S3 One Zone-IA, atau Reduced Redundancy sebagai kelas penyimpanan.

- Transisi objek ke kelas penyimpanan S3 Glacier Deep Archive hanya dapat menggunakan satu cara.

  Anda tidak dapat menggunakan aturan konfigurasi Siklus Hidup S3 untuk mengonversi kelas penyimpanan objek dari S3 Glacier Deep Archive ke kelas penyimpanan lainnya. Jika Anda ingin mengubah kelas penyimpanan dari objek yang diarsipkan ke kelas penyimpanan lain, Anda harus menggunakan operasi pemulihan untuk membuat salinan sementara dari objek terlebih dahulu. Kemudian gunakan operasi penyalinan untuk menimpa objek yang menentukan S3 Standard, S3 Intelligent-Tiering, S3 Standard-IA, S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval, atau Reduced Redundancy Storage sebagai kelas penyimpanan.

  > **Note**
  >
  > Operasi Salin untuk objek yang dipulihkan tidak didukung di konsol S3 Amazon untuk objek di kelas penyimpanan S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive. Untuk jenis operasi Copy ini, gunakan AWS Command Line Interface (AWS CLI), the AWS SDKs, atau REST API.

Objek yang disimpan di kelas penyimpanan S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive hanya dapat dilihat dan tersedia melalui Amazon S3. Objek ini tidak tersedia melalui layanan Amazon S3 Glacier yang terpisah.

Ini adalah objek Amazon S3, dan Anda dapat mengaksesnya hanya dengan menggunakan konsol Amazon S3 atau API Amazon S3. Anda tidak dapat mengakses objek yang diarsipkan melalui konsol Amazon S3 Glacier terpisah atau API Amazon S3 Glacier.

Pertimbangan biaya

Jika Anda berencana untuk mengarsipkan data yang jarang diakses selama beberapa bulan atau tahun, kelas penyimpanan S3 Glacier Flexible Retrieval dan S3 Glacier Deep Archive dapat mengurangi biaya penyimpanan Anda. Namun, untuk memastikan bahwa kelas penyimpanan S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive sesuai bagi Anda, pertimbangkan hal berikut ini:

- Biaya overhead penyimpanan–Saat Anda melakukan transisi objek ke kelas penyimpanan S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive, jumlah penyimpanan tetap ditambahkan ke setiap objek untuk mengakomodasi metadata pengelolaan objek.
  - Untuk setiap objek yang diarsipkan ke S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive, Amazon S3 menggunakan penyimpanan 8 KB untuk nama objek dan metadata lainnya. Amazon S3 menyimpan metadata ini sehingga Anda bisa mendapatkan daftar objek yang diarsipkan secara waktu nyata dengan menggunakan API Amazon S3. Untuk informasi selengkapnya, lihat Get Bucket (Daftar Objek). Anda dikenakan tarif Standar S3 untuk penyimpanan tambahan ini.
  - Untuk setiap objek yang diarsipkan ke S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive, Amazon S3 menambahkan penyimpanan 32 KB untuk indeks dan metadata terkait. Data ekstra ini diperlukan untuk mengidentifikasi dan memulihkan objek Anda. Anda dikenakan tarif S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive untuk penyimpanan tambahan ini.

  Jika Anda mengarsipkan objek kecil, pertimbangkan biaya penyimpanan ini. Pertimbangkan juga untuk menggabungkan banyak objek kecil menjadi sejumlah kecil objek besar untuk mengurangi biaya overhead.
- Jumlah hari yang Anda rencanakan untuk menyimpan objek yang diarsipkan–S3 Glacier Flexible Retrieval dan S3 Glacier Deep Archive adalah solusi pengarsipan jangka panjang. Periode durasi

penyimpanan minimal adalah 90 hari untuk kelas penyimpanan S3 Glacier Flexible Retrieval dan 180 hari untuk S3 Glacier Deep Archive. Menghapus data yang diarsipkan ke Amazon S3 Glacier tidak dikenakan biaya jika objek yang Anda hapus diarsipkan lebih dari periode durasi penyimpanan minimal. Jika Anda menghapus atau menimpa objek yang diarsipkan dalam periode durasi minimal, Amazon S3 mengenakan biaya penghapusan dini prorata. Untuk informasi tentang biaya penghapusan dini, lihat "Bagaimana saya dikenakan biaya untuk menghapus objek dari Amazon S3 Glacier yang berumur kurang dari 90 hari?" pertanyaan di FAQ Amazon S3.

- Biaya permintaan transisi S3 Glacier Flexible Retrieval dan S3 Glacier Deep Archive–Masing-masing objek yang Anda transisikan ke kelas penyimpanan S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive merupakan satu permintaan transisi. Ada biaya untuk setiap permintaan tersebut. Jika Anda berencana untuk mentransisikan banyak objek, pertimbangkan biaya permintaan. Jika Anda mengarsipkan campuran objek yang menyertakan objek kecil, terutama yang di bawah 128KB, sebaiknya gunakan filter ukuran objek siklus hidup untuk menyaring objek kecil dari transisi Anda guna mengurangi biaya permintaan.
- Biaya pemulihan data S3 Glacier Flexible Retrieval dan S3 Glacier Deep Archive–S3 Glacier Flexible Retrieval dan S3 Glacier Deep Archive dirancang untuk pengarsipan data jangka panjang yang jarang Anda akses. Untuk informasi tentang biaya pemulihan data, lihat "Berapa biaya untuk mengambil data dari Amazon S3 Glacier?" pertanyaan di FAQ Amazon S3. Untuk informasi tentang cara memulihkan data dari Amazon S3 Glacier, lihat Memulihkan objek yang diarsipkan.

> **Note**
>
> Siklus Hidup S3 mentransisikan objek ke S3 Glacier Flexible Retrieval dan S3 Glacier Deep Archive secara asinkron. Mungkin ada penundaan antara tanggal transisi dalam aturan konfigurasi Siklus Hidup S3 dan tanggal transisi fisik. Dalam hal ini Anda dikenakan tarif default dari kelas penyimpanan tempat Anda beralih berdasarkan tanggal transisi yang ditentukan dalam aturan.

Halaman detail produk Amazon S3 menyediakan informasi harga dan contoh perhitungan untuk mengarsipkan objek Amazon S3. Untuk informasi selengkapnya, lihat topik berikut:

- "Bagaimana biaya penyimpanan saya dihitung untuk objek Amazon S3 yang diarsipkan ke Amazon S3 Glacier?" di FAQ Amazon S3.
- "Bagaimana saya dikenakan biaya untuk menghapus objek dari Amazon S3 Glacier yang berumur kurang dari 90 hari?" di FAQ Amazon S3.

- “Berapa biaya untuk mengambil data dari Amazon S3 Glacier?” di FAQ Amazon S3.
- Harga Amazon S3 untuk biaya penyimpanan untuk berbagai kelas penyimpanan.

Memulihkan objek yang diarsipkan

Objek yang diarsipkan tidak dapat diakses secara real time. Anda harus memulai permintaan pemulihan terlebih dahulu, kemudian menunggu hingga salinan sementara objek tersedia selama durasi yang Anda tentukan dalam permintaan. Setelah Anda menerima salinan sementara dari objek yang dipulihkan, kelas penyimpanan objek tetaplah S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive. (A HeadObject atau GetObjectPermintaan operasi API akan mengembalikan S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive sebagai kelas penyimpanan.)

> **Note**
>
> Saat memulihkan arsip, Anda membayar untuk arsip tersebut (S3 Glacier Flexible Retrieval atau tingkat S3 Glacier Deep Archive) dan salinan yang Anda pulihkan sementara (tingkat penyimpanan S3 Standard). Untuk informasi selengkapnya tentang harga, silakan lihat Harga Amazon S3.

Anda dapat memulihkan salinan objek secara terprogram atau dengan menggunakan konsol Amazon S3. Amazon S3 hanya memproses satu permintaan pemulihan pada satu waktu per objek. Untuk informasi selengkapnya, lihat Memulihkan objek yang diarsipkan.

Mengakhiri objek

Anda dapat menambahkan tindakan transisi ke konfigurasi Siklus Hidup S3 untuk memberi tahu Amazon S3 agar menghapus objek di akhir masa pakainya. Saat objek mencapai akhir masa pakainya berdasarkan konfigurasi siklus hidupnya, Amazon S3 mengambil `Expiration` tindakan berdasarkan status Pembuatan Versi S3 tempat bucket berada:

- Bucket nonversioned — Amazon S3 mengantri objek untuk dihapus dan menghapusnya secara asinkron, menghapus objek secara permanen.
- Bucket dengan dukungan Penentuan Versi–Jika versi objek saat ini bukan penanda hapus, Amazon S3 menambahkan penanda hapus dengan ID versi unik. Ini membuat versi saat ini menjadi versi lama, dan penanda hapus menjadi versi saat ini.

- Bucket dengan Penentuan Versi ditangguhkan–Amazon S3 membuat penanda hapus dengan null sebagai ID versi. Penanda hapus ini menggantikan versi objek apa pun dengan ID versi null dalam hierarki versi, yang secara efektif menghapus objek.

Untuk bucket berversi (yaitu, berkemampuan versi atau ditangguhkan versi), ada beberapa pertimbangan yang memandu cara Amazon S3 menangani tindakan tersebut. `Expiration` Untuk bucket dengan dukungan Penentuan Versi atau Penentuan Versi ditangguhkan, berlaku berikut ini:

- Kedaluwarsa objek hanya berlaku untuk versi objek saat ini (tidak memiliki dampak pada versi objek lama).
- Amazon S3 tidak mengambil tindakan apa pun jika ada satu atau beberapa versi objek dan penanda hapus adalah versi saat ini.
- Jika versi objek saat ini adalah satu-satunya versi objek dan juga merupakan penanda hapus (juga disebut sebagai penanda hapus objek kedaluwarsa, ketika semua versi objek dihapus dan Anda hanya memiliki penanda hapus yang tersisa), Amazon S3 menghapus penanda hapus objek kedaluwarsa. Anda juga dapat menggunakan `Expiration` tindakan untuk mengarahkan Amazon S3 untuk menghapus penanda penghapusan objek yang kedaluwarsa. Sebagai contoh, lihat Menghapus penanda penghapusan objek yang kedaluwarsa dalam bucket berkemampuan versi.
- Anda dapat menggunakan elemen `NoncurrentVersionExpiration` tindakan untuk mengarahkan Amazon S3 untuk menghapus versi objek noncurrent secara permanen. Objek yang dihapus ini tidak dapat dipulihkan. Anda dapat mendasarkan kedaluwarsa ini pada sejumlah hari tertentu sejak objek menjadi tidak aktif. Selain jumlah hari, Anda juga dapat memberikan jumlah maksimum versi noncurrent untuk dipertahankan (antara 1 dan 100). Nilai ini menentukan berapa banyak versi lama yang harus ada sebelum Amazon S3 dapat melakukan tindakan terkait pada versi tertentu. Untuk menentukan jumlah maksimum versi noncurrent, Anda juga harus menyediakan `Filter` elemen. Jika Anda tidak menentukan `Filter` elemen, Amazon S3 menghasilkan `InvalidRequest` kesalahan saat Anda memberikan jumlah maksimum versi noncurrent. Untuk informasi selengkapnya tentang menggunakan elemen `NoncurrentVersionExpiration` tindakan, lihatthe section called “Elemen untuk mendeskripsikan tindakan siklus hidup”.
- Amazon S3 tidak mengambil tindakan apa pun pada versi objek yang tidak terkini yang menerapkan konfigurasi Kunci Objek S3.
- Untuk objek dengan status `Pending` replikasi, Amazon S3 tidak mengambil tindakan apa pun pada versi objek saat ini atau yang tidak saat ini.

Untuk informasi selengkapnya, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.

> **Important**
>
> Jika Anda memiliki beberapa aturan dalam konfigurasi Siklus Hidup S3, objek dapat memenuhi syarat untuk beberapa tindakan Siklus Hidup S3 pada hari yang sama. Dalam kasus tersebut, Amazon S3 mengikuti aturan umum ini:
>
> - Penghapusan permanen lebih diutamakan daripada transisi.
> - Transisi lebih diutamakan daripada pembuatan penanda hapus.
> - Ketika objek memenuhi syarat untuk transisi S3 Glacier Flexible Retrieval dan S3 Standard-IA (atau S3 One Zone-IA) transisi, Amazon S3 memilih transisi S3 Glacier Flexible Retrieval.
>
> Sebagai contoh, lihat Contoh filter yang tumpang tindih dan tindakan siklus hidup yang bertentangan.

Objek yang ada dan baru

Ketika Anda menambahkan konfigurasi Siklus Hidup ke dalam bucket, aturan konfigurasi berlaku untuk objek yang ada dan objek yang Anda tambahkan kemudian. Misalnya, jika Anda menambahkan aturan konfigurasi Siklus Hidup hari ini dengan tindakan kedaluwarsa yang menyebabkan objek dengan awalan tertentu kedaluwarsa 30 hari setelah pembuatan, Amazon S3 akan mengantri untuk menghapus objek yang ada yang berusia lebih dari 30 hari dan yang memiliki awalan yang ditentukan.

> **Important**
>
> Anda tidak dapat menggunakan kebijakan bucket untuk mencegah penghapusan atau transisi oleh aturan Siklus Hidup S3. Misalnya, meskipun kebijakan bucket Anda menolak semua tindakan untuk semua prinsipal, konfigurasi Siklus Hidup S3 Anda tetap berfungsi seperti biasa.

Cara mengetahui kapan objek akan kedaluwarsa

Untuk mengetahui kapan versi objek saat ini dijadwalkan kedaluwarsa, gunakan HeadObject atau GetObjectOperasi API. Operasi API ini mengembalikan header respons yang memberikan tanggal dan waktu di mana versi objek saat ini tidak lagi dapat di-cache.

> **Note**
>
> - Mungkin ada penundaan antara tanggal kedaluwarsa dan tanggal ketika Amazon S3 menghapus objek. Anda tidak dikenakan biaya kedaluwarsa atau waktu penyimpanan yang terkait dengan objek yang telah kedaluwarsa.
> - Sebelum memperbarui, menonaktifkan, atau menghapus aturan Siklus Hidup, gunakan operasi API (seperti `LIST` ListObjectsV2, ListObjectVersions, dan ListMultipartUploads) atau Membuat katalog dan menganalisis data Anda dengan S3 Inventory untuk memverifikasi bahwa Amazon S3 telah mentransisikan dan menghapus objek yang memenuhi syarat berdasarkan kasus penggunaan Anda.

Biaya durasi penyimpanan minimum

Jika Anda membuat aturan kedaluwarsa Siklus Hidup S3 yang menyebabkan objek dalam penyimpanan S3 Standard-IA atau S3 One Zone-IA selama kurang dari 30 hari kedaluwarsa, Anda akan dikenakan biaya selama 30 hari. Jika Anda membuat aturan kedaluwarsa Siklus Hidup yang menyebabkan objek dalam penyimpanan S3 Glacier Flexible Retrieval selama kurang dari 90 hari kedaluwarsa, Anda akan dikenakan biaya selama 90 hari. Jika Anda membuat aturan kedaluwarsa Siklus Hidup yang menyebabkan objek dalam penyimpanan S3 Glacier Deep Archive selama kurang dari 180 hari kedaluwarsa, Anda akan dikenakan biaya selama 180 hari.

Untuk informasi selengkapnya, lihat Harga Amazon S3.

Menyetel konfigurasi Siklus Hidup S3 pada bucket

Anda dapat menyetel konfigurasi Siklus Hidup Amazon S3 pada bucket menggunakan konsol Amazon S3, AWS Command Line Interface (AWS CLI), API REST Amazon S3, atau Amazon S3 AWS SDKs REST. Untuk informasi tentang konfigurasi Siklus Hidup S3, lihat Mengelola siklus hidup objek.

> **Note**
>
> Untuk melihat atau mengedit konfigurasi siklus hidup bucket direktori, gunakan AWS CLI AWS SDKs, atau Amazon S3 REST API. Untuk informasi selengkapnya, lihat Bekerja dengan Siklus Hidup S3 untuk bucket direktori.

Dalam konfigurasi Siklus Hidup S3, Anda menggunakan aturan siklus hidup untuk menentukan tindakan yang ingin dilakukan Amazon S3 selama masa pakai objek. Misalnya, Anda dapat menentukan aturan untuk mentransisikan objek ke kelas penyimpanan lain, objek arsip, atau objek kedaluwarsa (hapus) setelah jangka waktu tertentu.

Pertimbangan Siklus Hidup S3

Sebelum Anda mengatur konfigurasi siklus hidup, perhatikan hal berikut ini:

Penundaan propagasi konfigurasi siklus hidup

Saat Anda menambahkan konfigurasi Siklus Hidup S3 ke dalam bucket, biasanya ada beberapa jeda sebelum konfigurasi Siklus Hidup baru atau yang diperbarui sepenuhnya disebarkan ke semua sistem Amazon S3. Perkirakan penundaan beberapa menit sebelum konfigurasi benar-benar berpengaruh. Penundaan ini juga dapat terjadi saat Anda menghapus konfigurasi Siklus Hidup S3.

Transisi atau penundaan kedaluwarsa

Ada penundaan antara saat aturan siklus hidup terpenuhi dan saat tindakan untuk aturan selesai. Misalnya, seperangkat objek kedaluwarsa oleh aturan siklus hidup pada 1 Januari. Meskipun aturan kedaluwarsa telah dipenuhi pada 1 Januari, Amazon S3 mungkin tidak benar-benar menghapus objek ini sampai berhari-hari atau bahkan berminggu-minggu kemudian. Penundaan ini terjadi karena Siklus Hidup S3 mengantri objek untuk transisi atau kedaluwarsa secara asinkron. Namun, perubahan dalam penagihan biasanya diterapkan ketika aturan siklus hidup dipenuhi, bahkan jika tindakan tidak selesai. Untuk informasi selengkapnya, lihat Perubahan dalam penagihan. Untuk memantau efek pembaruan yang dibuat oleh aturan siklus hidup aktif, lihat the section called “Bagaimana cara memantau tindakan yang diambil oleh aturan siklus hidup saya?”

Memperbarui, menonaktifkan, atau menghapus aturan siklus hidup

Saat Anda menonaktifkan atau menghapus aturan siklus hidup, Amazon S3 berhenti menjadwalkan objek baru untuk dihapus atau dialihkan setelah penundaan kecil. Objek yang sudah dijadwalkan menjadi tidak dijadwalkan dan tidak dihapus atau ditransisikan.

> **Note**
>
> Sebelum memperbarui, menonaktifkan, atau menghapus aturan siklus hidup, gunakan operasi API (seperti `LIST` ListObjectsV2, ListObjectVersions, dan ListMultipartUploads) atau Membuat katalog dan menganalisis data Anda dengan S3 Inventory untuk memverifikasi bahwa Amazon S3 telah mentransisikan dan menghapus objek yang memenuhi syarat berdasarkan kasus penggunaan Anda. Jika Anda mengalami masalah dengan memperbarui, menonaktifkan, atau menghapus aturan siklus hidup, lihat. Memecahkan masalah Siklus Hidup Amazon S3

Objek yang ada dan baru

Ketika Anda menambahkan konfigurasi Siklus Hidup ke dalam bucket, aturan konfigurasi berlaku untuk objek yang ada dan objek yang Anda tambahkan kemudian. Misalnya, jika Anda menambahkan aturan konfigurasi Siklus Hidup hari ini dengan tindakan kedaluwarsa yang menyebabkan objek dengan awalan tertentu kedaluwarsa 30 hari setelah pembuatan, Amazon S3 akan mengantri untuk menghapus objek yang ada yang berusia lebih dari 30 hari dan yang memiliki awalan yang ditentukan.

Memantau pengaruh aturan siklus hidup

Untuk memantau efek pembaruan yang dibuat oleh aturan siklus hidup aktif, lihat the section called “Bagaimana cara memantau tindakan yang diambil oleh aturan siklus hidup saya?”

Perubahan penagihan

Mungkin ada jeda antara saat aturan konfigurasi Siklus Hidup dipenuhi dan saat tindakan yang dipicu dengan memenuhi aturan diambil. Namun, perubahan penagihan terjadi segera setelah aturan konfigurasi Siklus Hidup dipenuhi, meskipun tindakan belum diambil.

Misalnya, setelah waktu kedaluwarsa objek, Anda tidak dikenakan biaya untuk penyimpanan, meskipun objek tidak segera dihapus. Demikian juga, segera setelah waktu transisi objek berlalu, Anda dikenakan biaya tingkat penyimpanan S3 Glacier Flexible Retrieval, bahkan jika objek tidak segera dialihkan ke kelas penyimpanan S3 Glacier Flexible Retrieval.

Namun, transisi siklus hidup ke kelas penyimpanan S3 Intelligent-Tiering adalah pengecualian. Perubahan penagihan tidak terjadi sampai setelah objek bertransisi ke kelas penyimpanan S3 Intelligent-Tiering.

Aturan ganda atau bertentangan

Jika Anda memiliki beberapa aturan dalam konfigurasi Siklus Hidup S3, objek dapat memenuhi syarat untuk beberapa tindakan Siklus Hidup S3 pada hari yang sama. Dalam kasus tersebut, Amazon S3 mengikuti aturan umum ini:

- Penghapusan permanen lebih diutamakan daripada transisi.
- Transisi lebih diutamakan daripada pembuatan penanda hapus.
- Ketika objek memenuhi syarat untuk transisi S3 Glacier Flexible Retrieval dan S3 Standard-IA (atau S3 One Zone-IA) transisi, Amazon S3 memilih transisi S3 Glacier Flexible Retrieval.

Sebagai contoh, lihat Contoh filter yang tumpang tindih dan tindakan siklus hidup yang bertentangan.

Cara mengatur konfigurasi Siklus Hidup S3

Anda dapat menyetel konfigurasi Siklus Hidup Amazon S3 pada bucket tujuan umum menggunakan konsol Amazon S3, AWS Command Line Interface (AWS CLI), API REST Amazon S3, atau Amazon S3 AWS SDKs REST.

Untuk informasi tentang AWS CloudFormation templat dan contoh, lihat Bekerja dengan AWS CloudFormation templat dan AWS::S3::Bucket di Panduan Pengguna AWS CloudFormation .

Menggunakan konsol S3

Anda dapat menentukan aturan siklus hidup untuk semua objek atau subset objek dalam bucket dengan menggunakan awalan bersama (nama objek yang dimulai dengan string umum) atau tag. Dalam aturan siklus hidup, Anda dapat menentukan tindakan khusus untuk versi objek saat ini dan noncurrent. Untuk informasi selengkapnya, lihat berikut ini:

- Mengelola siklus hidup objek
- Mempertahankan beberapa versi objek dengan S3 Versioning

Untuk membuat aturan siklus hidup

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.

3. Dalam daftar bucket, pilih nama bucket yang ingin Anda buat aturan siklus hidup.
4. Pilih tab Manajemen, dan pilih Buat aturan siklus hidup.
5. Di Nama aturan siklus hidup, masukkan nama untuk aturan Anda.

   Nama dalam bucket harus unik.
6. Pilih cakupan aturan siklus hidup:

   - Untuk menerapkan aturan siklus hidup ini pada semua objek dengan awalan atau tanda khusus, pilih Batasi cakupan ke awalan atau tanda tertentu.
     - Untuk membatasi cakupan dengan awalan, di Awalan, masukkan awalan.
     - Untuk membatasi cakupan berdasarkan tanda, pilih Tambahkan tanda, dan masukkan kunci dan nilai tanda.

     Untuk informasi selengkapnya tentang awalan nama objek, lihat Menamai objek Amazon S3. Untuk informasi selengkapnya tentang tag objek, lihat Mengategorikan penyimpanan Anda menggunakan tag.
   - Untuk menerapkan aturan siklus hidup ini ke semua objek di bucket, pilih Aturan ini berlaku untuk semua objek di bucket, lalu pilih Saya mengakui bahwa aturan ini berlaku untuk semua objek di bucket.
7. Untuk memfilter aturan berdasarkan ukuran objek, Anda dapat memilih Tentukan ukuran objek minimum, Tentukan ukuran objek maksimum, atau kedua opsi.

   - Saat Anda menentukan nilai untuk ukuran objek Minimum atau ukuran objek Maksimum, nilainya harus lebih besar dari 0 byte dan hingga 5 TB. Anda dapat menentukan nilai ini dalam bita, KB, MB, atau GB.
   - Saat Anda menentukan kedua nilai, ukuran objek maksimum harus lebih besar dari ukuran objek minimum.

   > **Note**
   >
   > Ukuran objek Minimum dan filter ukuran objek Maksimum mengecualikan nilai yang ditentukan. Misalnya, jika Anda menyetel filter untuk objek kedaluwarsa yang memiliki ukuran objek Minimum 128 KB, objek yang persis 128 KB tidak akan kedaluwarsa. Sebaliknya, aturan hanya berlaku untuk objek yang berukuran lebih besar dari 128 KB.
8. Di bawah Tindakan aturan siklus hidup, pilih tindakan yang Anda ingin aturan siklus hidup Anda lakukan:

- Transisikan versi objek saat ini antar kelas penyimpanan
- Transisikan versi objek sebelumnya antar kelas penyimpanan
- Akhiri versi objek saat ini

> **Note**
>
> Untuk bucket yang tidak mengaktifkan Versi S3, versi saat ini yang kedaluwarsa menyebabkan Amazon S3 menghapus objek secara permanen. Untuk informasi selengkapnya, lihat the section called “Tindakan Siklus Hidup dan kondisi Penentuan Versi bucket”.

- Hapus secara permanen versi objek sebelumnya
- Hapus penanda hapus yang kedaluwarsa atau unggahan multibagian yang belum selesai

Tergantung pada tindakan yang Anda pilih, opsi yang berbeda akan muncul.

9. Untuk mentransisikan versi objek saat ini di antara kelas penyimpanan, di bawah Transisi versi objek saat ini di antara kelas penyimpanan, lakukan hal berikut:

   a. Dalam transisi kelas Storage, pilih kelas penyimpanan untuk transisi. Untuk daftar kemungkinan transisi, lihatthe section called “Transisi siklus hidup yang didukung”. Anda dapat memilih dari kelas penyimpanan berikut:

      - S3 Standard-IA
      - S3 Intelligent-Tiering
      - S3 One Zone-IA
      - S3 Glacier Instant Retrieval
      - S3 Glacier Flexible Retrieval
      - S3 Glacier Deep Archive

   b. Di Hari setelah pembuatan objek, masukkan jumlah hari setelah pembuatan untuk melakukan transisi objek.

   Untuk informasi selengkapnya tentang kelas penyimpanan, lihat Memahami dan mengelola kelas penyimpanan Amazon S3. Anda dapat menentukan transisi untuk versi objek saat ini atau sebelumnya atau untuk versi saat ini dan versi sebelumnya. Penentuan Versi memungkinkan

Anda untuk menyimpan beberapa versi dari sebuah objek di dalam satu bucket. Untuk informasi selengkapnya tentang penentuan versi, lihat Menggunakan konsol S3.

> **Important**
>
> Saat Anda memilih kelas penyimpanan S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval, atau Glacier Deep Archive, objek Anda tetap berada di Amazon S3. Anda tidak dapat mengaksesnya secara langsung melalui layanan Amazon S3 Glacier yang terpisah. Untuk informasi selengkapnya, lihat Transisi objek menggunakan Siklus Hidup Amazon S3.

10. Untuk mentransisikan versi noncurrent objek antar kelas penyimpanan, di bawah Transisi versi noncurrent objek antar kelas penyimpanan, lakukan hal berikut:

    a. Dalam transisi kelas Storage, pilih kelas penyimpanan untuk transisi. Untuk daftar kemungkinan transisi, lihatthe section called “Transisi siklus hidup yang didukung”. Anda dapat memilih dari kelas penyimpanan berikut:

    - S3 Standard-IA
    - S3 Intelligent-Tiering
    - S3 One Zone-IA
    - S3 Glacier Instant Retrieval
    - S3 Glacier Flexible Retrieval
    - S3 Glacier Deep Archive

    b. Dalam Hari setelah objek menjadi tidak aktif, masukkan jumlah hari setelah pembuatan untuk transisi objek.

11. Untuk mengakhiri versi objek saat ini, di bawah Akhiri versi objek saat ini, di Jumlah hari setelah pembuatan objek, masukkan jumlah hari.

> **Important**
>
> Dalam bucket nonversioned, tindakan kedaluwarsa menghasilkan Amazon S3 menghapus objek secara permanen. Untuk informasi lebih lanjut tentang tindakan siklus hidup, lihat Elemen untuk mendeskripsikan tindakan siklus hidup.

12. Untuk menghapus objek versi sebelumnya secara permanen, di bawah Hapus objek versi sebelumnya secara permanen, di Hari setelah objek menjadi versi lama, masukkan jumlah hari.

Anda dapat secara opsional menentukan jumlah versi yang lebih baru untuk dipertahankan dengan memasukkan nilai di bawah Jumlah versi yang lebih baru untuk dipertahankan.

13. Di Bawah Hapus penanda hapus yang kedaluwarsa atau unggahan multibagian yang belum selesai, pilih Hapus penanda hapus objek kedaluwarsa dan Hapus unggahan multibagian yang belum selesai. Kemudian, masukkan jumlah hari setelah inisiasi pengunggahan multibagian yang ingin Anda akhiri dan bersihkan unggahan multibagian yang belum selesai.

    Untuk informasi lebih lanjut tentang unggahan multibagian, lihat Mengunggah dan menyalin objek menggunakan unggahan multipart di Amazon S3.

14. Pilih Buat aturan.

    Jika aturan tidak berisi kesalahan apa pun, Amazon S3 mengaktifkannya, dan Anda dapat melihatnya di tab Manajemen pada Aturan siklus hidup.

Menggunakan AWS CLI

Anda dapat menggunakan AWS CLI perintah berikut untuk mengelola konfigurasi Siklus Hidup S3:

- `put-bucket-lifecycle-configuration`
- `get-bucket-lifecycle-configuration`
- `delete-bucket-lifecycle`

Untuk petunjuk cara menyiapkan AWS CLI, lihat Mengembangkan dengan Amazon S3 menggunakan AWS CLI di Referensi API Amazon S3.

Konfigurasi Amazon Siklus Hidup S3 adalah berkas XML. Tetapi ketika Anda menggunakan AWS CLI, Anda tidak dapat menentukan format XMLnya. Anda harus menentukan format JSON sebagai gantinya. Berikut ini adalah contoh konfigurasi siklus hidup XMLdan konfigurasi JSON setara yang dapat Anda tentukan dalam sebuah perintah. AWS CLI

Pertimbangkan contoh konfigurasi Siklus Hidup S3 berikut.

Example Contoh 1

Example

XML

```
<LifecycleConfiguration>
```

```
    <Rule>
        <ID>ExampleRule</ID>
        <Filter>
           <Prefix>documents/</Prefix>
        </Filter>
        <Status>Enabled</Status>
        <Transition>
           <Days>365</Days>
           <StorageClass>GLACIER</StorageClass>
        </Transition>
        <Expiration>
             <Days>3650</Days>
        </Expiration>
    </Rule>
</LifecycleConfiguration>
```

JSON

```
{
    "Rules": [
        {
            "Filter": {
                "Prefix": "documents/"
            },
            "Status": "Enabled",
            "Transitions": [
                {
                    "Days": 365,
                    "StorageClass": "GLACIER"
                }
            ],
            "Expiration": {
                "Days": 3650
            },
            "ID": "ExampleRule"
        }
    ]
}
```

Example Contoh 2

Example

XML

```
<LifecycleConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
    <Rule>
        <ID>id-1</ID>
        <Expiration>
            <Days>1</Days>
        </Expiration>
        <Filter>
            <And>
                <Prefix>myprefix</Prefix>
                <Tag>
                    <Key>mytagkey1</Key>
                    <Value>mytagvalue1</Value>
                </Tag>
                <Tag>
                    <Key>mytagkey2</Key>
                    <Value>mytagvalue2</Value>
                </Tag>
            </And>
        </Filter>
        <Status>Enabled</Status>
    </Rule>
</LifecycleConfiguration>
```

JSON

```
{
    "Rules": [
        {
            "ID": "id-1",
            "Filter": {
                "And": {
                    "Prefix": "myprefix",
                    "Tags": [
                        {
                            "Value": "mytagvalue1",
                            "Key": "mytagkey1"
```

```
                        },
                        {
                            "Value": "mytagvalue2",
                            "Key": "mytagkey2"
                        }
                    ]
                }
            },
            "Status": "Enabled",
            "Expiration": {
                "Days": 1
            }
        }
    ]
}
```

Anda dapat menguji `put-bucket-lifecycle-configuration` sebagai berikut.

Untuk menguji konfigurasi

1. Simpan konfigurasi Siklus Hidup JSON dalam sebuah file (misalnya,). *`lifecycle.json`*
2. Jalankan AWS CLI perintah berikut untuk menyetel konfigurasi Siklus Hidup di bucket Anda. Ganti *`user input placeholders`* dengan informasi Anda sendiri.

   ```
   $ aws s3api put-bucket-lifecycle-configuration  \
   --bucket amzn-s3-demo-bucket  \
   --lifecycle-configuration file://lifecycle.json
   ```

3. Untuk memverifikasi, ambil konfigurasi Siklus Hidup S3 dengan menggunakan perintah sebagai `get-bucket-lifecycle-configuration` AWS CLI berikut:

   ```
   $ aws s3api get-bucket-lifecycle-configuration  \
   --bucket amzn-s3-demo-bucket
   ```

4. Untuk menghapus konfigurasi Siklus Hidup S3, gunakan `delete-bucket-lifecycle` AWS CLI perintah sebagai berikut:

   ```
   aws s3api delete-bucket-lifecycle \
   --bucket amzn-s3-demo-bucket
   ```

Menggunakan AWS SDKs

Java

Anda dapat menggunakan AWS SDK untuk Java untuk mengelola konfigurasi Siklus Hidup S3 bucket. Untuk informasi selengkapnya tentang mengelola konfigurasi Siklus Hidup S3, lihat Mengelola siklus hidup objek.

> **Note**
>
> Saat Anda menambahkan konfigurasi Siklus Hidup S3 ke bucket, Amazon S3 menggantikan konfigurasi Siklus Hidup bucket saat ini, jika ada. Untuk memperbarui konfigurasi, Anda mengambilnya, membuat perubahan yang diinginkan, kemudian menambahkan konfigurasi yang direvisi ke bucket.

Contoh berikut menunjukkan cara menggunakan konfigurasi Siklus Hidup bucket AWS SDK untuk Java untuk menambah, memperbarui, dan menghapus konfigurasi Siklus Hidup. Contoh ini melakukan hal berikut:

- Menambahkan konfigurasi Siklus Hidup ke dalam bucket.
- Membuka kembali konfigurasi Siklus Hidup dan memperbaruinya dengan menambahkan aturan lain.
- Menambahkan konfigurasi Siklus Hidup yang telah dimodifikasi ke dalam bucket. Amazon S3 menggantikan konfigurasi yang ada.
- Mengambil konfigurasi lagi dan memverifikasi bahwa ia memiliki jumlah aturan yang tepat dengan mencetak jumlah aturan.
- Menghapus konfigurasi Siklus Hidup dan memverifikasi bahwa konfigurasi telah dihapus dengan mencoba mengambilnya kembali.

Untuk petunjuk cara membuat dan menguji sampel yang berfungsi, lihat Memulai di Panduan AWS SDK untuk Java Pengembang.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
```

```
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.BucketLifecycleConfiguration;
import com.amazonaws.services.s3.model.BucketLifecycleConfiguration.Transition;
import com.amazonaws.services.s3.model.StorageClass;
import com.amazonaws.services.s3.model.Tag;
import com.amazonaws.services.s3.model.lifecycle.LifecycleAndOperator;
import com.amazonaws.services.s3.model.lifecycle.LifecycleFilter;
import com.amazonaws.services.s3.model.lifecycle.LifecyclePrefixPredicate;
import com.amazonaws.services.s3.model.lifecycle.LifecycleTagPredicate;

import java.io.IOException;
import java.util.Arrays;

public class LifecycleConfiguration {

        public static void main(String[] args) throws IOException {
                Regions clientRegion = Regions.DEFAULT_REGION;
                String bucketName = "*** Bucket name ***";

                // Create a rule to archive objects with the "glacierobjects/"
 prefix to Glacier
                // immediately.
                BucketLifecycleConfiguration.Rule rule1 = new
 BucketLifecycleConfiguration.Rule()
                                .withId("Archive immediately rule")
                                .withFilter(new LifecycleFilter(new
 LifecyclePrefixPredicate("glacierobjects/")))
                                .addTransition(new
 Transition().withDays(0).withStorageClass(StorageClass.Glacier))
                                .withStatus(BucketLifecycleConfiguration.ENABLED);

                // Create a rule to transition objects to the Standard-Infrequent
 Access storage
                // class
                // after 30 days, then to Glacier after 365 days. Amazon S3 will
 delete the
                // objects after 3650 days.
                // The rule applies to all objects with the tag "archive" set to
 "true".
                BucketLifecycleConfiguration.Rule rule2 = new
 BucketLifecycleConfiguration.Rule()
                                .withId("Archive and then delete rule")
```

```
                                .withFilter(new LifecycleFilter(new
LifecycleTagPredicate(new Tag("archive", "true"))))
                                .addTransition(new Transition().withDays(30)

.withStorageClass(StorageClass.StandardInfrequentAccess))
                                .addTransition(new
Transition().withDays(365).withStorageClass(StorageClass.Glacier))
                                .withExpirationInDays(3650)
                                .withStatus(BucketLifecycleConfiguration.ENABLED);

                // Add the rules to a new BucketLifecycleConfiguration.
                BucketLifecycleConfiguration configuration = new
BucketLifecycleConfiguration()
                                .withRules(Arrays.asList(rule1, rule2));

                try {
                        AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                                        .withCredentials(new
ProfileCredentialsProvider())
                                        .withRegion(clientRegion)
                                        .build();

                        // Save the configuration.
                        s3Client.setBucketLifecycleConfiguration(bucketName,
configuration);

                        // Retrieve the configuration.
                        configuration =
s3Client.getBucketLifecycleConfiguration(bucketName);

                        // Add a new rule with both a prefix predicate and a tag
predicate.
                        configuration.getRules().add(new
BucketLifecycleConfiguration.Rule().withId("NewRule")
                                        .withFilter(new LifecycleFilter(new
LifecycleAndOperator(
                                                        Arrays.asList(new
LifecyclePrefixPredicate("YearlyDocuments/"),
                                                                        new
LifecycleTagPredicate(new Tag(

   "expire_after",

   "ten_years"))))))
```

```
                                        .withExpirationInDays(3650)

 .withStatus(BucketLifecycleConfiguration.ENABLED));

                        // Save the configuration.
                        s3Client.setBucketLifecycleConfiguration(bucketName,
 configuration);

                        // Retrieve the configuration.
                        configuration =
 s3Client.getBucketLifecycleConfiguration(bucketName);

                        // Verify that the configuration now has three rules.
                        configuration =
 s3Client.getBucketLifecycleConfiguration(bucketName);
                        System.out.println("Expected # of rules = 3; found: " +
 configuration.getRules().size());

                        // Delete the configuration.
                        s3Client.deleteBucketLifecycleConfiguration(bucketName);

                        // Verify that the configuration has been deleted by
 attempting to retrieve it.
                        configuration =
 s3Client.getBucketLifecycleConfiguration(bucketName);
                        String s = (configuration == null) ? "No configuration
 found." : "Configuration found.";
                        System.out.println(s);
                } catch (AmazonServiceException e) {
                        // The call was transmitted successfully, but Amazon S3
 couldn't process
                        // it, so it returned an error response.
                        e.printStackTrace();
                } catch (SdkClientException e) {
                        // Amazon S3 couldn't be contacted for a response, or the
 client
                        // couldn't parse the response from Amazon S3.
                        e.printStackTrace();
                }
        }
}
```

.NET

Anda dapat menggunakan AWS SDK untuk .NET untuk mengelola konfigurasi Siklus Hidup S3 pada bucket. Untuk informasi selengkapnya tentang mengelola konfigurasi Siklus Hidup, lihat Mengelola siklus hidup objek.

> **Note**
>
> Saat Anda menambahkan konfigurasi Siklus Hidup, Amazon S3 menggantikan konfigurasi yang ada pada bucket yang ditentukan. Untuk memperbarui konfigurasi, Anda harus mengembalikan konfigurasi Siklus Hidup terlebih dahulu, membuat perubahan, kemudian menambahkan konfigurasi Siklus Hidup yang telah direvisi ke bucket.

Contoh berikut menunjukkan cara menggunakan konfigurasi Siklus Hidup bucket AWS SDK untuk .NET untuk menambah, memperbarui, dan menghapus konfigurasi Siklus Hidup bucket. Contoh kode melakukan hal sebagai berikut:

- Menambahkan konfigurasi Siklus Hidup ke dalam bucket.
- Membuka kembali konfigurasi Siklus Hidup dan memperbaruinya dengan menambahkan aturan lain.
- Menambahkan konfigurasi Siklus Hidup yang telah dimodifikasi ke dalam bucket. Amazon S3 menggantikan konfigurasi Siklus Hidup yang ada.
- Mengambil kembali konfigurasi dan memverifikasinya dengan mencetak jumlah aturan dalam konfigurasi.
- Menghapus konfigurasi Siklus Hidup dan memverifikasi penghapusan.

Untuk informasi tentang menyiapkan dan menjalankan contoh kode, lihat Memulai AWS SDK untuk .NET di Panduan AWS SDK untuk .NET Pengembang.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
```

```
{
    class LifecycleTest
    {
        private const string bucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 client;
        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            AddUpdateDeleteLifecycleConfigAsync().Wait();
        }

        private static async Task AddUpdateDeleteLifecycleConfigAsync()
        {
            try
            {
                var lifeCycleConfiguration = new LifecycleConfiguration()
                {
                    Rules = new List<LifecycleRule>
                        {
                            new LifecycleRule
                            {
                                 Id = "Archive immediately rule",
                                 Filter = new LifecycleFilter()
                                 {
                                     LifecycleFilterPredicate = new
 LifecyclePrefixPredicate()
                                     {
                                         Prefix = "glacierobjects/"
                                     }
                                 },
                                 Status = LifecycleRuleStatus.Enabled,
                                 Transitions = new List<LifecycleTransition>
                                 {
                                      new LifecycleTransition
                                      {
                                           Days = 0,
                                           StorageClass = S3StorageClass.Glacier
                                      }
                                  },
                            },
                            new LifecycleRule
```

```
                            {
                                 Id = "Archive and then delete rule",
                                  Filter = new LifecycleFilter()
                                 {
                                     LifecycleFilterPredicate = new
LifecyclePrefixPredicate()
                                     {
                                         Prefix = "projectdocs/"
                                     }
                                 },
                                 Status = LifecycleRuleStatus.Enabled,
                                 Transitions = new List<LifecycleTransition>
                                 {
                                      new LifecycleTransition
                                      {
                                           Days = 30,
                                           StorageClass =
S3StorageClass.StandardInfrequentAccess
                                      },
                                      new LifecycleTransition
                                      {
                                        Days = 365,
                                        StorageClass = S3StorageClass.Glacier
                                      }
                                 },
                                 Expiration = new LifecycleRuleExpiration()
                                 {
                                       Days = 3650
                                 }
                            }
                        }
                };

                // Add the configuration to the bucket.
                await AddExampleLifecycleConfigAsync(client,
lifeCycleConfiguration);

                // Retrieve an existing configuration.
                lifeCycleConfiguration = await RetrieveLifecycleConfigAsync(client);

                // Add a new rule.
                lifeCycleConfiguration.Rules.Add(new LifecycleRule
                {
                    Id = "NewRule",
```

```
                    Filter = new LifecycleFilter()
                    {
                        LifecycleFilterPredicate = new LifecyclePrefixPredicate()
                        {
                            Prefix = "YearlyDocuments/"
                        }
                    },
                    Expiration = new LifecycleRuleExpiration()
                    {
                        Days = 3650
                    }
                });

                // Add the configuration to the bucket.
                await AddExampleLifecycleConfigAsync(client,
lifeCycleConfiguration);

                // Verify that there are now three rules.
                lifeCycleConfiguration = await RetrieveLifecycleConfigAsync(client);
                Console.WriteLine("Expected # of rulest=3; found:{0}",
lifeCycleConfiguration.Rules.Count);

                // Delete the configuration.
                await RemoveLifecycleConfigAsync(client);

                // Retrieve a nonexistent configuration.
                lifeCycleConfiguration = await RetrieveLifecycleConfigAsync(client);

            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered ***. Message:'{0}' when writing
an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
        }

        static async Task AddExampleLifecycleConfigAsync(IAmazonS3 client,
LifecycleConfiguration configuration)
        {
```

```
            PutLifecycleConfigurationRequest request = new
 PutLifecycleConfigurationRequest
            {
                BucketName = bucketName,
                Configuration = configuration
            };
            var response = await client.PutLifecycleConfigurationAsync(request);
        }

        static async Task<LifecycleConfiguration>
 RetrieveLifecycleConfigAsync(IAmazonS3 client)
        {
            GetLifecycleConfigurationRequest request = new
 GetLifecycleConfigurationRequest
            {
                BucketName = bucketName
            };
            var response = await client.GetLifecycleConfigurationAsync(request);
            var configuration = response.Configuration;
            return configuration;
        }

        static async Task RemoveLifecycleConfigAsync(IAmazonS3 client)
        {
            DeleteLifecycleConfigurationRequest request = new
 DeleteLifecycleConfigurationRequest
            {
                BucketName = bucketName
            };
            await client.DeleteLifecycleConfigurationAsync(request);
        }
    }
}
```

Ruby

Anda dapat menggunakan AWS SDK untuk Ruby untuk mengelola konfigurasi Siklus Hidup S3 pada bucket dengan menggunakan kelas AWS::S3::BucketLifecycleConfiguration. Untuk informasi selengkapnya tentang mengelola konfigurasi Siklus Hidup S3, lihat. Mengelola siklus hidup objek

Penggunaan API REST

Topik berikut dalam Referensi API Amazon Simple Storage Service menjelaskan operasi REST API yang terkait dengan konfigurasi Siklus Hidup S3:

- PutBucketLifecycleConfiguration
- GetBucketLifecycleConfiguration
- DeleteBucketLifecycle

Pemecahan Masalah Siklus Hidup S3

Untuk masalah umum yang mungkin terjadi saat bekerja dengan Siklus Hidup S3, lihat. the section called “Memecahkan masalah siklus hidup”

Bagaimana Siklus Hidup S3 berinteraksi dengan konfigurasi bucket lainnya

Selain konfigurasi Siklus Hidup S3, Anda dapat mengaitkan konfigurasi lain dengan bucket Anda. Bagian ini menjelaskan bagaimana konfigurasi Siklus Hidup S3 berkaitan dengan konfigurasi bucket lainnya.

Siklus Hidup S3 dan Versi S3

Anda dapat menambahkan konfigurasi Siklus Hidup S3 ke bucket tanpa versi dan bucket dengan dukungan Penentuan Versi. Untuk informasi selengkapnya, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.

Bucket yang dengan dukungan Penentuan Versi mempertahankan satu versi objek saat ini, dan nol atau lebih versi objek yang lama. Anda dapat menentukan aturan siklus hidup terpisah untuk versi objek saat ini dan noncurrent.

Untuk informasi selengkapnya, lihat Elemen konfigurasi Siklus Hidup.

> **Important**
>
> Jika Anda memiliki beberapa aturan dalam konfigurasi Siklus Hidup S3, objek dapat memenuhi syarat untuk beberapa tindakan Siklus Hidup S3 pada hari yang sama. Dalam kasus tersebut, Amazon S3 mengikuti aturan umum ini:
>
> - Penghapusan permanen lebih diutamakan daripada transisi.

- Transisi lebih diutamakan daripada pembuatan penanda hapus.
- Saat sebuah objek memenuhi syarat untuk transisi S3 Glacier Flexible Retrieval dan S3 Standard-IA (atau S3 One Zone-IA), Amazon S3 memilih transisi S3 Glacier Flexible Retrieval.

Sebagai contoh, lihat Contoh filter yang tumpang tindih dan tindakan siklus hidup yang bertentangan.

Konfigurasi Siklus Hidup S3 pada bucket berkemampuan MFA

Konfigurasi Siklus Hidup S3 pada bucket otentikasi multi-faktor yang dikonfigurasi untuk penghapusan MFA tidak didukung. Untuk informasi selengkapnya, lihat Mengonfigurasi penghapusan MFA.

Siklus hidup S3 dan pencatatan

Tindakan Siklus Hidup Amazon S3 tidak ditangkap oleh AWS CloudTrail pencatatan tingkat objek. CloudTrailmenangkap permintaan API yang dibuat ke titik akhir Amazon S3 eksternal, sedangkan tindakan Siklus Hidup S3 dilakukan dengan menggunakan titik akhir Amazon S3 internal.

Anda dapat mengaktifkan log akses server Amazon S3 di bucket S3 untuk menangkap tindakan terkait S3 Lifecycle, seperti transisi objek ke kelas penyimpanan lain dan kedaluwarsa objek yang menghasilkan penghapusan permanen atau penghapusan logis. Untuk informasi selengkapnya, lihat the section called “Pencatatan akses server”.

Jika Anda telah mengaktifkan log di bucket Anda, log akses server Amazon S3 melaporkan hasil dari operasi berikut.

Log operasi	Deskripsi
`S3.EXPIRE.OBJECT`	Amazon S3 menghapus objek secara permanen karena tindakan siklus hidup. `Expiration`
`S3.CREATE.DELETEMARKER`	Amazon S3 secara logis menghapus versi saat ini dengan menambahkan penanda hapus di bucket berkemampuan versi.

Log operasi	Deskripsi
S3.TRANSITION_SIA.OBJECT	Amazon S3 mentransisikan objek ke kelas penyimpanan S3 Standard-IA.
S3.TRANSITION_ZIA.OBJECT	Amazon S3 mentransisikan objek ke kelas penyimpanan S3 Standard-IA.
S3.TRANSITION_INT.OBJECT	Amazon S3 mentransisikan objek ke kelas penyimpanan S3 Intelligent-Tiering.
S3.TRANSITION_GIR.OBJECT	Amazon S3 memulai transisi objek ke kelas penyimpanan S3 Glacier Instant Retrieval.
S3.TRANSITION.OBJECT	Amazon S3 memulai transisi objek ke kelas penyimpanan S3 Glacier Flexible Retrieval.
S3.TRANSITION_GDA.OBJECT	Amazon S3 memulai transisi objek ke kelas penyimpanan S3 Glacier Deep Archive.
S3.DELETE.UPLOAD	Amazon S3 membatalkan unggahan multipart yang tidak lengkap.

> **Note**
>
> Catatan log akses server Amazon S3 dikirimkan dengan upaya terbaik dan tidak dapat digunakan untuk penghitungan lengkap semua permintaan Amazon S3.

Mengonfigurasi pemberitahuan acara Siklus Hidup S3

Untuk menerima pemberitahuan saat Amazon S3 menghapus objek atau mentransisikannya ke kelas penyimpanan Amazon S3 lain sebagai hasil dari mengikuti aturan Siklus Hidup S3, Anda dapat mengatur pemberitahuan peristiwa Amazon S3.

Anda dapat menerima pemberitahuan untuk peristiwa Siklus Hidup S3 berikut:

- Peristiwa transisi — Dengan menggunakan jenis `s3:LifecycleTransition` peristiwa, Anda dapat menerima pemberitahuan saat objek dialihkan dari satu kelas penyimpanan Amazon S3 ke kelas penyimpanan Amazon S3 lainnya dengan konfigurasi Siklus Hidup S3.
- Peristiwa kedaluwarsa (hapus) - Dengan menggunakan jenis `LifecycleExpiration` acara, Anda dapat menerima pemberitahuan kapan pun Amazon S3 menghapus objek berdasarkan konfigurasi Siklus Hidup S3 Anda.

  Ada dua jenis acara kedaluwarsa:

  - Jenis `s3:LifecycleExpiration:Delete` acara akan memberi tahu Anda saat objek dalam bucket yang tidak berversi dihapus. `s3:LifecycleExpiration:Delete`juga memberi tahu Anda ketika versi objek dihapus secara permanen oleh konfigurasi Siklus Hidup S3.
  - Jenis `s3:LifecycleExpiration:DeleteMarkerCreated` peristiwa akan memberi tahu Anda saat Siklus Hidup S3 membuat penanda hapus setelah versi objek saat ini dalam bucket berversi dihapus. Siklus Hidup S3 menetapkan waktu pembuatan penanda hapus menjadi 00:00 UTC (tengah malam) hari ini. Waktu pembuatan ini mungkin berbeda dari waktu acara dalam `s3:LifecycleExpiration:DeleteMarkerCreated` pemberitahuan yang dikirim S3. Untuk informasi selengkapnya, lihat the section called “Menghapus versi objek”.

Amazon S3 dapat mempublikasikan pemberitahuan acara ke topik Amazon Simple Notification Service (Amazon SNS), antrian Amazon Simple Queue Service (Amazon SQS), atau fungsi. AWS Lambda Untuk informasi selengkapnya, lihat Notifikasi Peristiwa Amazon S3.

Untuk petunjuk tentang cara mengonfigurasi Pemberitahuan Acara Amazon S3, lihat Mengaktifkan notifikasi peristiwa menggunakan Amazon SQS, Amazon SNS, dan. AWS Lambda

Berikut ini adalah contoh pesan yang dikirimkan Amazon S3 untuk menerbitkan peristiwa `s3:LifecycleExpiration:Delete`. Untuk informasi selengkapnya, lihat the section called “Struktur pesan peristiwa”.

```
{
   "Records":[
      {
         "eventVersion":"2.3",
         "eventSource":"aws:s3",
         "awsRegion":"us-west-2",
         "eventTime":"1970-01-01T00:00:00.000Z",
         "eventName":"LifecycleExpiration:Delete",
         "userIdentity":{
```

```
            "principalId":"s3.amazonaws.com"
         },
         "requestParameters":{
            "sourceIPAddress":"s3.amazonaws.com"
         },
         "responseElements":{
            "x-amz-request-id":"C3D13FE58DE4C810",
            "x-amz-id-2":"FMyUVURIY8/IgAtTv8xRjskZQpcIZ9KG4V5Wp6S7S/
JRWeUWerMUE5JgHvANOjpD"
         },
         "s3":{
            "s3SchemaVersion":"1.0",
            "configurationId":"testConfigRule",
            "bucket":{
               "name":"amzn-s3-demo-bucket",
               "ownerIdentity":{
                  "principalId":"A3NL1KOZZKExample"
               },
               "arn":"arn:aws:s3:::amzn-s3-demo-bucket"
            },
            "object":{
               "key":"expiration/delete",
               "sequencer":"0055AED6DCD90281E5",
            }
         }
      }
   ]
}
```

Pesan yang dikirim Amazon S3 untuk mempublikasikan `s3:LifecycleTransition` acara juga menyertakan informasi berikut:

```
"lifecycleEventData":{
    "transitionEventData": {
        "destinationStorageClass": the destination storage class for the object
    }
}
```

Elemen konfigurasi Siklus Hidup

Konfigurasi Siklus Hidup S3 terdiri dari aturan Siklus Hidup yang menyertakan berbagai elemen yang menjelaskan tindakan yang dilakukan Amazon S3 selama dan masa pakai objek. Anda menetapkan

konfigurasi Siklus Hidup Amazon S3 sebagai XML, yang terdiri dari satu atau beberapa aturan Siklus Hidup, di mana setiap aturan terdiri dari satu atau beberapa elemen.

```
<LifecycleConfiguration>
    <Rule>
         <Element>
    </Rule>
    <Rule>
         <Element>
         <Element>
    </Rule>
</LifecycleConfiguration>
```

Setiap aturan terdiri atas hal berikut:

- Metadata aturan yang menyertakan ID aturan, dan status yang menunjukkan apakah aturan diaktifkan atau dinonaktifkan. Jika aturan dinonaktifkan, Amazon S3 tidak akan melakukan tindakan yang ditentukan dalam aturan tersebut.
- Filter yang mengidentifikasi objek yang aturan berlaku. Anda dapat menentukan filter dengan menggunakan ukuran objek, key prefix objek, satu atau beberapa tag objek, atau kombinasi filter.
- Satu atau beberapa tindakan transisi atau kedaluwarsa dengan tanggal atau periode waktu dalam masa pakai objek saat Anda ingin Amazon S3 melakukan tindakan tertentu.

Topik

- Elemen ID
- Elemen status
- Elemen filter
- Elemen untuk mendeskripsikan tindakan siklus hidup
- Menambahkan filter ke aturan Siklus Hidup

Bagian berikut menjelaskan elemen XML dalam konfigurasi Siklus Hidup S3. Untuk contoh konfigurasi, lihat Contoh konfigurasi Siklus Hidup S3.

Elemen ID

Konfigurasi Siklus Hidup S3 dapat memiliki hingga 1.000 aturan. Batas ini tidak dapat disesuaikan. `<ID>`Elemen secara unik mengidentifikasi aturan. Panjang ID dibatasi hingga 255 karakter.

Elemen status

Nilai `<Status>` elemen dapat berupa `Enabled` atau`Disabled`. Jika aturan dinonaktifkan, Amazon S3 tidak akan melakukan tindakan yang ditentukan dalam aturan tersebut.

Elemen filter

Aturan Siklus Hidup S3 dapat diterapkan ke semua atau subset objek dalam bucket berdasarkan `<Filter>` elemen yang Anda tentukan dalam aturan.

Anda dapat memfilter objek berdasarkan awalan kunci, tag objek, atau kombinasi keduanya (dalam hal ini Amazon S3 menggunakan `AND` logis untuk menggabungkan filter). Untuk contoh dan informasi selengkapnya tentang filter, lihatMenambahkan filter ke aturan Siklus Hidup.

- Menentukan filter dengan menggunakan awalan kunci — Contoh ini menunjukkan aturan Siklus Hidup S3 yang berlaku untuk subset objek berdasarkan awalan nama kunci (). `logs/` Misalnya, aturan Siklus Hidup berlaku untuk objek`logs/mylog.txt`,`logs/temp1.txt`, dan. `logs/test.txt` Aturan tidak berlaku untuk objek `example.jpg`.

  ```
  <LifecycleConfiguration>
      <Rule>
          <Filter>
             <Prefix>logs/</Prefix>
          </Filter>
          transition/expiration actions
           ...
      </Rule>
      ...
  </LifecycleConfiguration>
  ```

  Jika Anda ingin menerapkan tindakan siklus hidup ke subset objek berdasarkan awalan nama kunci yang berbeda, tentukan aturan terpisah. Pada setiap aturan, tentukan filter berbasis awalan. Misalnya, untuk mendeskripsikan tindakan siklus hidup objek dengan awalan kunci `projectA/` dan`projectB/`, Anda menentukan dua aturan sebagai berikut:

  ```
  <LifecycleConfiguration>
      <Rule>
          <Filter>
             <Prefix>projectA/</Prefix>
          </Filter>
          transition/expiration actions
  ```

```
        ...
    </Rule>

    <Rule>
        <Filter>
           <Prefix>projectB/</Prefix>
        </Filter>
        transition/expiration actions
        ...
    </Rule>
</LifecycleConfiguration>
```

Untuk informasi selengkapnya tentang kunci objek, lihat Menamai objek Amazon S3.

- Menentukan filter berdasarkan tag objek - Dalam contoh berikut, aturan Siklus Hidup menentukan filter berdasarkan tag (*key*) dan value (). *value* Aturan ini kemudian hanya berlaku untuk subset objek dengan tag tertentu.

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
           <Tag>
              <Key>key</Key>
              <Value>value</Value>
           </Tag>
        </Filter>
        transition/expiration actions
        ...
    </Rule>
</LifecycleConfiguration>
```

Anda dapat menentukan filter berdasarkan beberapa tag. Anda harus membungkus tag dalam <And> elemen, seperti yang ditunjukkan pada contoh berikut. Aturan tersebut mengarahkan Amazon S3 untuk melakukan tindakan siklus hidup pada objek dengan dua tag (dengan kunci dan nilai tag khusus).

```
<LifecycleConfiguration>
    <Rule>
      <Filter>
         <And>
            <Tag>
               <Key>key1</Key>
```

```
                <Value>value1</Value>
             </Tag>
             <Tag>
                <Key>key2</Key>
                <Value>value2</Value>
             </Tag>
              ...
          </And>
      </Filter>
      transition/expiration actions
    </Rule>
</Lifecycle>
```

Aturan Siklus Hidup berlaku untuk objek yang memiliki kedua tag yang ditentukan. Amazon S3 menjalankan AND logis. Perhatikan hal berikut:

- Setiap tag harus sama persis dengan kunci dan nilai. Jika Anda hanya menentukan <Key> elemen dan tidak ada <Value> elemen, aturan hanya akan berlaku untuk objek yang cocok dengan kunci tag dan yang tidak memiliki nilai yang ditentukan.
- Aturan ini berlaku untuk subset objek yang memiliki semua tag yang ditentukan dalam aturan. Jika sebuah objek memiliki tag tambahan yang ditentukan, aturan tersebut akan tetap berlaku.

> **Note**
>
> Ketika Anda menentukan beberapa tag di filter, setiap kunci tag harus unik.

- Menentukan filter berdasarkan awalan dan satu atau beberapa tag — Dalam aturan Siklus Hidup, Anda dapat menentukan filter berdasarkan awalan kunci dan satu atau beberapa tag. Sekali lagi, Anda harus membungkus semua elemen filter ini dalam <And> elemen, sebagai berikut:

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
          <And>
             <Prefix>key-prefix</Prefix>
             <Tag>
                <Key>key1</Key>
                <Value>value1</Value>
             </Tag>
             <Tag>
                <Key>key2</Key>
```

```
                <Value>value2</Value>
             </Tag>
              ...
          </And>
        </Filter>
        <Status>Enabled</Status>
        transition/expiration actions
    </Rule>
</LifecycleConfiguration>
```

Amazon S3 menggabungkan filter ini dengan menggunakan logika. AND Artinya, aturan berlaku untuk subset objek dengan key prefix yang ditentukan dan tag yang ditentukan. Filter hanya dapat memiliki satu awalan, dan nol atau lebih tag.

- Anda dapat menentukan filter kosong, yang dalam hal demikian aturan berlaku untuk semua objek dalam bucket.

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
        </Filter>
        <Status>Enabled</Status>
        transition/expiration actions
    </Rule>
</LifecycleConfiguration>
```

- Untuk memfilter aturan berdasarkan ukuran objek, Anda dapat menentukan ukuran minimum (`ObjectSizeGreaterThan`) atau ukuran maksimum (`ObjectSizeLessThan`), atau Anda dapat menentukan rentang ukuran objek.

  Besar objek dalam byte. Secara default, objek yang lebih kecil dari 128 KB tidak akan dialihkan ke kelas penyimpanan apa pun, kecuali jika Anda menentukan ukuran minimum (`ObjectSizeGreaterThan`) yang lebih kecil atau ukuran maksimum (`ObjectSizeLessThan`). Untuk informasi selengkapnya, lihat Contoh: Mengizinkan objek yang lebih kecil dari 128 KB untuk dialihkan.

```
                <LifecycleConfiguration>
    <Rule>
        <Filter>
            <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
        </Filter>
```

```
        <Status>Enabled</Status>
        transition/expiration actions
    </Rule>
</LifecycleConfiguration>
```

> **Note**
>
> ObjectSizeLessThanFilter ObjectSizeGreaterThan dan mengecualikan nilai yang ditentukan. Misalnya, jika Anda menyetel objek berukuran 128 KB hingga 1024 KB untuk berpindah dari kelas penyimpanan Standar S3 ke kelas penyimpanan IA Standar S3, objek yang persis 1024 KB dan 128 KB tidak akan beralih ke IA Standar S3. Sebagai gantinya, aturan hanya akan berlaku untuk objek yang lebih besar dari 128 KB dan berukuran kurang dari 1024 KB.

Jika Anda menentukan rentang ukuran objek, bilangan bulat ObjectSizeGreaterThan harus kurang dari nilai ObjectSizeLessThan. Saat menggunakan lebih dari satu filter, Anda harus membungkus filter dalam elemen <And>. Contoh berikut menunjukkan bagaimana menentukan objek dalam kisaran antara 500 byte dan 64.000 byte.

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
            <And>
                <Prefix>key-prefix</Prefix>
                <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
                <ObjectSizeLessThan>64000</ObjectSizeLessThan>
            </And>
        </Filter>
        <Status>Enabled</Status>
        transition/expiration actions
    </Rule>
</LifecycleConfiguration>
```

Elemen untuk mendeskripsikan tindakan siklus hidup

Anda dapat mengarahkan Amazon S3 untuk melakukan tindakan tertentu dalam masa pakai objek dengan menentukan satu atau beberapa tindakan yang telah ditentukan berikut dalam aturan Siklus Hidup S3. Efek tindakan ini tergantung pada keadaan Penentuan Versi bucket Anda.

- **`Transition`**elemen tindakan - Anda menentukan `Transition` tindakan untuk mengalihkan objek dari satu kelas penyimpanan ke kelas penyimpanan lainnya. Untuk informasi selengkapnya tentang objek transisi, lihat Transisi yang didukung. Saat tanggal atau periode waktu tertentu dalam masa pakai objek tercapai, Amazon S3 melakukan transisi.

  Untuk bucket berversi (bucket dengan dukungan Penentuan Versi atau dengan ditangguhkan), tindakan `Transition` berlaku untuk versi objek saat ini. Untuk mengelola versi lama, Amazon S3 menentukan tindakan `NoncurrentVersionTransition` (dijelaskan nanti dalam topik ini).

- **`Expiration`**elemen tindakan — `Expiration` Tindakan akan kedaluwarsa objek yang diidentifikasi dalam aturan dan berlaku untuk objek yang memenuhi syarat di salah satu kelas penyimpanan Amazon S3. Untuk informasi selengkapnya tentang kelas penyimpanan, lihat Memahami dan mengelola kelas penyimpanan Amazon S3. Amazon S3 membuat semua objek kedaluwarsa tidak tersedia. Apakah objek secara permanen dihapus tergantung pada status Penentuan Versi bucket.
  - Bucket nonversioned — `Expiration` Tindakan ini menghasilkan Amazon S3 menghapus objek secara permanen.
  - Bucket berversi–Untuk bucket berversi (yaitu, dengan dukungan Penentuan Versi atau dengan Penentuan Versi ditangguhkan), ada beberapa pertimbangan yang memandu cara Amazon S3 menangani tindakan `Expiration`. Untuk bucket dengan dukungan Penentuan Versi atau Penentuan Versi ditangguhkan, berlaku berikut ini:
    - Tindakan `Expiration` hanya berlaku untuk versi saat ini (tidak memiliki dampak pada versi objek lama).
    - Amazon S3 tidak mengambil tindakan apa pun jika ada satu atau beberapa versi objek dan penanda hapus adalah versi saat ini.
    - Jika versi objek saat ini adalah satu-satunya versi objek dan juga merupakan penanda hapus (juga disebut sebagai penanda hapus objek kedaluwarsa, ketika semua versi objek dihapus dan Anda hanya memiliki penanda hapus yang tersisa), Amazon S3 menghapus penanda hapus objek kedaluwarsa. Anda juga dapat menggunakan tindakan kedaluwarsa guna mengarahkan Amazon S3 untuk menghapus penanda hapus objek yang kedaluwarsa.

Sebagai contoh, lihat Menghapus penanda penghapusan objek yang kedaluwarsa dalam bucket berkemampuan versi.

Untuk informasi selengkapnya, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.

Pertimbangkan juga hal berikut saat mengatur Amazon S3 untuk mengelola kedaluwarsa:

- Bucket dengan dukungan Penentuan Versi

  Jika versi objek saat ini bukan penanda hapus, Amazon S3 menambahkan penanda hapus dengan ID versi unik. Ini membuat versi saat ini menjadi versi lama, dan penanda hapus menjadi versi saat ini.

- Bucket dengan Penentuan Versi ditangguhkan

  Dalam bucket yang ditangguhkan versi, tindakan kedaluwarsa menyebabkan Amazon S3 membuat penanda hapus dengan ID versi. `null` Penanda hapus ini menggantikan versi objek apa pun dengan ID versi null dalam hierarki versi, yang secara efektif menghapus objek.

Selain itu, Amazon S3 memberikan tindakan berikut yang dapat Anda gunakan untuk mengelola versi objek lama dalam bucket berversi (yaitu, dengan dukungan Penentuan Versi dan Penentuan Versi ditangguhkan).

- **`NoncurrentVersionTransition`**elemen tindakan - Gunakan tindakan ini untuk menentukan kapan Amazon S3 mentransisikan objek ke kelas penyimpanan yang ditentukan. Anda dapat mendasarkan transisi ini pada sejumlah hari tertentu sejak objek menjadi noncurrent (`<NoncurrentDays>`). Selain jumlah hari, Anda juga dapat menentukan jumlah versi noncurrent (`<NewerNoncurrentVersions>`) yang akan dipertahankan (antara 1 dan 100). Nilai ini menentukan berapa banyak versi noncurrent yang lebih baru yang harus ada sebelum Amazon S3 dapat mentransisikan versi tertentu. Amazon S3 akan mentransisikan versi noncurrent tambahan apa pun di luar nomor yang ditentukan untuk dipertahankan. Agar transisi terjadi, nilai `<NoncurrentDays>` dan `<NewerNoncurrentVersions>` nilai harus dilampaui.

  Untuk menentukan jumlah versi noncurrent yang akan dipertahankan, Anda juga harus menyediakan `<Filter>` elemen. Jika Anda tidak menentukan `<Filter>` elemen, Amazon S3 menghasilkan `InvalidRequest` kesalahan saat Anda menentukan jumlah versi noncurrent yang akan dipertahankan.

Untuk informasi selengkapnya tentang objek transisi, lihat Transisi yang didukung. Untuk detail tentang cara Amazon S3 menghitung tanggal saat Anda menentukan jumlah hari dalam tindakan `NoncurrentVersionTransition`, lihat Aturan siklus hidup: Berdasarkan usia objek.

- **`NoncurrentVersionExpiration`**elemen tindakan - Gunakan tindakan ini untuk mengarahkan Amazon S3 untuk menghapus versi objek yang tidak aktif secara permanen. Objek yang dihapus ini tidak dapat dipulihkan. Anda dapat mendasarkan kedaluwarsa ini pada sejumlah hari tertentu sejak objek menjadi noncurrent (). `<NoncurrentDays>` Selain jumlah hari, Anda juga dapat menentukan jumlah versi noncurrent (`<NewerNoncurrentVersions>`) yang akan dipertahankan (antara 1 dan 100). Nilai ini menentukan berapa banyak versi noncurrent yang lebih baru yang harus ada sebelum Amazon S3 dapat kedaluwarsa versi tertentu. Amazon S3 akan secara permanen menghapus versi noncurrent tambahan di luar nomor yang ditentukan untuk disimpan. Agar penghapusan terjadi, nilai `<NoncurrentDays>` dan `<NewerNoncurrentVersions>` nilai harus dilampaui.

  Untuk menentukan jumlah versi noncurrent yang akan dipertahankan, Anda juga harus menyediakan `<Filter>` elemen. Jika Anda tidak menentukan `<Filter>` elemen, Amazon S3 menghasilkan `InvalidRequest` kesalahan saat Anda menentukan jumlah versi noncurrent yang akan dipertahankan.

  Penghapusan objek lama yang tertunda ini dapat membantu ketika Anda perlu memperbaiki penghapusan atau penghapusan yang tidak disengaja. Misalnya, Anda dapat mengonfigurasi aturan kedaluwarsa untuk menghapus versi lama lima hari setelah menjadi versi lama. Misalnya, pada 1/1/2014 pukul 10:30 UTC, Anda membuat objek yang disebut `photo.gif` (versi ID 111111). Pada 1/2/2014 pukul 11:30 UTC, Anda secara tidak sengaja menghapus `photo.gif` (ID versi 111111), yang membuat penanda hapus dengan ID versi baru (seperti ID versi 4857693). Sekarang Anda memiliki waktu lima hari untuk memulihkan versi asli `photo.gif` (ID versi 111111) sebelum penghapusan bersifat permanen. Pada 1/8/2014 pukul 00:00 UTC, aturan Siklus Hidup untuk kedaluwarsa berjalan dan dihapus secara permanen `photo.gif` (ID versi 111111), lima hari setelah menjadi versi noncurrent.

  Untuk detail tentang cara Amazon S3 menghitung tanggal saat Anda menentukan jumlah hari dalam tindakan `NoncurrentVersionExpiration`, lihat Aturan siklus hidup: Berdasarkan usia objek.

> **Note**
>
> Konfigurasi siklus hidup kedaluwarsa objek tidak menghapus unggahan multibagian yang tidak lengkap. Untuk menghapus unggahan multibagian yang tidak lengkap, Anda harus menggunakan tindakan konfigurasi `AbortIncompleteMultipartUpload` Siklus Hidup yang dijelaskan nanti di bagian ini.

Selain tindakan transisi dan kedaluwarsa, Anda dapat menggunakan tindakan konfigurasi Siklus Hidup berikut untuk mengarahkan Amazon S3 agar menghentikan unggahan multibagian yang tidak lengkap atau menghapus penanda penghapusan objek kedaluwarsa:

- **`AbortIncompleteMultipartUpload`**elemen tindakan - Gunakan elemen ini untuk mengatur waktu maksimum (dalam beberapa hari) yang Anda inginkan agar unggahan multipart tetap berlangsung. Jika unggahan multibagian yang berlaku (ditentukan oleh nama kunci yang `prefix` ditentukan dalam aturan Siklus Hidup) tidak berhasil diselesaikan dalam jangka waktu yang telah ditentukan, Amazon S3 menghentikan unggahan multibagian yang tidak lengkap. Untuk informasi selengkapnya, lihat Membatalkan unggahan multibagian.

  > **Note**
  >
  > Anda tidak dapat menentukan tindakan siklus hidup ini dalam aturan yang memiliki filter yang menggunakan tag objek.

- **`ExpiredObjectDeleteMarker`**elemen aksi — Dalam bucket berkemampuan versi, penanda hapus dengan nol versi noncurrent disebut sebagai penanda penghapusan objek kedaluwarsa. Anda dapat menggunakan tindakan siklus hidup ini untuk mengarahkan Amazon S3 untuk menghapus penanda penghapusan objek yang kedaluwarsa. Sebagai contoh, lihat Menghapus penanda penghapusan objek yang kedaluwarsa dalam bucket berkemampuan versi.

  > **Note**
  >
  > Anda tidak dapat menentukan tindakan siklus hidup ini dalam aturan yang memiliki filter yang menggunakan tag objek.

Cara Amazon S3 menghitung berapa lama sebuah objek menjadi versi lama

Pada bucket dengan dukungan Penentuan Versi, Anda dapat memiliki beberapa versi objek. Selalu ada satu versi saat ini, dan nol atau beberapa versi lama. Setiap kali Anda mengunggah sebuah objek, versi saat ini disimpan sebagai versi lama dan versi baru yang ditambahkan, penerusnya, menjadi versi saat ini. Untuk menentukan jumlah hari saat suatu objek menjadi versi lama, Amazon S3 melihat kapan penerusnya dibuat. Amazon S3 menggunakan jumlah hari sejak penerusnya dibuat sejak jumlah hari suatu objek menjadi versi lama.

> **Memulihkan versi sebelumnya dari objek saat menggunakan konfigurasi Siklus Hidup S3**
>
> Seperti yang dijelaskan dalamMemulihkan versi sebelumnya, Anda dapat menggunakan salah satu dari dua metode berikut untuk mengambil versi objek sebelumnya:
>
> - Metode 1 — Salin versi objek yang tidak terkini ke dalam ember yang sama. Objek yang disalin menjadi versi saat ini dari objek tersebut dan semua versi objek dipertahankan.
> - Metode 2 — Hapus versi objek saat ini secara permanen. Ketika Anda menghapus versi objek saat ini, Anda, akibatnya, mengubah versi lama ke versi saat ini dari objek tersebut.
>
> Saat Anda menggunakan aturan konfigurasi Siklus Hidup S3 dengan bucket berkemampuan versi, sebaiknya gunakan Metode 1 sebagai praktik terbaik.
> Siklus hidup S3 beroperasi di bawah model yang akhirnya konsisten. Versi saat ini yang Anda hapus secara permanen mungkin tidak hilang sampai perubahan menyebar ke semua sistem Amazon S3. (Oleh karena itu, Amazon S3 mungkin untuk sementara tidak mengetahui penghapusan ini.) Sementara itu, aturan siklus hidup yang Anda konfigurasikan untuk mengakhiri objek lama mungkin secara permanen menghapus objek lama, termasuk objek yang ingin Anda pulihkan. Jadi, menyalin versi lama, seperti yang direkomendasikan dalam Metode 1, adalah alternatif yang lebih aman.

Tindakan Siklus Hidup dan kondisi Penentuan Versi bucket

Aturan siklus hidup: Berdasarkan usia objek

Anda dapat menentukan periode waktu, dalam jumlah hari sejak pembuatan (atau modifikasi) objek, ketika Amazon S3 dapat mengambil tindakan yang ditentukan.

Saat Anda menentukan jumlah hari dalam tindkaan `Transition` dan `Expiration` dalam konfigurasi Siklus Hidup S3, perhatikan hal berikut:

- Nilai yang Anda tentukan adalah jumlah hari sejak pembuatan objek saat tindakan akan terjadi.
- Amazon S3 menghitung waktu dengan menambahkan jumlah hari yang ditentukan dalam aturan ke waktu pembuatan objek dan membulatkan waktu yang dihasilkan ke hari berikutnya pada tengah malam UTC. Misalnya, jika sebuah objek dibuat pada 1/15/2014 pukul 10:30 UTC dan Anda menentukan 3 hari dalam aturan transisi, maka tanggal transisi objek akan dihitung sebagai 1/19/2014 00:00 UTC.

> **Note**
>
> Amazon S3 hanya mempertahankan tanggal modifikasi terakhir untuk setiap objek. Misalnya, konsol Amazon S3 menunjukkan tanggal modifikasi terakhir di panel Properti objek. Ketika Anda awalnya membuat objek baru, tanggal ini mencerminkan tanggal bahwa objek dibuat. Jika Anda mengganti objek, tanggal akan berubah sesuai dengan perubahan tersebut. Oleh karena itu, tanggal pembuatan identik dengan Tanggal modifikasi terakhir.

Saat menentukan jumlah hari pada tindakan `NoncurrentVersionTransition` dan `NoncurrentVersionExpiration` dalam konfigurasi Siklus Hidup, perhatikan hal berikut:

- Nilai yang Anda tentukan adalah jumlah hari sejak versi objek menjadi noncurrent (yaitu, ketika objek ditimpa atau dihapus) bahwa Amazon S3 akan melakukan tindakan pada objek atau objek yang ditentukan.
- Amazon S3 menghitung waktu dengan menambahkan jumlah hari yang ditentukan dalam aturan ke waktu ketika versi penerus baru dari objek dibuat dan membulatkan waktu yang dihasilkan ke hari berikutnya pada tengah malam UTC. Misalnya, di bucket Anda, misalkan Anda memiliki versi objek saat ini yang dibuat pada 1/1/2014 pukul 10:30 UTC. Jika versi baru dari objek yang menggantikan versi saat ini dibuat pada 1/15/2014 pukul 10:30 UTC, dan Anda menentukan 3 hari dalam aturan transisi, tanggal transisi objek dihitung sebagai 1/19/2014 00:00 UTC.

Aturan siklus hidup: Berdasarkan tanggal tertentu

Saat menentukan tindakan dalam aturan Siklus Hidup S3, Anda dapat menentukan tanggal kapan Anda ingin Amazon S3 untuk mengambil tindakan. Ketika tanggal tertentu tiba, Amazon S3 menerapkan tindakan ke semua objek yang memenuhi syarat (berdasarkan kriteria filter).

Jika Anda menentukan tindakan Siklus Hidup S3 dengan tanggal sebelumnya, semua objek yang memenuhi syarat akan segera memenuhi syarat untuk tindakan siklus hidup tersebut.

> **Important**
>
> Tindakan berbasis tanggal bukanlah tindakan satu kali. Amazon S3 terus menerapkan tindakan berbasis tanggal bahkan setelah tanggal berlalu, selama status aturan adalah `Enabled`.
> Misalnya, Anda menentukan `Expiration` tindakan berbasis tanggal untuk menghapus semua objek (asumsikan bahwa tidak ada filter yang ditentukan dalam aturan). Pada tanggal yang ditentukan, Amazon S3 menjadikan mengakhiri semua objek di dalam bucket. Amazon S3 juga terus kedaluwarsa objek baru yang Anda buat di ember. Untuk menghentikan tindakan siklus hidup, Anda harus menghapus tindakan dari aturan siklus hidup, menonaktifkan aturan, atau menghapus aturan dari konfigurasi siklus hidup.

Nilai tanggal harus sesuai dengan format ISO 8601. Waktunya adalah selalu tengah malam UTC.

> **Note**
>
> Anda tidak dapat membuat aturan Siklus Hidup berbasis tanggal menggunakan konsol Amazon S3, tetapi Anda dapat melihat, menonaktifkan, atau menghapus aturan tersebut.

Menambahkan filter ke aturan Siklus Hidup

Filter adalah elemen aturan Lifecyle opsional yang dapat Anda gunakan untuk menentukan objek mana yang diterapkan aturan tersebut.

Elemen-elemen berikut dapat digunakan untuk menyaring objek:

Awalan kunci

Anda dapat memfilter objek berdasarkan awalan. Jika Anda ingin menerapkan tindakan siklus hidup ke subset objek dengan awalan berbeda, buat aturan terpisah untuk setiap tindakan.

Tag objek

Anda dapat memfilter objek berdasarkan satu atau beberapa tag. Setiap tag harus cocok dengan kunci dan nilai dengan tepat, dan, jika Anda menentukan beberapa tag, setiap kunci tag harus unik. Filter dengan beberapa tag objek berlaku untuk subset objek yang memiliki semua tag yang ditentukan. Jika suatu objek memiliki tag tambahan yang ditentukan, filter akan tetap berlaku.

> **Note**
>
> Jika Anda hanya menentukan `Key` elemen dan tidak ada `Value` elemen, aturan hanya akan berlaku untuk objek yang cocok dengan kunci tag dan yang tidak memiliki nilai yang ditentukan.

Ukuran objek minimum atau maksimum

Anda dapat memfilter objek berdasarkan ukuran. Anda dapat menentukan ukuran minimum (`ObjectSizeGreaterThan`) atau ukuran maksimum (`ObjectSizeLessThan`), atau Anda dapat menentukan rentang ukuran objek dalam filter yang sama. Besar objek dalam byte. Ukuran filter maksimum adalah 5 TB. Amazon S3 menerapkan ukuran objek minimum default ke konfigurasi siklus hidup. Untuk informasi selengkapnya, lihat Contoh: Mengizinkan objek yang lebih kecil dari 128 KB untuk dialihkan.

Anda dapat menggabungkan elemen filter yang berbeda dalam hal ini Amazon S3 menggunakan logika. `AND`

Contoh filter

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan elemen filter yang berbeda:

- Menentukan filter dengan menggunakan awalan kunci — Contoh ini menunjukkan aturan Siklus Hidup S3 yang berlaku untuk subset objek berdasarkan awalan nama kunci (). `logs/` Misalnya, aturan Siklus Hidup berlaku untuk objek`logs/mylog.txt`,`logs/temp1.txt`, dan. `logs/test.txt` Aturan tidak berlaku untuk objek `example.jpg`.

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
           <Prefix>logs/</Prefix>
        </Filter>
        transition/expiration actions
         ...
    </Rule>
    ...
</LifecycleConfiguration>
```

> **Note**
>
> Jika Anda memiliki satu atau lebih awalan yang dimulai dengan karakter yang sama, Anda dapat menyertakan semua awalan tersebut dalam aturan Anda dengan menentukan awalan parsi tanpa garis miring () di filter. / Misalnya, anggaplah Anda memiliki awalan ini:
>
> ```
> sales1999/
>                 sales2000/
>                 sales2001/
> ```
>
> Untuk menyertakan ketiga awalan dalam aturan Anda, tentukan `sales` sebagai awalan dalam aturan siklus hidup Anda.

Jika Anda ingin menerapkan tindakan siklus hidup ke subset objek berdasarkan awalan nama kunci yang berbeda, tentukan aturan terpisah. Pada setiap aturan, tentukan filter berbasis awalan. Misalnya, untuk mendeskripsikan tindakan siklus hidup objek dengan awalan kunci `projectA/` dan`projectB/`, Anda menentukan dua aturan sebagai berikut:

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
           <Prefix>projectA/</Prefix>
        </Filter>
        transition/expiration actions
         ...
    </Rule>

    <Rule>
        <Filter>
           <Prefix>projectB/</Prefix>
        </Filter>
        transition/expiration actions
         ...
    </Rule>
</LifecycleConfiguration>
```

Untuk informasi selengkapnya tentang kunci objek, lihat Menamai objek Amazon S3.

- Menentukan filter berdasarkan tag objek - Dalam contoh berikut, aturan Siklus Hidup menentukan filter berdasarkan tag (*key*) dan value (). *value* Aturan ini kemudian hanya berlaku untuk subset objek dengan tag tertentu.

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
           <Tag>
              <Key>key</Key>
              <Value>value</Value>
           </Tag>
        </Filter>
        transition/expiration actions
        ...
    </Rule>
</LifecycleConfiguration>
```

Anda dapat menentukan filter berdasarkan beberapa tag. Anda harus membungkus tag dalam <And> elemen, seperti yang ditunjukkan pada contoh berikut. Aturan tersebut mengarahkan Amazon S3 untuk melakukan tindakan siklus hidup pada objek dengan dua tag (dengan kunci dan nilai tag khusus).

```
<LifecycleConfiguration>
    <Rule>
      <Filter>
         <And>
            <Tag>
               <Key>key1</Key>
               <Value>value1</Value>
            </Tag>
            <Tag>
               <Key>key2</Key>
               <Value>value2</Value>
            </Tag>
             ...
          </And>
      </Filter>
      transition/expiration actions
    </Rule>
</Lifecycle>
```

Aturan Siklus Hidup berlaku untuk objek yang memiliki kedua tag yang ditentukan. Amazon S3 menjalankan AND logis. Perhatikan hal berikut:

- Setiap tag harus sama persis dengan kunci dan nilai. Jika Anda hanya menentukan <Key> elemen dan tidak ada <Value> elemen, aturan hanya akan berlaku untuk objek yang cocok dengan kunci tag dan yang tidak memiliki nilai yang ditentukan.
- Aturan ini berlaku untuk subset objek yang memiliki semua tag yang ditentukan dalam aturan. Jika sebuah objek memiliki tag tambahan yang ditentukan, aturan tersebut akan tetap berlaku.

> **Note**
>
> Ketika Anda menentukan beberapa tag di filter, setiap kunci tag harus unik.

- Menentukan filter berdasarkan awalan dan satu atau beberapa tag — Dalam aturan Siklus Hidup, Anda dapat menentukan filter berdasarkan awalan kunci dan satu atau beberapa tag. Sekali lagi, Anda harus membungkus semua elemen filter ini dalam <And> elemen, sebagai berikut:

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
          <And>
             <Prefix>key-prefix</Prefix>
             <Tag>
                <Key>key1</Key>
                <Value>value1</Value>
             </Tag>
             <Tag>
                <Key>key2</Key>
                <Value>value2</Value>
             </Tag>
              ...
          </And>
        </Filter>
        <Status>Enabled</Status>
        transition/expiration actions
    </Rule>
</LifecycleConfiguration>
```

Amazon S3 menggabungkan filter ini dengan menggunakan logika. AND Artinya, aturan berlaku untuk subset objek dengan key prefix yang ditentukan dan tag yang ditentukan. Filter hanya dapat memiliki satu awalan, dan nol atau lebih tag.

- Menentukan filter kosong — Anda dapat menentukan filter kosong, dalam hal ini aturan berlaku untuk semua objek dalam ember.

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
        </Filter>
        <Status>Enabled</Status>
        transition/expiration actions
    </Rule>
</LifecycleConfiguration>
```

- > Menentukan filter ukuran objek - Untuk memfilter aturan berdasarkan ukuran objek, Anda dapat menentukan ukuran minimum (`ObjectSizeGreaterThan`) atau ukuran maksimum (`ObjectSizeLessThan`), atau Anda dapat menentukan berbagai ukuran objek.

  Besar objek dalam byte. Ukuran filter maksimum adalah 5 TB. Beberapa kelas penyimpanan memiliki batasan ukuran objek minimum. Untuk informasi selengkapnya, lihat Membandingkan kelas penyimpanan Amazon S3.

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
            <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
        </Filter>
        <Status>Enabled</Status>
        transition/expiration actions
    </Rule>
</LifecycleConfiguration>
```

> **Note**
>
> `ObjectSizeLessThan`Filter `ObjectSizeGreaterThan` dan mengecualikan nilai yang ditentukan. Misalnya, jika Anda menyetel objek berukuran 128 KB hingga 1024 KB untuk berpindah dari kelas penyimpanan Standar S3 ke kelas penyimpanan IA Standar S3, objek yang persis 1024 KB dan 128 KB tidak akan beralih ke IA Standar S3. Sebagai gantinya,

aturan hanya akan berlaku untuk objek yang lebih besar dari 128 KB dan berukuran kurang dari 1024 KB.

Jika Anda menentukan rentang ukuran objek, bilangan bulat `ObjectSizeGreaterThan` harus kurang dari nilai `ObjectSizeLessThan`. Saat menggunakan lebih dari satu filter, Anda harus membungkus filter dalam elemen <And>. Contoh berikut menunjukkan bagaimana menentukan objek dalam kisaran antara 500 byte dan 64.000 byte.

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
            <And>
                <Prefix>key-prefix</Prefix>
                <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
                <ObjectSizeLessThan>64000</ObjectSizeLessThan>
            </And>
        </Filter>
        <Status>Enabled</Status>
        transition/expiration actions
    </Rule>
</LifecycleConfiguration>
```

Bagaimana Amazon S3 menangani konflik dalam konfigurasi siklus hidup

Umumnya, Siklus hidup Amazon S3 mengoptimalkan biaya. Misalnya, jika dua kebijakan kedaluwarsa tumpang tindih, kebijakan kedaluwarsa yang lebih pendek akan dipenuhi sehingga data tidak disimpan lebih lama dari yang diharapkan. Demikian pula, jika dua kebijakan transisi tumpang tindih, Siklus Hidup S3 mentransisi objek Anda ke kelas penyimpanan berbiaya rendah.

Dalam kedua kasus tersebut, Siklus Hidup S3 mencoba memilih jalur yang paling murah bagi Anda. Pengecualian untuk aturan umum ini adalah dengan kelas penyimpanan S3 Intelligent-Tiering. S3 Intelligent-Tiering disukai oleh S3 Lifecycle dibandingkan kelas penyimpanan apa pun, selain kelas penyimpanan S3 Glacier dan S3 Glacier Deep Archive.

Jika Anda memiliki beberapa aturan dalam konfigurasi Siklus Hidup S3, objek dapat memenuhi syarat untuk beberapa tindakan Siklus Hidup S3 pada hari yang sama. Dalam kasus tersebut, Amazon S3 mengikuti aturan umum ini:

- Penghapusan permanen lebih diutamakan daripada transisi.
- Transisi lebih diutamakan daripada pembuatan penanda hapus.
- Ketika objek memenuhi syarat untuk transisi S3 Glacier Flexible Retrieval dan S3 Standard-IA (atau S3 One Zone-IA) transisi, Amazon S3 memilih transisi S3 Glacier Flexible Retrieval.

Contoh filter yang tumpang tindih dan tindakan siklus hidup yang bertentangan

Anda dapat menentukan konfigurasi Siklus Hidup S3 yang kemudian Anda menentukan tindakan, atau awalan yang tumpang tindih. Contoh berikut menunjukkan cara Amazon S3 menyelesaikan potensi konflik.

Example 1: Awalan tumpang-tindih (tidak ada konflik)

Konfigurasi contoh berikut memiliki dua aturan yang menetapkan awalan yang tumpang tindih sebagai berikut:

- Aturan pertama menetapkan filter kosong, yang menunjukkan semua objek dalam bucket.
- Aturan kedua menetapkan awalan nama kunci (`logs/`), yang hanya menunjukkan subset objek.

Aturan 1 meminta Amazon S3 menghapus semua objek satu tahun setelah pembuatan. Aturan 2 meminta Amazon S3 mentransisikan subset objek ke kelas penyimpanan S3 Standard-IA 30 hari setelah pembuatan.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Rule 1</ID>
    <Filter>
    </Filter>
    <Status>Enabled</Status>
    <Expiration>
      <Days>365</Days>
    </Expiration>
  </Rule>
  <Rule>
    <ID>Rule 2</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
```

```
    <Transition>
      <StorageClass>STANDARD_IA</StorageClass>
      <Days>30</Days>
    </Transition>
   </Rule>
</LifecycleConfiguration>
```

Karena tidak ada konflik dalam kasus ini, Amazon S3 akan mentransisikan objek dengan awalan `logs/` ke kelas penyimpanan S3 Standard-IA 30 hari setelah pembuatan. Ketika objek apa pun mencapai satu tahun setelah pembuatan, itu akan dihapus.

Example 2: Tindakan siklus hidup yang bertentangan

Dalam konfigurasi contoh ini, ada dua aturan yang mengarahkan Amazon S3 untuk melakukan dua tindakan berbeda pada set objek yang sama dalam masa aktif objek:

- Kedua aturan menentukan awalan nama kunci yang sama, sehingga kedua aturan berlaku pada objek yang sama.
- Kedua aturan menentukan 365 hari yang sama setelah pembuatan objek saat aturan berlaku.
- Satu aturan mengarahkan Amazon S3 untuk mentransisi objek ke kelas penyimpanan S3 Standard-IA dan aturan lain ingin Amazon S3 mengakhiri objek pada saat yang sama.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Rule 1</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Expiration>
      <Days>365</Days>
    </Expiration>
  </Rule>
  <Rule>
    <ID>Rule 2</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <StorageClass>STANDARD_IA</StorageClass>
```

```
      <Days>365</Days>
    </Transition>
   </Rule>
</LifecycleConfiguration>
```

Dalam kasus ini, karena Anda ingin objek kedaluwarsa (dihapus), tidak ada artinya mengubah kelas penyimpanan, jadi Amazon S3 memilih tindakan kedaluwarsa pada objek tersebut.

Example 3: Awalan tumpang-tindih yang mengakibatkan tindakan siklus hidup yang bertentangan

Dalam contoh ini, konfigurasi memiliki dua aturan, yang menetapkan awalan tumpang tindih sebagai berikut:

- Aturan 1 menetapkan awalan kosong (menunjukkan semua objek).
- Aturan 2 menetapkan awalan nama kunci (`logs/`) yang mengidentifikasi bagian dari semua objek.

Untuk subset objek dengan awalan nama kunci `logs/`, tindakan Siklus Hidup S3 dalam kedua aturan berlaku. Satu aturan mengarahkan Amazon S3 transisi objek 10 hari setelah pembuatan, dan aturan lain mengarahkan Amazon S3 mentransisi objek 365 hari setelah pembuatan.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Rule 1</ID>
    <Filter>
      <Prefix></Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <StorageClass>STANDARD_IA</StorageClass>
      <Days>10</Days>
    </Transition>
  </Rule>
  <Rule>
    <ID>Rule 2</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <StorageClass>STANDARD_IA</StorageClass>
      <Days>365</Days>
```

```
    </Transition>
   </Rule>
</LifecycleConfiguration>
```

Dalam kasus ini, Amazon S3 memilih untuk melakukan transisi 10 hari setelah pembuatan.

Example 4: Pemfilteran berbasis tag dan tindakan siklus hidup bertentangan yang diakibatkan

Anggaplah Anda memiliki konfigurasi Siklus Hidup S3 berikut yang memiliki dua aturan, masing-masing menetapkan filter tag:

- Aturan 1 menetapkan filter berbasis tag (`tag1/value1`). Aturan ini mengarahkan Amazon S3 mentransisi objek ke kelas penyimpanan S3 Glacier Flexible Retrieval 365 hari setelah pembuatan.
- Aturan 2 menetapkan filter berbasis tag (`tag2/value2`). Aturan ini mengarahkan Amazon S3 untuk mengakhiri objek 14 hari setelah pembuatan.

Konfigurasi Siklus Hidup S3 ditampilkan dalam contoh berikut.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Rule 1</ID>
    <Filter>
      <Tag>
         <Key>tag1</Key>
         <Value>value1</Value>
      </Tag>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <StorageClass>GLACIER</StorageClass>
      <Days>365</Days>
    </Transition>
  </Rule>
  <Rule>
    <ID>Rule 2</ID>
    <Filter>
      <Tag>
         <Key>tag2</Key>
         <Value>value2</Value>
      </Tag>
    </Filter>
    <Status>Enabled</Status>
```

```
    <Expiration>
      <Days>14</Days>
    </Expiration>
   </Rule>
</LifecycleConfiguration>
```

Jika suatu objek memiliki kedua tag, Amazon S3 harus memutuskan aturan mana yang harus diikuti. Dalam kasus ini, Amazon S3 mengakhiri objek 14 hari setelah pembuatan. Objek dihapus, dan oleh karena itu tindakan transisi tidak berlaku.

Contoh konfigurasi Siklus Hidup S3

Bagian ini memberikan contoh konfigurasi Siklus Hidup S3. Setiap contoh menunjukkan bagaimana Anda dapat menentukan XML dalam setiap contoh skenario.

Topik

- Mengarsipkan semua objek dalam satu hari setelah pembuatan
- Menonaktifkan aturan Siklus Hidup untuk sementara
- Menjajarkan kelas penyimpanan selama masa pakai objek
- Menentukan beberapa aturan
- Menentukan aturan siklus aktif untuk bucket yang diaktifkan versioning-nya
- Menghapus penanda penghapusan objek yang kedaluwarsa dalam bucket berkemampuan versi
- Konfigurasi Lifecycle untuk membatalkan unggahan multibagian
- Objek noncurrent yang kedaluwarsa yang tidak memiliki data
- Contoh: Mengizinkan objek yang lebih kecil dari 128 KB untuk dialihkan

Mengarsipkan semua objek dalam satu hari setelah pembuatan

Setiap aturan Siklus Hidup S3 mencakup filter yang dapat Anda gunakan untuk mengidentifikasi bagian objek dalam bucket Anda yang berlaku aturan Siklus Hidup S3. Konfigurasi Siklus Hidup S3 berikut menunjukkan contoh cara menentukan filter.

- Dalam aturan konfigurasi Siklus Hidup S3 ini, filter menentukan awalan kunci (`tax/`). Oleh karena itu, aturan berlaku untuk objek dengan awalan nama kunci `tax/`, seperti `tax/doc1.txt` dan `tax/doc2.txt`.

Aturan tersebut menetapkan dua tindakan yang mengarahkan Amazon S3 untuk melakukan hal berikut:

- Transisi objek ke kelas penyimpanan S3 Glacier Flexible Retrieval 365 hari (satu tahun) setelah pembuatan.
- Menghapus objek (tindakan `Expiration`) 3.650 hari (10 tahun) setelah pembuatan.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Transition and Expiration Rule</ID>
    <Filter>
       <Prefix>tax/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>365</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
    <Expiration>
      <Days>3650</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

Alih-alih menentukan usia objek dalam hal hari setelah pembuatan, Anda dapat menentukan tanggal untuk setiap tindakan. Namun, Anda tidak dapat menggunakan `Date` dan `Days` dalam aturan yang sama.

- Jika Anda ingin aturan Siklus Hidup S3 berlaku pada semua objek dalam bucket, tentukan awalan kosong. Dalam konfigurasi berikut, aturan menentukan tindakan `Transition` yang mengarahkan Amazon S3 untuk melakukan transisi objek ke kelas penyimpanan S3 Glacier Flexible Retrieval 0 hari setelah pembuatan. Aturan ini berarti bahwa objek memenuhi syarat untuk arsip ke S3 Glacier Flexible Retrieval pada tengah malam UTC setelah pembuatan. Untuk informasi selengkapnya tentang batasan siklus hidup, lihat Kendala dan pertimbangan untuk transisi.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Archive all object same-day upon creation</ID>
    <Filter>
      <Prefix></Prefix>
    </Filter>
```

```
    <Status>Enabled</Status>
    <Transition>
      <Days>0</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

- Anda dapat menentukan nol atau satu awalan nama kunci dan nol atau lebih tag objek pada suatu filter. Kode contoh berikut ini menerapkan aturan Siklus Hidup S3 pada subset objek dengan awalan kunci `tax/` dan untuk objek yang memiliki dua tag dengan kunci dan nilai spesifik. Saat Anda menentukan lebih dari satu filter, Anda harus menyertakan elemen <And> seperti yang ditunjukkan (Amazon S3 menerapkan AND logis untuk menggabungkan kondisi filter yang ditentukan).

```
...
<Filter>
   <And>
      <Prefix>tax/</Prefix>
      <Tag>
         <Key>key1</Key>
         <Value>value1</Value>
      </Tag>
      <Tag>
         <Key>key2</Key>
         <Value>value2</Value>
      </Tag>
    </And>
</Filter>
...
```

- Anda dapat memfilter objek berdasarkan tag saja. Misalnya, aturan Siklus Hidup S3 berikut berlaku untuk objek yang memiliki dua label yang ditentukan (tidak menyebutkan awalan apa pun).

```
...
<Filter>
   <And>
      <Tag>
         <Key>key1</Key>
         <Value>value1</Value>
      </Tag>
```

```
      <Tag>
         <Key>key2</Key>
         <Value>value2</Value>
      </Tag>
    </And>
</Filter>
...
```

> **Important**
>
> Jika Anda memiliki beberapa aturan dalam konfigurasi Siklus Hidup S3, objek dapat memenuhi syarat untuk beberapa tindakan Siklus Hidup S3 pada hari yang sama. Dalam kasus tersebut, Amazon S3 mengikuti aturan umum ini:
>
> - Penghapusan permanen lebih diutamakan daripada transisi.
> - Transisi lebih diutamakan daripada pembuatan penanda hapus.
> - Saat objek memenuhi syarat untuk transisi S3 Glacier Flexible Retrieval dan S3 Standard-IA (atau S3 One Zone-IA), Amazon S3 memilih transisi Pengambilan Fleksibel S3 Glacier.
>
> Sebagai contoh, lihat Contoh filter yang tumpang tindih dan tindakan siklus hidup yang bertentangan.

Menonaktifkan aturan Siklus Hidup untuk sementara

Anda dapat menonaktifkan sementara aturan Siklus Hidup S3 menggunakan elemen. `status` Ini dapat berguna jika Anda ingin menguji aturan baru atau memecahkan masalah dengan konfigurasi Anda, tanpa menimpa aturan yang ada. Konfigurasi Siklus Hidup S3 berikut menetapkan dua aturan:

- Aturan 1 mengarahkan Amazon S3 untuk melakukan transisi objek dengan awalan `logs/` ke kelas penyimpanan S3 Glacier Flexible Retrieval segera setelah pembuatan.
- Aturan 2 mengarahkan Amazon S3 untuk melakukan transisi objek dengan awalan `documents/` ke kelas penyimpanan S3 Glacier Flexible Retrieval segera setelah pembuatan.

Dalam konfigurasi, Aturan 1 diaktifkan dan Aturan 2 dinonaktifkan. Amazon S3 mengabaikan aturan yang dinonaktifkan.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Rule1</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>0</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
  </Rule>
  <Rule>
    <ID>Rule2</ID>
    <Filter>
      <Prefix>documents/</Prefix>
    </Filter>
    <Status>Disabled</Status>
    <Transition>
      <Days>0</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

Menjajarkan kelas penyimpanan selama masa pakai objek

Dalam contoh ini, Anda menggunakan konfigurasi Siklus Hidup S3 untuk menurunkan kelas penyimpanan objek selama masa aktifnya. Penurunan tingkat dapat membantu mengurangi biaya penyimpanan. Untuk informasi selengkapnya tentang harga, lihat Harga Amazon S3.

Konfigurasi Siklus Hidup S3 berikut ini menentukan aturan yang berlaku untuk objek dengan awalan nama kunci `logs/`. Aturan menetapkan tindakan berikut ini:

- Dua tindakan transisi:
  - Transisi objek ke kelas penyimpanan S3 Standard-IA 30 hari setelah pembuatan.
  - Transisi objek ke kelas penyimpanan S3 Glacier Flexible Retrieval 90 hari setelah pembuatan.

- Satu tindakan kedaluwarsa yang mengarahkan Amazon S3 untuk menghapus objek setahun setelah pembuatan.

```
<LifecycleConfiguration>
  <Rule>
    <ID>example-id</ID>
    <Filter>
       <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>30</Days>
      <StorageClass>STANDARD_IA</StorageClass>
    </Transition>
    <Transition>
      <Days>90</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
    <Expiration>
      <Days>365</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

Note

Anda dapat menggunakan satu aturan untuk menjelaskan semua tindakan Siklus Hidup S3 jika semua tindakan berlaku pada set objek yang sama (diidentifikasi oleh filter). Jika tidak, Anda dapat menambahkan beberapa aturan dengan masing-masing menetapkan filter yang berbeda.

Important

Jika Anda memiliki beberapa aturan dalam konfigurasi Siklus Hidup S3, objek dapat memenuhi syarat untuk beberapa tindakan Siklus Hidup S3 pada hari yang sama. Dalam kasus tersebut, Amazon S3 mengikuti aturan umum ini:

- Penghapusan permanen lebih diutamakan daripada transisi.

- Transisi lebih diutamakan daripada pembuatan penanda hapus.
- Saat objek memenuhi syarat untuk transisi S3 Glacier Flexible Retrieval dan S3 Standard-IA (atau S3 One Zone-IA), Amazon S3 memilih transisi Pengambilan Fleksibel S3 Glacier.

Sebagai contoh, lihat Contoh filter yang tumpang tindih dan tindakan siklus hidup yang bertentangan.

Menentukan beberapa aturan

Anda dapat menentukan beberapa aturan jika Anda menginginkan tindakan Siklus Hidup S3 yang berbeda dari objek yang berbeda. Konfigurasi Siklus Hidup S3 berikut memiliki dua aturan:

- Aturan 1 berlaku untuk objek dengan awalan nama kunci `classA/`. Aturan ini mengarahkan Amazon S3 untuk melakukan transisi objek ke kelas penyimpanan S3 Glacier Flexible Retrieval satu tahun setelah pembuatan dan mengakhiri objek ini 10 tahun setelah pembuatan.
- Aturan 2 berlaku untuk objek dengan awalan nama kunci `classB/`. Aturan ini mengarahkan Amazon S3 untuk melakukan transisi objek ke kelas penyimpanan S3 Standard-IA 90 hari setelah pembuatan dan menghapusnya satu tahun setelah pembuatan.

```
<LifecycleConfiguration>
    <Rule>
        <ID>ClassADocRule</ID>
        <Filter>
           <Prefix>classA/</Prefix>
        </Filter>
        <Status>Enabled</Status>
        <Transition>
           <Days>365</Days>
           <StorageClass>GLACIER</StorageClass>
        </Transition>
        <Expiration>
             <Days>3650</Days>
        </Expiration>
    </Rule>
    <Rule>
        <ID>ClassBDocRule</ID>
        <Filter>
```

```
            <Prefix>classB/</Prefix>
        </Filter>
        <Status>Enabled</Status>
        <Transition>
           <Days>90</Days>
           <StorageClass>STANDARD_IA</StorageClass>
        </Transition>
        <Expiration>
             <Days>365</Days>
        </Expiration>
    </Rule>
</LifecycleConfiguration>
```

Important

Jika Anda memiliki beberapa aturan dalam konfigurasi Siklus Hidup S3, objek dapat memenuhi syarat untuk beberapa tindakan Siklus Hidup S3 pada hari yang sama. Dalam kasus tersebut, Amazon S3 mengikuti aturan umum ini:

- Penghapusan permanen lebih diutamakan daripada transisi.
- Transisi lebih diutamakan daripada pembuatan penanda hapus.
- Saat objek memenuhi syarat untuk transisi S3 Glacier Flexible Retrieval dan S3 Standard-IA (atau S3 One Zone-IA), Amazon S3 memilih transisi Pengambilan Fleksibel S3 Glacier.

Sebagai contoh, lihat Contoh filter yang tumpang tindih dan tindakan siklus hidup yang bertentangan.

Menentukan aturan siklus aktif untuk bucket yang diaktifkan versioning-nya

Misalkan Anda memiliki bucket dengan dukungan Penentuan Versi, yang berarti untuk setiap objek Anda memiliki versi saat ini dan nol atau lebih versi lama. (Untuk informasi selengkapnya tentang Penentuan Versi S3, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.)

Dalam contoh berikut, Anda ingin mempertahankan sejarah satu tahun, dan mempertahankan 5 versi tidak terkini. Konfigurasi Siklus Hidup S3 mendukung penyimpanan 1 hingga 100 versi objek apa pun. Ketahuilah bahwa lebih dari 5 versi noncurrent yang lebih baru harus ada sebelum Amazon S3 dapat kedaluwarsa versi tertentu. Amazon S3 akan secara permanen menghapus versi

noncurrent tambahan di luar nomor yang ditentukan untuk disimpan. Agar penghapusan terjadi, nilai `NoncurrentDays` dan `NewerNoncurrentVersions` nilai harus dilampaui.

Untuk menghemat biaya penyimpanan, Anda ingin memindahkan versi lama ke S3 Glacier Flexible Retrieval 30 hari setelah menjadi versi lama (dengan asumsi objek lama ini adalah data dingin yang tidak memerlukan akses waktu nyata). Selain itu, Anda mengharapkan frekuensi akses versi saat ini berkurang 90 hari setelah pembuatan, jadi Anda dapat memilih untuk memindahkan objek ini ke kelas penyimpanan IA Standar S3.

```
<LifecycleConfiguration>
    <Rule>
        <ID>sample-rule</ID>
        <Filter>
           <Prefix></Prefix>
        </Filter>
        <Status>Enabled</Status>
        <Transition>
           <Days>90</Days>
           <StorageClass>STANDARD_IA</StorageClass>
        </Transition>
        <NoncurrentVersionTransition>
            <NoncurrentDays>30</NoncurrentDays>
            <StorageClass>GLACIER</StorageClass>
        </NoncurrentVersionTransition>
       <NoncurrentVersionExpiration>
            <NewerNoncurrentVersions>5</NewerNoncurrentVersions>
            <NoncurrentDays>365</NoncurrentDays>
       </NoncurrentVersionExpiration>
    </Rule>
</LifecycleConfiguration>
```

Menghapus penanda penghapusan objek yang kedaluwarsa dalam bucket berkemampuan versi

Bucket dengan dukungan Penentuan Versi memiliki satu versi saat ini dan nol atau lebih versi lama untuk setiap objek. Ketika Anda menghapus sebuah objek, perhatikan hal berikut ini:

- Jika Anda tidak menentukan ID versi dalam permintaan penghapusan Anda, Amazon S3 menambahkan penanda hapus alih-alih menghapus objek. Versi objek saat ini menjadi versi lama, dan penanda hapus menjadi versi saat ini.

- Jika Anda menentukan ID versi dalam permintaan penghapusan, Amazon S3 akan menghapus versi objek secara permanen (penanda hapus tidak dibuat).
- Penanda hapus dengan nol versi lama disebut sebagai penanda hapus objek kedaluwarsa.

Contoh ini menunjukkan skenario yang dapat membuat penanda hapus objek kedaluwarsa dalam bucket Anda, dan cara Anda dapat menggunakan konfigurasi Siklus Hidup S3 untuk mengarahkan Amazon S3 untuk menghapus penanda hapus objek kedaluwarsa.

Misalkan Anda menulis konfigurasi Siklus Hidup S3 yang menggunakan `NoncurrentVersionExpiration` tindakan untuk menghapus versi noncurrent 30 hari setelah menjadi noncurrent dan mempertahankan 10 versi noncurrent, seperti yang ditunjukkan pada contoh berikut. Ketahuilah bahwa lebih dari 10 versi noncurrent yang lebih baru harus ada sebelum Amazon S3 dapat kedaluwarsa versi tertentu. Amazon S3 akan secara permanen menghapus versi noncurrent tambahan di luar nomor yang ditentukan untuk disimpan. Agar penghapusan terjadi, nilai `NoncurrentDays` dan `NewerNoncurrentVersions` nilai harus dilampaui.

```
<LifecycleConfiguration>
    <Rule>
        ...
        <NoncurrentVersionExpiration>
            <NewerNoncurrentVersions>10</NewerNoncurrentVersions>
            <NoncurrentDays>30</NoncurrentDays>
        </NoncurrentVersionExpiration>
    </Rule>
</LifecycleConfiguration>
```

`NoncurrentVersionExpiration`Tindakan tidak berlaku untuk versi objek saat ini. Ini hanya menghapus versi lama.

Untuk versi objek saat ini, Anda memiliki opsi berikut untuk mengelola masa aktifnya, tergantung pada apakah versi objek saat ini mengikuti siklus hidup yang ditetapkan dengan baik:

- Versi objek saat ini mengikuti siklus hidup yang ditentukan dengan baik.

  Dalam kasus ini, Anda dapat menggunakan konfigurasi Siklus Hidup S3 dengan tindakan `Expiration` guna mengarahkan Amazon S3 untuk menghapus versi saat ini, seperti yang ditunjukkan pada contoh berikut.

  ```
  <LifecycleConfiguration>
  ```

```
    <Rule>
        ...
        <Expiration>
           <Days>60</Days>
        </Expiration>
        <NoncurrentVersionExpiration>
            <NewerNoncurrentVersions>10</NewerNoncurrentVersions>
            <NoncurrentDays>30</NoncurrentDays>
        </NoncurrentVersionExpiration>
    </Rule>
</LifecycleConfiguration>
```

Dalam contoh ini, Amazon S3 menghapus versi saat ini 60 hari setelah dibuat dengan menambahkan penanda hapus untuk setiap versi objek saat ini. Proses ini membuat versi saat ini menjadi lama, dan penanda hapus menjadi versi saat ini. Untuk informasi selengkapnya, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.

> **Note**
>
> Anda tidak dapat menentukan tag `Days` dan `ExpiredObjectDeleteMarker` tag pada aturan yang sama. Saat menentukan tag `Days`, Amazon S3 secara otomatis melakukan pembersihan `ExpiredObjectDeleteMarker` saat penanda hapus sudah cukup tua untuk memenuhi kriteria usia. Untuk membersihkan penanda hapus segera setelah menjadi satu-satunya versi, buat aturan terpisah hanya dengan tag `ExpiredObjectDeleteMarker`.

Tindakan `NoncurrentVersionExpiration` dalam konfigurasi Siklus Hidup S3 yang sama akan menghapus objek lama 30 hari setelah menjadi lama. Dengan demikian, dalam contoh ini, semua versi objek dihapus secara permanen 90 hari setelah pembuatan objek. Ketahuilah bahwa dalam contoh ini, lebih dari 10 versi noncurrent yang lebih baru harus ada sebelum Amazon S3 dapat kedaluwarsa versi tertentu. Amazon S3 akan secara permanen menghapus versi noncurrent tambahan di luar nomor yang ditentukan untuk disimpan. Agar penghapusan terjadi, nilai `NoncurrentDays` dan `NewerNoncurrentVersions` nilai harus dilampaui.

Meskipun penanda hapus objek kedaluwarsa dibuat selama proses ini, Amazon S3 mendeteksi dan menghapus penanda hapus objek kedaluwarsa untuk Anda.

- Versi objek saat ini tidak memiliki siklus hidup yang didefinisikan dengan baik.

Dalam kasus ini, Anda dapat menghapus objek secara manual ketika Anda tidak memerlukannya, membuat penanda hapus dengan satu atau lebih versi lama. Jika konfigurasi Siklus Hidup S3 dengan tindakan `NoncurrentVersionExpiration` menghapus semua versi lama, sekarang Anda memiliki penanda hapus objek kedaluwarsa.

Khusus untuk skenario ini, konfigurasi Siklus Hidup S3 menyediakan tindakan `Expiration` yang dapat Anda gunakan untuk menghapus penanda hapus objek kedaluwarsa.

```
<LifecycleConfiguration>
    <Rule>
       <ID>Rule 1</ID>
        <Filter>
          <Prefix>logs/</Prefix>
        </Filter>
        <Status>Enabled</Status>
        <Expiration>
           <ExpiredObjectDeleteMarker>true</ExpiredObjectDeleteMarker>
        </Expiration>
        <NoncurrentVersionExpiration>
            <NewerNoncurrentVersions>10</NewerNoncurrentVersions>
            <NoncurrentDays>30</NoncurrentDays>
        </NoncurrentVersionExpiration>
    </Rule>
</LifecycleConfiguration>
```

Dengan mengatur elemen `ExpiredObjectDeleteMarker` ke `true` dalam tindakan `Expiration`, Anda mengarahkan Amazon S3 untuk menghapus penanda hapus objek kedaluwarsa.

> **Note**
>
> Saat Anda menggunakan tindakan Siklus Hidup S3 `ExpiredObjectDeleteMarker`, aturan tidak dapat menentukan filter berbasis tag.

Konfigurasi Lifecycle untuk membatalkan unggahan multibagian

Anda dapat menggunakan operasi API REST unggahan multibagian Amazon S3 untuk mengunggah objek besar dalam beberapa bagian. Untuk informasi lebih lanjut tentang unggahan multibagian, lihat Mengunggah dan menyalin objek menggunakan unggahan multipart di Amazon S3.

Dengan menggunakan konfigurasi Siklus Hidup S3, Anda dapat mengarahkan Amazon S3 untuk menghentikan unggahan multibagian yang tidak lengkap (diidentifikasi dengan awalan nama kunci yang ditentukan dalam aturan) jika tidak selesai dalam jumlah hari tertentu setelah inisiasi. Saat Amazon S3 membatalkan unggahan multibagian, Amazon S3 menghapus semua bagian yang terkait dengan unggahan multibagian. Proses ini membantu mengontrol biaya penyimpanan dengan memastikan bahwa Anda tidak memiliki unggahan multibagian yang belum selesai dengan bagian-bagian yang disimpan di Amazon S3.

> **Note**
>
> Saat Anda menggunakan tindakan Siklus Hidup S3 `AbortIncompleteMultipartUpload`, aturan tidak dapat menentukan filter berbasis tag.

Berikut ini adalah contoh konfigurasi Siklus Hidup S3 yang menentukan aturan dengan tinndakan `AbortIncompleteMultipartUpload`. Tindakan ini mengarahkan Amazon S3 untuk menghentikan pengunggahan multibagian yang tidak selesai tujuh hari setelah inisiasi.

```
<LifecycleConfiguration>
    <Rule>
        <ID>sample-rule</ID>
        <Filter>
           <Prefix>SomeKeyPrefix/</Prefix>
        </Filter>
        <Status>rule-status</Status>
        <AbortIncompleteMultipartUpload>
          <DaysAfterInitiation>7</DaysAfterInitiation>
        </AbortIncompleteMultipartUpload>
    </Rule>
</LifecycleConfiguration>
```

Objek noncurrent yang kedaluwarsa yang tidak memiliki data

Anda dapat membuat aturan yang mentransisikan objek hanya berdasarkan ukurannya. Anda dapat menentukan ukuran minimum (`ObjectSizeGreaterThan`) atau ukuran maksimum (`ObjectSizeLessThan`), atau Anda dapat menentukan rentang ukuran objek dalam bita. Saat menggunakan lebih dari satu filter, seperti awalan dan aturan ukuran, Anda harus membungkus filter dalam elemen `<And>`.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Transition with a prefix and based on size</ID>
    <Filter>
       <And>
          <Prefix>tax/</Prefix>
          <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
       </And>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>365</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

Jika Anda menentukan rentang dengan menggunakan kedua elemen `ObjectSizeGreaterThan` dan `ObjectSizeLessThan`, ukuran objek maksimum harus lebih besar dari ukuran objek minimum. Saat menggunakan lebih dari satu filter, Anda harus membungkus filter dalam elemen `<And>`. Contoh berikut menunjukkan bagaimana menentukan objek dalam kisaran antara 500 byte dan 64.000 byte. Saat Anda menentukan rentang, `ObjectSizeLessThan` filter `ObjectSizeGreaterThan` dan mengecualikan nilai yang ditentukan. Untuk informasi selengkapnya, lihat the section called “Elemen filter”.

```
<LifecycleConfiguration>
    <Rule>
        ...
          <And>
             <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
             <ObjectSizeLessThan>64000</ObjectSizeLessThan>
          </And>
    </Rule>
```

```
</LifecycleConfiguration>
```

Anda juga dapat membuat aturan untuk secara khusus menghapus objek noncurrent yang tidak memiliki data, termasuk objek penanda hapus noncurrent yang dibuat dalam bucket berkemampuan versi. Contoh berikut menggunakan `NoncurrentVersionExpiration` tindakan untuk menghapus versi noncurrent 30 hari setelah menjadi noncurrent dan mempertahankan 10 versi noncurrent. Contoh ini juga menggunakan `ObjectSizeLessThan` elemen untuk memfilter hanya objek tanpa data.

Ketahuilah bahwa lebih dari 10 versi noncurrent yang lebih baru harus ada sebelum Amazon S3 dapat kedaluwarsa versi tertentu. Amazon S3 akan secara permanen menghapus versi noncurrent tambahan di luar nomor yang ditentukan untuk disimpan. Agar penghapusan terjadi, nilai `NoncurrentDays` dan `NewerNoncurrentVersions` nilai harus dilampaui.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Expire noncurrent with size less than 1 byte</ID>
    <Filter>
       <ObjectSizeLessThan>1</ObjectSizeLessThan>
    </Filter>
    <Status>Enabled</Status>
    <NoncurrentVersionExpiration>
       <NewerNoncurrentVersions>10</NewerNoncurrentVersions>
       <NoncurrentDays>30</NoncurrentDays>
    </NoncurrentVersionExpiration>
  </Rule>
</LifecycleConfiguration>
```

Contoh: Mengizinkan objek yang lebih kecil dari 128 KB untuk dialihkan

Amazon S3 menerapkan perilaku default ke konfigurasi Siklus Hidup Anda yang mencegah objek yang lebih kecil dari 128 KB dialihkan ke kelas penyimpanan apa pun. Anda dapat mengizinkan objek yang lebih kecil untuk bertransisi dengan menambahkan filter ukuran minimum (`ObjectSizeGreaterThan`) atau ukuran maksimum (`ObjectSizeLessThan`) yang menentukan ukuran yang lebih kecil ke konfigurasi. Contoh berikut memungkinkan objek yang lebih kecil dari 128 KB untuk transisi ke kelas penyimpanan S3 Glacier Instant Retrieval:

```
<LifecycleConfiguration>
  <Rule>
    <ID>Allow small object transitions</ID>
```

```
    <Filter>
          <ObjectSizeGreaterThan>1</ObjectSizeGreaterThan>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>365</Days>
      <StorageClass>GLACIER_IR</StorageClass>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

Note

Pada bulan September 2024, Amazon S3 memperbarui perilaku transisi default untuk objek kecil, sebagai berikut:

- Perilaku transisi default yang diperbarui — Mulai September 2024, perilaku default mencegah objek yang lebih kecil dari 128 KB dialihkan ke kelas penyimpanan apa pun.
- Perilaku transisi default sebelumnya — Sebelum September 2024, perilaku default memungkinkan objek yang lebih kecil dari 128 KB untuk dialihkan hanya ke kelas penyimpanan S3 Glacier dan S3 Glacier Deep Archive.

Konfigurasi yang dibuat sebelum September 2024 mempertahankan perilaku transisi sebelumnya kecuali Anda memodifikasinya. Artinya, jika Anda membuat, mengedit, atau menghapus aturan, perilaku transisi default untuk konfigurasi Anda akan berubah menjadi perilaku yang diperbarui. Jika kasus penggunaan diperlukan, Anda dapat mengubah perilaku transisi default sehingga objek yang lebih kecil dari 128KB akan bertransisi ke S3 Glacier dan S3 Glacier Deep Archive. Untuk melakukan ini, gunakan `x-amz-transition-object-size-minimum-default` header opsional dalam PutBucketLifecycleConfigurationpermintaan.

Contoh berikut menunjukkan bagaimana menggunakan `x-amz-transition-object-size-minimum-default` header dalam PutBucketLifecycleConfigurationmeminta untuk menerapkan perilaku transisi `varies_by_storage_class` default ke konfigurasi Siklus Hidup S3. Perilaku ini memungkinkan objek yang lebih kecil dari 128 KB untuk bertransisi ke kelas penyimpanan S3 Glacier atau S3 Glacier Deep Archive. Secara default, semua kelas penyimpanan lainnya akan mencegah transisi yang lebih kecil dari 128 KB. Anda masih dapat menggunakan filter khusus

untuk mengubah ukuran transisi minimum untuk kelas penyimpanan apa pun. Filter khusus selalu diutamakan daripada perilaku transisi default:

```
HTTP/1.1 200
x-amz-transition-object-size-minimum-default: varies_by_storage_class
<?xml version="1.0" encoding="UTF-8"?>
...
```

Memecahkan masalah Siklus Hidup Amazon S3

Informasi berikut dapat membantu Anda memecahkan masalah umum pada aturan Siklus Hidup S3.

Topik

- Saya menjalankan operasi daftar di bucket saya dan melihat objek yang menurut saya sudah kedaluwarsa atau ditransisikan oleh aturan siklus hidup.
- Bagaimana cara memantau tindakan yang diambil oleh aturan siklus hidup saya?
- Jumlah objek S3 saya masih meningkat, bahkan setelah menyiapkan aturan siklus hidup pada bucket yang mendukung versi.
- Bagaimana cara mengosongkan bucket S3 saya dengan menggunakan aturan siklus hidup?
- Tagihan Amazon S3 saya meningkat setelah mentransisikan objek ke kelas penyimpanan berbiaya lebih rendah.
- Saya telah memperbarui kebijakan bucket saya, tetapi objek S3 saya masih dihapus oleh aturan siklus hidup yang kedaluwarsa.
- Dapatkah saya memulihkan objek S3 yang dikahiri oleh aturan Siklus Hidup S3?
- Mengapa tindakan kedaluwarsa dan siklus hidup transisi saya tidak terjadi?
- Bagaimana cara mengecualikan awalan dari aturan siklus hidup saya?
- Bagaimana saya bisa menyertakan beberapa awalan dalam aturan siklus hidup saya?

Saya menjalankan operasi daftar di bucket saya dan melihat objek yang menurut saya sudah kedaluwarsa atau ditransisikan oleh aturan siklus hidup.

Transisi objek Siklus Hidup S3 dan waktu kedaluwarsa objek adalah operasi asinkron. Oleh karena itu, dapat terjadi penundaan antara waktu objek memenuhi syarat untuk kedaluwarsa atau transisi dan waktu ketika objek tersebut benar-benar ditransisikan atau kedaluwarsa. Perubahan dalam penagihan diterapkan segera setelah aturan siklus hidup terpenuhi, bahkan jika tindakan tidak

selesai. Pengecualian untuk perilaku ini adalah jika Anda memiliki aturan siklus hidup yang disetel ke kelas penyimpanan S3 Intelligent-Tiering. Dalam hal ini, perubahan penagihan tidak terjadi hingga objek telah ditransisikan ke S3 Intelligent-Tiering. Untuk informasi selengkapnya tentang perubahan dalam penagihan, lihat Menyetel konfigurasi siklus hidup pada bucket.

> **Note**
>
> Amazon S3 tidak mentransisikan objek yang lebih kecil dari 128 KB dari kelas penyimpanan S3 Standard atau S3 Standard-IA ke kelas penyimpanan S3 Intelligent-Tiering, S3 Standard-IA, atau S3 One Zone-IA.

Bagaimana cara memantau tindakan yang diambil oleh aturan siklus hidup saya?

Untuk memantau tindakan yang diambil oleh aturan siklus hidup, Anda dapat menggunakan fitur berikut:

- Pemberitahuan Acara S3 - Anda dapat mengatur Pemberitahuan Acara S3 sehingga Anda diberi tahu tentang kedaluwarsa atau peristiwa transisi Siklus Hidup S3.
- Log akses server S3 — Anda dapat mengaktifkan log akses server untuk bucket S3 Anda untuk menangkap tindakan Siklus Hidup S3, seperti transisi objek ke kelas penyimpanan lain atau kedaluwarsa objek. Untuk informasi lebih lanjut, lihat Siklus hidup dan pencatatan.

Untuk melihat perubahan dalam penyimpanan yang disebabkan oleh tindakan siklus hidup setiap hari, sebaiknya gunakan dasbor Lensa Penyimpanan S3 alih-alih menggunakan metrik Amazon. CloudWatch Di dasbor Lensa Penyimpanan, Anda dapat melihat metrik berikut, yang memantau jumlah atau ukuran objek:

- Byte versi terkini
- Jumlah objek versi saat ini
- Byte bukan versi terkini
- Jumlah objek versi tidak terkini
- Hapus Jumlah Objek Penanda
- Hapus byte penyimpanan penanda
- Byte unggahan multibagian yang tidak lengkap
- Jumlah objek unggahan multibagian yang tidak lengkap

Jumlah objek S3 saya masih meningkat, bahkan setelah menyiapkan aturan siklus hidup pada bucket yang mendukung versi.

Dalam bucket berkemampuan versi, saat objek kedaluwarsa, objek tersebut tidak sepenuhnya dihapus dari bucket. Sebagai gantinya, penanda hapus dibuat sebagai versi terbaru dari objek. Hapus penanda masih dihitung sebagai objek. Oleh karena itu, jika aturan siklus hidup dibuat agar hanya mengakhiri versi saat ini, maka jumlah objek di bucket S3 sebenarnya akan meningkat dan bukan turun.

Misalnya, bucket S3 dengan Penentuan Versi yang diaktifkan memiliki 100 objek, dan aturan siklus hidup diatur agar mengakhiri versi objek saat ini setelah 7 hari. Setelah hari ketujuh, jumlah objek meningkat menjadi 200 karena 100 penanda hapus dibuat selain dari 100 objek yang asli, yang sekarang menjadi versi tidak terkini. Untuk informasi selengkapnya tentang tindakan aturan konfigurasi Siklus Hidup S3 untuk bucket dengan Penentuan Versi diaktifkan, lihat Mengatur konfigurasi siklus hidup pada bucket.

Untuk menghapus objek secara permanen, tambahkan konfigurasi siklus hidup tambahan untuk menghapus versi objek sebelumnya, penanda hapus kedaluwarsa, dan unggahan multibagian yang tidak lengkap. Untuk petunjuk tentang cara membuat aturan siklus hidup baru, lihat Mengatur konfigurasi siklus hidup pada bucket.

> **Note**
>
> - Amazon S3 membulatkan transisi atau tanggal kedaluwarsa suatu objek ke tengah malam UTC pada hari berikutnya.
>
> Saat mengevaluasi objek untuk tindakan siklus hidup, Amazon S3 menggunakan waktu pembuatan objek di UTC. Misalnya, pertimbangkan bucket nonversioned dengan aturan siklus hidup yang dikonfigurasi untuk objek kedaluwarsa setelah satu hari. Misalkan sebuah objek dibuat pada 1 Januari pukul 17:05 Pacific Daylight Time (PDT), yang sesuai dengan 2 Januari pukul 00:05 UTC. Objek menjadi berumur satu hari pada 00:05 UTC pada 3 Januari, yang membuatnya memenuhi syarat untuk kedaluwarsa ketika S3 Lifecycle mengevaluasi objek pada 00:00 UTC pada 4 Januari.
>
> Karena tindakan siklus hidup Amazon S3 terjadi secara asinkron, mungkin ada beberapa penundaan antara tanggal yang ditentukan dalam aturan siklus hidup dan transisi fisik objek yang sebenarnya. Untuk informasi selengkapnya, lihat Transisi atau penundaan kedaluwarsa.

Untuk informasi lebih lanjut, lihat Aturan siklus hidup: Berdasarkan usia objek.

- Untuk objek S3 yang dilindungi oleh Kunci Objek, versi saat ini tidak dihapus secara permanen. Sebagai gantinya, penanda hapus ditambahkan ke objek, membuatnya menjadi tidak terkini. Versi tidak terkini kemudian dipertahankan dan tidak kedaluwarsa secara permanen.

Bagaimana cara mengosongkan bucket S3 saya dengan menggunakan aturan siklus hidup?

Aturan Siklus Hidup S3 adalah alat yang efektif untuk mengosongkan bucket S3 dengan jutaan objek. Untuk menghapus sejumlah besar objek dari bucket S3 Anda, pastikan untuk menggunakan dua pasang aturan siklus hidup ini:

- Akhiri versi objek saat ini dan Hapus secara permanen versi objek sebelumnya
- Hapus penanda hapus kedaluwarsa dan Hapus unggahan multibagian yang tidak lengkap

Untuk langkah-langkah cara membuat aturan konfigurasi siklus hidup baru, lihat Mengatur konfigurasi siklus hidup pada bucket.

> **Note**
>
> Untuk objek S3 yang dilindungi oleh Kunci Objek, versi saat ini tidak dihapus secara permanen. Sebagai gantinya, penanda hapus ditambahkan ke objek, membuatnya menjadi tidak terkini. Versi tidak terkini kemudian dipertahankan dan tidak kedaluwarsa secara permanen.

Tagihan Amazon S3 saya meningkat setelah mentransisikan objek ke kelas penyimpanan berbiaya lebih rendah.

Ada beberapa alasan mengapa tagihan Anda mungkin meningkat setelah mentransisikan objek ke kelas penyimpanan berbiaya lebih rendah:

- Biaya overhead S3 Glacier untuk objek-objek kecil

Untuk setiap objek yang ditransisikan ke S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive, total overhead sebesar 40 KB akan dikaitkan dengan pembaruan penyimpanan ini. Sebagai bagian dari overhead 40 KB, 8 KB digunakan untuk menyimpan metadata dan nama objek. 8 KB ini dibebankan sesuai dengan tarif Standar S3. 32 KB sisanya digunakan untuk pengindeksan dan metadata terkait. 32 KB ini dikenakan biaya sesuai dengan harga S3 Glacier Flexible Retrieval atau S3 Glacier Deep Archive.

Oleh karena itu, jika Anda menyimpan banyak objek berukuran lebih kecil, kami tidak menyarankan menggunakan transisi siklus hidup. Sebagai gantinya, untuk mengurangi biaya tambahan, pertimbangkan mengumpulkan banyak objek kecil menjadi lebih kecil sebelum menyimpannya di Amazon S3. Untuk informasi lebih lanjut tentang pertimbangan biaya, lihat Transisi ke kelas penyimpanan (pengarsipan objek) S3 Glacier Flexible Retrieval dan S3 Glacier Deep Archive.

- Biaya penyimpanan minimum

  Beberapa kelas penyimpanan S3 memiliki persyaratan durasi penyimpanan minimum. Objek yang dihapus, ditimpa, atau ditransisikan dari kelas-kelas tersebut sebelum durasi minimum dipenuhi dikenakan biaya transisi awal atau penghapusan prorata. Persyaratan durasi penyimpanan minimum ini adalah sebagai berikut:

  - S3 Standard-IA dan S3 One Zone-IA-30 hari
  - S3 Glacier Flexible Retrieval dan S3 Glacier Instant Retrieval-90 hari
  - S3 Glacier Deep Archive-180 hari

  Untuk informasi selengkapnya tentang persyaratan ini, lihat bagian Batasan dari Objek transisi menggunakan Siklus Hidup S3. Untuk informasi harga S3 umum, lihat harga Amazon S3 dan Kalkulator Harga AWS.

- Biaya transisi siklus hidup

  Setiap kali objek ditransisikan ke kelas penyimpanan yang berbeda berdasarkan aturan siklus hidup, Amazon S3 menghitung transisi tersebut sebagai satu permintaan transisi. Biaya untuk permintaan transisi ini merupakan tambahan dari biaya kelas penyimpanan ini. Jika Anda berencana untuk mentransisi sejumlah besar objek, pertimbangkan biaya permintaan saat melakukan transisi ke tingkat yang lebih rendah. Untuk informasi selengkapnya, lihat Harga Amazon S3.

Saya telah memperbarui kebijakan bucket saya, tetapi objek S3 saya masih dihapus oleh aturan siklus hidup yang kedaluwarsa.

Pernyataan `Deny` dalam kebijakan bucket tidak mencegah berakhirnya objek yang ditentukan dalam aturan siklus hidup. Tindakan siklus hidup (seperti transisi atau kedaluwarsa) tidak menggunakan operasi `DeleteObject` S3. Sebagai gantinya, tindakan Siklus Hidup S3 dilakukan dengan menggunakan titik akhir S3 internal. (Untuk informasi lebih lanjut, lihat Siklus hidup dan pencatatan.)

Untuk mencegah aturan siklus hidup Anda mengambil tindakan apa pun, Anda harus mengedit, menghapus, atau menonaktifkan aturan tersebut.

Dapatkah saya memulihkan objek S3 yang dikahiri oleh aturan Siklus Hidup S3?

Satu-satunya cara untuk memulihkan objek yang diakhiri oleh Siklus Hidup S3 adalah melalui Penentuan Versi, yang harus dilakukan sebelum objek memenuhi syarat untuk kedaluwarsa. Anda tidak dapat membatalkan operasi kedaluwarsa yang dilakukan oleh aturan siklus hidup. Jika objek dihapus secara permanen oleh aturan Siklus Hidup S3 yang ada, Anda tidak dapat memulihkan objek ini. Untuk mengaktifkan Penentuan Versi pada bucket, lihat the section called “Mempertahankan beberapa versi objek”.

Jika Anda telah menerapkan Penentuan Versi ke bucket dan versi objek yang tidak terkini masih utuh, Anda dapat memulihkan versi sebelumnya dari objek yang kedaluwarsa. Untuk informasi selengkapnya tentang perilaku tindakan aturan Siklus Hidup S3 dan status Penentuan Versi, lihat tabel status aksi Siklus Hidup dan Penentuan Versi bucket di Elemen untuk menjelaskan tindakan siklus hidup.

> **Note**
>
> Jika bucket S3 dilindungi oleh AWS Backup atau Replikasi S3, Anda mungkin juga dapat menggunakan fitur ini untuk memulihkan objek yang kedaluwarsa.

Mengapa tindakan kedaluwarsa dan siklus hidup transisi saya tidak terjadi?

Untuk bucket berkemampuan versi atau ditangguhkan versi, pertimbangan berikut memandu cara Amazon S3 menangani tindakan Kedaluwarsa:

- Kedaluwarsa objek hanya berlaku untuk versi objek saat ini (tidak memiliki dampak pada versi objek lama).

- Amazon S3 tidak mengambil tindakan apa pun jika ada satu atau beberapa versi objek dan penanda hapus adalah versi saat ini.
- Amazon S3 tidak mengambil tindakan apa pun pada versi objek yang tidak terkini yang menerapkan Kunci Objek S3.
- Untuk objek dengan status `PENDING` replikasi, Amazon S3 tidak mengambil tindakan apa pun versi objek saat ini atau tidak terkini.

Transisi kelas penyimpanan siklus hidup memiliki kendala berikut:

- Secara default, objek yang lebih kecil dari 128 KB tidak akan bertransisi ke kelas penyimpanan apa pun.
- Objek harus disimpan setidaknya selama 30 hari sebelum beralih ke S3 Standard-IA atau S3 One Zone-IA.
- Untuk mengaktifkan versi atau membuat versi bucket yang ditangguhkan, objek dengan status `PENDING` replikasi tidak dapat dialihkan.

Bagaimana cara mengecualikan awalan dari aturan siklus hidup saya?

Siklus Hidup S3 tidak mendukung pengecualian awalan dalam aturan Anda. Sebagai gantinya, gunakan tag untuk menandai semua objek yang ingin Anda sertakan dalam aturan. Untuk informasi selengkapnya tentang penggunaan tag dalam aturan siklus hidup Anda, lihat. the section called “Mengarsipkan semua objek dalam satu hari setelah pembuatan”

Bagaimana saya bisa menyertakan beberapa awalan dalam aturan siklus hidup saya?

Siklus Hidup S3 tidak mendukung menyertakan beberapa awalan dalam aturan Anda. Sebagai gantinya, gunakan tag untuk menandai semua objek yang ingin Anda sertakan dalam aturan. Untuk informasi selengkapnya tentang penggunaan tag dalam aturan siklus hidup Anda, lihat. the section called “Mengarsipkan semua objek dalam satu hari setelah pembuatan”

Namun, jika Anda memiliki satu atau lebih awalan yang dimulai dengan karakter yang sama, Anda dapat menyertakan semua awalan tersebut dalam aturan Anda dengan menentukan awalan parsi tanpa garis miring () di filter. / Misalnya, anggaplah Anda memiliki awalan ini:

```
sales1999/
sales2000/
sales2001/
```

Untuk menyertakan ketiga awalan dalam aturan Anda, tentukan `<Prefix>sales</Prefix>` dalam aturan siklus hidup Anda.

Pencatatan log dan pemantauan di Amazon S3

Pemantauan adalah bagian penting dalam menjaga keandalan, ketersediaan, dan kinerja Amazon S3 dan solusi Anda AWS . Kami menyarankan untuk mengumpulkan data pemantauan dari semua bagian AWS solusi Anda sehingga Anda dapat lebih mudah men-debug kegagalan multipoint jika terjadi. Namun sebelum Anda mulai memantau Amazon S3, buat rencana pemantauan yang mencakup jawaban atas pertanyaan berikut:

- Apa tujuan pemantauan Anda?
- Sumber daya apa yang akan Anda pantau?
- Seberapa sering Anda akan memantau sumber daya ini?
- Alat pemantauan apa yang akan Anda gunakan?
- Siapa yang akan melakukan tugas pemantauan?
- Siapa yang harus diberi tahu saat terjadi kesalahan?

Untuk informasi lebih lanjut tentang pencatatan dan pemantauan di Amazon S3, lihat topik berikut.

> **Note**
>
> Untuk informasi selengkapnya tentang menggunakan kelas penyimpanan Amazon S3 Express One Zone dengan bucket direktori, lihat S3 Express One Zone dan Bekerja dengan bucket direktori.

Pemantauan adalah bagian penting dalam menjaga keandalan, ketersediaan, dan kinerja Amazon S3 dan solusi Anda AWS . Anda harus mengumpulkan data pemantauan dari semua bagian AWS solusi Anda sehingga Anda dapat lebih mudah men-debug kegagalan multi-titik jika terjadi. AWS menyediakan beberapa alat untuk memantau sumber daya Amazon S3 Anda dan menanggapi potensi insiden.

CloudWatch Alarm Amazon

Menggunakan CloudWatch alarm Amazon, Anda menonton satu metrik selama periode waktu yang Anda tentukan. Jika metrik melebihi ambang batas tertentu, pemberitahuan akan dikirim ke topik atau AWS Auto Scaling kebijakan Amazon SNS. CloudWatch alarm tidak memanggil tindakan karena mereka berada dalam keadaan tertentu. Sebaliknya, kondisi tersebut harus

diubah dan dipertahankan selama periode tertentu. Untuk informasi selengkapnya, lihat Memantau metrik dengan Amazon CloudWatch.

AWS CloudTrail Log

CloudTrail menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di Amazon S3. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat ke Amazon S3, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan. Untuk informasi selengkapnya, lihat Mencatat panggilan API Amazon S3 menggunakan AWS CloudTrail.

Amazon GuardDuty

Amazon GuardDuty adalah layanan deteksi ancaman yang terus memantau akun, container, beban kerja, dan data di AWS lingkungan Anda untuk mengidentifikasi potensi ancaman atau risiko keamanan terhadap bucket S3 Anda. GuardDuty juga menyediakan konteks yang kaya tentang ancaman yang dideteksinya. GuardDuty memantau log AWS CloudTrail manajemen untuk ancaman dan memunculkan informasi keamanan yang relevan. Misalnya, GuardDuty akan mencakup faktor permintaan API, seperti pengguna yang membuat permintaan, lokasi permintaan dibuat, dan API spesifik yang diminta, yang mungkin tidak biasa di lingkungan Anda. GuardDuty S3 Protection memantau peristiwa data S3 yang dikumpulkan oleh CloudTrail dan mengidentifikasi perilaku yang berpotensi anomali dan berbahaya di semua bucket S3 di lingkungan Anda.

Log Akses Amazon S3

Log akses server memberikan catatan terperinci tentang permintaan yang dibuat ke bucket. Log akses server bermanfaat untuk berbagai macam aplikasi. Misalnya, informasi log akses dapat berguna dalam audit keamanan dan akses. Untuk informasi selengkapnya, lihat Pencatatan permintaan dengan pencatatan akses server.

AWS Trusted Advisor

Trusted Advisor mengacu pada praktik terbaik yang dipelajari dari melayani ratusan ribu AWS pelanggan. Trusted Advisor memeriksa AWS lingkungan Anda dan kemudian membuat rekomendasi ketika ada peluang untuk menghemat uang, meningkatkan ketersediaan dan kinerja sistem, atau membantu menutup kesenjangan keamanan. Semua AWS pelanggan memiliki akses ke lima Trusted Advisor cek. Pelanggan dengan paket dukungan Bisnis atau Perusahaan dapat melihat semua Trusted Advisor pemeriksaan.

Trusted Advisor memiliki pemeriksaan terkait Amazon S3 berikut:

- Konfigurasi pencatatan log bucket Amazon S3.
- Pemeriksaan keamanan untuk bucket Amazon S3 yang memiliki izin akses terbuka.
- Pengecekan toleransi kesalahan untuk bucket Amazon S3 yang tidak memiliki Penentuan Versi yang diaktifkan, atau memiliki Penentuan Versi yang ditangguhkan.

Untuk informasi selengkapnya, lihat AWS Trusted Advisor di Dukungan Panduan Pengguna.

Lensa Penyimpanan Amazon S3

Lensa Penyimpanan Amazon S3 adalah fitur analisis penyimpanan cloud yang dapat Anda gunakan untuk mendapatkan visibilitas seluruh organisasi ke dalam penggunaan dan aktivitas penyimpanan objek. Anda dapat menggunakan metrik Lensa Penyimpanan S3 untuk menghasilkan wawasan ringkasan, seperti mencari tahu berapa banyak penyimpanan yang Anda miliki di seluruh organisasi atau bucket dan prefiks yang paling cepat berkembang. Anda juga dapat menggunakan metrik Lensa Penyimpanan S3 untuk mengidentifikasi peluang optimasi biaya, menerapkan praktik terbaik untuk perlindungan data dan keamanan, serta meningkatkan kinerja beban kerja aplikasi.

Lensa Penyimpanan S3 menggabungkan metrik Anda dan menampilkan informasi di bagian Snapshot akun di halaman Bucket konsol Amazon S3. Lensa Penyimpanan S3 juga menyediakan dasbor interaktif yang dapat Anda gunakan untuk memvisualisasikan wawasan dan tren, menandai outlier, serta menerima rekomendasi untuk mengoptimalkan biaya penyimpanan dan menerapkan praktik terbaik perlindungan data. Dasbor Anda memiliki opsi penelusuran untuk menghasilkan dan memvisualisasikan wawasan di tingkat grup organisasi, akun, kelas penyimpanan, bucket Wilayah AWS, awalan, atau Storage Lens. Untuk informasi selengkapnya, lihat Memahami Lensa Penyimpanan Amazon S3.

Inventaris Amazon S3

Inventaris Amazon S3 menghasilkan daftar objek dan metadata yang dapat Anda gunakan untuk menanyakan dan mengelola objek Anda. Anda dapat menggunakan laporan inventaris ini untuk menghasilkan data terperinci seperti ukuran objek, tanggal modifikasi terakhir, status enkripsi, dan bidang lainnya. Laporan tersebut tersedia setiap hari atau mingguan untuk secara otomatis memberikan daftar terbaru.

Misalnya, Anda dapat menggunakan Inventaris Amazon S3 untuk mengaudit dan melaporkan status replikasi dan enkripsi objek Anda untuk kebutuhan bisnis, kepatuhan, dan peraturan. Anda juga dapat menggunakan Inventaris Amazon S3 untuk menyederhanakan dan mempercepat alur kerja bisnis dan pekerjaan data besar, yang menyediakan alternatif terjadwal untuk operasi API

sinkron Amazon S3. `List` Inventaris Amazon S3 tidak menggunakan operasi `List` API untuk mengaudit objek Anda dan tidak memengaruhi tingkat permintaan bucket Anda. Untuk informasi selengkapnya, lihat Membuat katalog dan menganalisis data Anda dengan S3 Inventory.

Notifikasi Peristiwa Amazon S3

Dengan fitur Pemberitahuan Acara Amazon S3, Anda menerima notifikasi saat peristiwa tertentu terjadi di bucket S3 Anda. Untuk mengaktifkan notifikasi, tambahkan konfigurasi notifikasi yang mengidentifikasi peristiwa yang ingin dipublikasikan Amazon S3. Untuk informasi selengkapnya, lihat Notifikasi Peristiwa Amazon S3.

Amazon S3 dan AWS X-Ray

AWS X-Ray terintegrasi dengan Amazon S3 untuk melacak permintaan upstream untuk memperbarui bucket S3 aplikasi Anda. Jika layanan melacak permintaan dengan menggunakan X-Ray SDK, Amazon S3 dapat mengirim header penelusuran ke pelanggan acara hilir seperti Λ, Amazon SQS, dan Amazon SNS. X-Ray mengaktifkan pesan penelusuran untuk notifikasi peristiwa Amazon S3. Anda dapat menggunakan peta jejak X-Ray untuk melihat koneksi antara Amazon S3 dan layanan lain yang digunakan aplikasi Anda. Untuk informasi selengkapnya, lihat Amazon S3 dan X-Ray.

Praktik keamanan terbaik berikut juga membahas pencatatan dan pemantauan:

- Identify and audit all your Amazon S3 buckets
- Implement monitoring using Amazon Web Services monitoring tools
- Aktifkan AWS Config
- Enable Amazon S3 server access logging
- Use CloudTrail
- Monitor Amazon Web Services security advisories

Topik

- Alat-alat pemantauan
- Opsi pencatatan untuk Amazon S3
- Mencatat panggilan API Amazon S3 menggunakan AWS CloudTrail
- Pencatatan permintaan dengan pencatatan akses server
- Memantau metrik dengan Amazon CloudWatch

- Notifikasi Peristiwa Amazon S3
- Menilai aktivitas penyimpanan dan penggunaan dengan Lensa Penyimpanan Amazon S3
- Membuat katalog dan menganalisis data Anda dengan S3 Inventory

Alat-alat pemantauan

AWS menyediakan berbagai alat yang dapat Anda gunakan untuk memantau Amazon S3. Anda dapat mengonfigurasi beberapa alat ini untuk melakukan pemantauan untuk Anda, sementara beberapa alat memerlukan intervensi manual. Kami menyarankan agar Anda mengautomasi tugas pemantauan sebanyak mungkin.

Alat pemantauan otomatis

Anda dapat menggunakan alat pemantauan otomatis berikut untuk melihat Amazon S3 dan melaporkan saat terjadi kesalahan:

- CloudWatch Alarm Amazon — Tonton satu metrik selama periode waktu yang Anda tentukan, dan lakukan satu atau beberapa tindakan berdasarkan nilai metrik relatif terhadap ambang batas tertentu selama beberapa periode waktu. Tindakannya adalah pemberitahuan yang dikirim ke topik Amazon Simple Notification Service (Amazon SNS) atau kebijakan Amazon Auto EC2 Scaling. CloudWatch alarm tidak memanggil tindakan hanya karena mereka berada dalam keadaan tertentu. Status harus diubah dan dipelihara selama jangka waktu tertentu. Untuk informasi selengkapnya, lihat Memantau metrik dengan Amazon CloudWatch.
- AWS CloudTrail Pemantauan Log - Bagikan file log antar akun, pantau file CloudTrail log secara real time dengan mengirimkannya ke CloudWatch Log, menulis aplikasi pemrosesan log di Java, dan validasi bahwa file log Anda tidak berubah setelah pengiriman oleh CloudTrail. Untuk informasi selengkapnya, lihat Mencatat panggilan API Amazon S3 menggunakan AWS CloudTrail.

Alat pemantauan manual

Bagian penting lainnya dari pemantauan Amazon S3 melibatkan pemantauan secara manual item-item yang tidak CloudWatch tercakup oleh alarm. Amazon S3,, CloudWatch Trusted Advisor, dan AWS Management Console dasbor lainnya memberikan at-a-glance tampilan keadaan lingkungan Anda. AWS Anda mungkin ingin mengaktifkan pencatatan akses server, yang melacak permintaan akses ke bucket Anda. Setiap catatan log akses memberikan perincian tentang permintaan akses tunggal, seperti pemohon, nama bucket, waktu permintaan, tindakan permintaan, status respons,

dan kode kesalahan, jika ada. Untuk informasi selengkapnya, lihat Pencatatan permintaan dengan pencatatan akses server.

- Dasbor Amazon S3 menunjukkan hal berikut:
  - Bucket Anda dan objek serta properti yang dimuat di dalamnya
- CloudWatch Halaman beranda menunjukkan yang berikut:
  - Alarm dan status saat ini
  - Grafik alarm dan sumber daya
  - Status kesehatan layanan

  Selain itu, Anda dapat menggunakan CloudWatch untuk melakukan hal berikut:
  - Buat dasbor yang disesuaikan untuk memantau layanan yang Anda pedulikan.
  - Data metrik grafik untuk memecahkan masalah dan menemukan tren.
  - Cari dan telusuri semua metrik AWS sumber daya Anda.
  - Membuat dan mengedit alarm untuk menerima notifikasi terkait masalah.
- AWS Trusted Advisor dapat membantu Anda memantau AWS sumber daya Anda untuk meningkatkan kinerja, keandalan, keamanan, dan efektivitas biaya. Empat pemeriksaan Trusted Advisor tersedia bagi semua pengguna; lebih dari 50 pemeriksaan tersedia bagi pengguna dengan perencanaan dukungan Bisnis atau Korporasi. Untuk informasi selengkapnya, lihat AWS Trusted Advisor.

  Trusted Advisor memiliki pemeriksaan ini yang berhubungan dengan Amazon S3:
  - Pemeriksaan konfigurasi pencatatan log bucket Amazon S3.
  - Pemeriksaan keamanan untuk bucket Amazon S3 yang memiliki izin akses terbuka.
  - Pengecekan toleransi kesalahan untuk bucket Amazon S3 yang tidak memiliki Penentuan Versi yang diaktifkan, atau memiliki Penentuan Versi yang ditangguhkan.

Opsi pencatatan untuk Amazon S3

Anda dapat merekam tindakan yang diambil oleh pengguna, peran, atau sumber daya Layanan AWS Amazon S3 dan memelihara catatan log untuk tujuan audit dan kepatuhan. Untuk melakukan ini, Anda dapat menggunakan pencatatan akses server, pencatatan AWS CloudTrail , atau kombinasi keduanya. Sebaiknya gunakan CloudTrail untuk mencatat tindakan tingkat ember dan tingkat objek untuk sumber daya Amazon S3 Anda. Untuk informasi selengkapnya tentang setiap opsi, lihat bagian berikut:

- Pencatatan permintaan dengan pencatatan akses server
- Mencatat panggilan API Amazon S3 menggunakan AWS CloudTrail

Tabel berikut mencantumkan properti kunci CloudTrail log dan log akses server Amazon S3. Untuk memastikan bahwa CloudTrail memenuhi persyaratan keamanan Anda, tinjau tabel dan catatan.

Properti log	AWS CloudTrail	Log server Amazon S3
Dapat diteruskan ke sistem lain (Amazon CloudWatch Log, Amazon Events) CloudWatch	Ya	Tidak
Kirim log ke lebih dari satu tujuan (misalnya, kirim log yang sama ke dua bucket yang berbeda)	Ya	Tidak
Mengaktifkan log untuk subset objek (prefiks)	Ya	Tidak
Pengiriman log akun silang (bucket target dan sumber yang dimiliki oleh akun yang berbeda)	Ya	Tidak
Validasi integritas file log dengan menggunakan tanda tangan digital atau hashing	Ya	Tidak
Default atau pilihan enkripsi untuk file log	Ya	Tidak
Operasi objek (dengan menggunakan Amazon S3) APIs	Ya	Ya

Properti log	AWS CloudTrail	Log server Amazon S3
Operasi bucket (dengan menggunakan Amazon S3) APIs	Ya	Ya
UI yang dapat dicari untuk log	Ya	Tidak
Bidang untuk parameter Kunci Objek, Amazon S3 Pilih properti untuk catatan log	Ya	Tidak
Bidang untuk `Object Size`, `Total Time`, `Turn-Around Time` , dan `HTTP Referer` untuk catatan log	Tidak	Ya
Transisi Lifecycle, kedaluwarsa, pemulihan	Tidak	Ya
Pencatatan log kunci dalam operasi penghapusan batch	Ya	Ya
Kegagalan autentikasi[1]	Tidak	Ya
Akun tempat log dikirimkan	Pemilik bucket[2], dan pemohon	Hanya pemilik bucket
Performance and Cost	AWS CloudTrail	Amazon S3 Server Logs
Harga	Peristiwa manajemen (pengiriman pertama) gratis; peristiwa data dikenakan biaya, selain penyimpanan log	Tidak ada biaya lain selain penyimpanan log
Kecepatan pengiriman log	Peristiwa data setiap 5 menit; peristiwa manajemen setiap 15 menit	Dalam beberapa jam

Properti log	AWS CloudTrail	Log server Amazon S3
Format log	JSON	Berkas log dengan catatan yang dipisahkan oleh ruang, yang tidak terbatas pada baris baru

Catatan

1. CloudTrail tidak mengirimkan log untuk permintaan yang gagal otentikasi (di mana kredensi yang diberikan tidak valid) atau yang gagal karena pengalihan (kode kesalahan). `301 Moved Permanently` Namun demikian, hal tersebut mencakup log untuk permintaan yang gagal diterima otorisasi (`AccessDenied`) dan permintaan yang dibuat oleh pengguna anonim.
2. Pemilik bucket S3 menerima CloudTrail log ketika akun tidak memiliki akses penuh ke objek dalam permintaan. Untuk informasi selengkapnya, lihat Tindakan tingkat objek Amazon S3 dalam skenario lintas akun.
3. S3 tidak mendukung pengiriman CloudTrail log atau log akses server ke pemohon atau pemilik bucket untuk permintaan titik akhir VPC saat kebijakan titik akhir VPC menyangkalnya atau untuk permintaan yang gagal sebelum kebijakan VPC dievaluasi.

Mencatat panggilan API Amazon S3 menggunakan AWS CloudTrail

Amazon S3 terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau. Layanan AWS CloudTrail menangkap semua panggilan API untuk Amazon S3 sebagai peristiwa. Panggilan yang diambil termasuk panggilan dari konsol Amazon S3 dan panggilan kode ke operasi API Amazon S3. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat ke Amazon S3, alamat IP dari mana permintaan dibuat, kapan dibuat, dan detail tambahan.

Setiap entri peristiwa atau log berisi informasi tentang entitas yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan berikut hal ini:

- Baik permintaan tersebut dibuat dengan kredensial pengguna root atau pengguna.
- Apakah permintaan dibuat atas nama pengguna IAM Identity Center.

- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna gabungan.
- Apakah permintaan tersebut dibuat oleh Layanan AWS lain.

CloudTrail aktif di Anda Akun AWS ketika Anda membuat akun dan Anda secara otomatis memiliki akses ke riwayat CloudTrail Acara. Riwayat CloudTrail Acara menyediakan catatan yang dapat dilihat, dapat dicari, dapat diunduh, dan tidak dapat diubah dari 90 hari terakhir dari peristiwa manajemen yang direkam dalam file. Wilayah AWS Untuk informasi selengkapnya, lihat Bekerja dengan riwayat CloudTrail Acara di Panduan AWS CloudTrail Pengguna. Tidak ada CloudTrail biaya untuk melihat riwayat Acara.

Untuk catatan acara yang sedang berlangsung dalam 90 hari Akun AWS terakhir Anda, buat jejak atau penyimpanan data acara CloudTrail Danau.

CloudTrail jalan setapak

Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Semua jalur yang dibuat menggunakan AWS Management Console Multi-region. Anda dapat membuat jalur Single-region atau Multi-region dengan menggunakan. AWS CLI Membuat jejak Multi-wilayah disarankan karena Anda menangkap aktivitas Wilayah AWS di semua akun Anda. Jika Anda membuat jejak wilayah Tunggal, Anda hanya dapat melihat peristiwa yang dicatat di jejak. Wilayah AWS Untuk informasi selengkapnya tentang jejak, lihat Membuat jejak untuk Anda Akun AWS dan Membuat jejak untuk organisasi di Panduan AWS CloudTrail Pengguna.

Anda dapat mengirimkan satu salinan acara manajemen yang sedang berlangsung ke bucket Amazon S3 Anda tanpa biaya CloudTrail dengan membuat jejak, namun, ada biaya penyimpanan Amazon S3. Untuk informasi selengkapnya tentang CloudTrail harga, lihat AWS CloudTrail Harga. Untuk informasi tentang harga Amazon S3, lihat Harga Amazon S3.

CloudTrail Menyimpan data acara danau

CloudTrail Lake memungkinkan Anda menjalankan kueri berbasis SQL pada acara Anda. CloudTrail Lake mengubah peristiwa yang ada dalam format JSON berbasis baris ke format Apache ORC. ORC adalah format penyimpanan kolumnar yang dioptimalkan untuk pengambilan data dengan cepat. Peristiwa digabungkan ke dalam penyimpanan data peristiwa, yang merupakan kumpulan peristiwa yang tidak dapat diubah berdasarkan kriteria yang Anda pilih dengan menerapkan pemilih acara tingkat lanjut. Penyeleksi yang Anda terapkan ke penyimpanan data acara mengontrol peristiwa mana yang bertahan dan tersedia untuk Anda

kueri. Untuk informasi lebih lanjut tentang CloudTrail Danau, lihat Bekerja dengan AWS CloudTrail Danau di Panduan AWS CloudTrail Pengguna.

CloudTrail Penyimpanan data acara danau dan kueri menimbulkan biaya. Saat Anda membuat penyimpanan data acara, Anda memilih opsi harga yang ingin Anda gunakan untuk penyimpanan data acara. Opsi penetapan harga menentukan biaya untuk menelan dan menyimpan peristiwa, dan periode retensi default dan maksimum untuk penyimpanan data acara. Untuk informasi selengkapnya tentang CloudTrail harga, lihat AWS CloudTrail Harga.

Anda dapat menyimpan file log dalam bucket selama yang Anda inginkan, tetapi Anda juga dapat menentukan aturan siklus hidup Amazon S3 untuk mengarsipkan atau menghapus file log secara otomatis. Secara default, file log Anda dienkripsi dengan menggunakan enkripsi di sisi server (SSE) Amazon S3.

Menggunakan CloudTrail log dengan log akses server Amazon S3 dan Log CloudWatch

AWS CloudTrail log menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di Amazon S3, sementara log akses server Amazon S3 menyediakan catatan terperinci untuk permintaan yang dibuat ke bucket S3. Untuk informasi lebih lanjut tentang cara kerja pencatatan yang berbeda, dan properti, performa, dan biayanya, lihat the section called “Opsi pencatatan”.

Anda dapat menggunakan AWS CloudTrail log bersama dengan log akses server untuk Amazon S3. CloudTrail log memberi Anda pelacakan API terperinci untuk operasi tingkat ember dan tingkat objek Amazon S3. Log akses server untuk Amazon S3 memberi Anda visibilitas ke operasi tingkat objek pada data Anda di Amazon S3. Untuk informasi lebih lanjut tentang log akses server, lihat Pencatatan permintaan dengan pencatatan akses server.

Anda juga dapat menggunakan CloudTrail log bersama dengan Amazon CloudWatch untuk Amazon S3. CloudTrail integrasi dengan CloudWatch Log memberikan aktivitas API tingkat ember S3 yang ditangkap oleh aliran CloudWatch log di CloudTrail grup log yang Anda tentukan CloudWatch . Anda dapat membuat CloudWatch alarm untuk memantau aktivitas API tertentu dan menerima pemberitahuan email saat aktivitas API tertentu terjadi. Untuk informasi selengkapnya tentang CloudWatch alarm untuk memantau aktivitas API tertentu, lihat Panduan AWS CloudTrail Pengguna. Untuk informasi selengkapnya tentang penggunaan CloudWatch dengan Amazon S3, lihat. Memantau metrik dengan Amazon CloudWatch

> **Note**
>
> S3 tidak mendukung pengiriman CloudTrail log ke pemohon atau pemilik bucket untuk permintaan titik akhir VPC saat kebijakan titik akhir VPC menolaknya.

CloudTrail melacak dengan panggilan API SOAP Amazon S3

CloudTrail melacak panggilan API SOAP Amazon S3. Dukungan SOAP Amazon S3 melalui HTTP dihentikan, tetapi masih tersedia melalui HTTPS. Untuk informasi selengkapnya tentang dukungan Amazon S3 SOAP, lihat Lampiran: SOAP API di Referensi API Amazon S3.

> **Important**
>
> Fitur Amazon S3 baru tidak didukung untuk SOAP. Kami menyarankan Anda menggunakan REST API atau AWS SDKs.

Tabel berikut menunjukkan tindakan SOAP Amazon S3 yang dilacak oleh logging. CloudTrail

Nama SOAP API	Nama peristiwa API yang digunakan di CloudTrail log
ListAllMyBuckets	`ListBuckets`
CreateBucket	`CreateBucket`
DeleteBucket	`DeleteBucket`
GetBucketAccessControlPolicy	`GetBucketAcl`
SetBucketAccessControlPolicy	`PutBucketAcl`
GetBucketLoggingStatus	`GetBucketLogging`
SetBucketLoggingStatus	`PutBucketLogging`

Untuk informasi selengkapnya tentang CloudTrail Amazon S3, lihat topik berikut:

Topik

- Acara Amazon S3 CloudTrail
- CloudTrail entri file log untuk Amazon S3 dan S3 di Outposts
- Mengaktifkan pencatatan CloudTrail peristiwa untuk bucket dan objek S3
- Mengidentifikasi permintaan Amazon S3 menggunakan CloudTrail

Acara Amazon S3 CloudTrail

> **Important**
>
> Amazon S3 sudah menerapkan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3) sebagai tingkat dasar enkripsi bagi setiap bucket di Amazon S3. Mulai 5 Januari 2023, semua unggahan objek baru ke Amazon S3 secara otomatis akan dienkripsi tanpa biaya tambahan dan tidak akan berdampak pada kinerja. Status enkripsi otomatis untuk konfigurasi enkripsi default bucket S3 dan untuk unggahan objek baru tersedia di AWS CloudTrail log, S3 Inventory, S3 Storage Lens, konsol Amazon S3, dan sebagai header respons API Amazon S3 tambahan di dan. AWS Command Line Interface AWS SDKs Untuk informasi selengkapnya, lihat FAQ enkripsi default.

Bagian ini memberikan informasi tentang peristiwa yang dicatat oleh S3. CloudTrail

Peristiwa data Amazon S3 di CloudTrail

Peristiwa data memberikan informasi tentang operasi sumber daya yang dilakukan pada atau di sumber daya (misalnya, membaca atau menulis ke objek Amazon S3). Ini juga dikenal sebagai operasi bidang data. Peristiwa data seringkali merupakan aktivitas volume tinggi. Secara default, CloudTrail tidak mencatat peristiwa data. Riwayat CloudTrail peristiwa tidak merekam peristiwa data.

Biaya tambahan berlaku untuk peristiwa data. Untuk informasi lebih lanjut tentang harga CloudTrail, lihat Harga AWS CloudTrail.

Anda dapat mencatat peristiwa data untuk jenis sumber daya Amazon S3 menggunakan CloudTrail konsol AWS CLI, atau operasi CloudTrail API. Untuk informasi selengkapnya tentang cara mencatat peristiwa data, lihat Mencatat peristiwa data dengan AWS Management Console dan Logging peristiwa data dengan AWS Command Line Interface di Panduan AWS CloudTrail Pengguna.

Tabel berikut mencantumkan jenis sumber daya Amazon S3 yang dapat Anda log peristiwa data. Kolom tipe peristiwa data (konsol) menunjukkan nilai yang akan dipilih dari daftar tipe peristiwa Data di CloudTrail konsol. Kolom nilai resources.type menunjukkan **`resources.type`** nilai, yang akan Anda tentukan saat mengonfigurasi penyeleksi acara lanjutan menggunakan or. AWS CLI CloudTrail APIs CloudTrailKolom Data yang APIs dicatat ke menampilkan panggilan API yang dicatat CloudTrail untuk jenis sumber daya.

Jenis peristiwa data (konsol)	nilai resources.type	Data APIs masuk ke CloudTrail
S3	`AWS::S3::Object`	• AbortMultipartUpload • CompleteMultipartUpload • CopyObject • CreateMultipartUpload • DeleteObject • DeleteObjectTagging • DeleteObjects • GetObject • GetObjectAcl • GetObjectAttributes • GetObjectLegalHold • GetObjectRetention • GetObjectTagging • GetObjectTorrent • HeadObject • HeadBucket • ListObjectVersions • ListObjects • ListParts • PutObject • PutObjectAcl • PutObjectLegalHold

Jenis peristiwa data (konsol)	nilai resources.type	Data APIs masuk ke CloudTrail
		• PutObjectRetention • PutObjectTagging • RestoreObject • SelectObjectContent • UploadPart • UploadPartCopy
S3 Express Satu Zona	`AWS::S3Express::Object`	• AbortMultipartUpload • CompleteMultipartUpload • CreateSession • CopyObject • CreateMultipartUpload • DeleteObject • DeleteObjects • GetObject • GetObjectAttributes • HeadBucket • HeadObject • ListObjectsV2 • ListParts • PutObject • UploadPart • UploadPartCopy

Jenis peristiwa data (konsol)	nilai resources.type	Data APIs masuk ke CloudTrail
Titik Akses S3	`AWS::S3::Access Point`	• AbortMultipartUpload • CompleteMultipartUpload • CopyObject (hanya salinan wilayah yang sama) • CreateMultipartUpload • DeleteObject • DeleteObjectTagging • GetBucketAcl • GetBucketCors • GetBucketLocation • GetBucketNotificationConfiguration • GetBucketPolicy • GetObject • GetObjectAcl • GetObjectAttributes • GetObjectLegalHold • GetObjectRetention • GetObjectTagging • HeadBucket • HeadObject • ListObjects • ListObjectsV2 • ListObjectVersions • ListParts • Presign • PutObject • PutObjectLegalHold

Jenis peristiwa data (konsol)	nilai resources.type	Data APIs masuk ke CloudTrail
		• PutObjectRetention • PutObjectAcl • PutObjectTagging • RestoreObject • UploadPart • UploadPartCopy (hanya salinan wilayah yang sama)

Jenis peristiwa data (konsol)	nilai resources.type	Data APIs masuk ke CloudTrail
S3 Object Lambda	`AWS::S3ObjectLambda::AccessPoint`	• AbortMultipartUpload • CompleteMultipartUpload • CopyObject (hanya salinan wilayah yang sama) • CreateMultipartUpload • DeleteObject • DeleteObjectTagging • GetObject • GetObjectAcl • GetObjectLegalHold • GetObjectRetention • GetObjectTagging • HeadObject • ListObjects • ListObjectVersions • ListParts • PutObject • PutObjectLegalHold • PutObjectRetention • PutObjectAcl • PutObjectTagging • RestoreObject • UploadPart • WriteGetObjectResponse

Jenis peristiwa data (konsol)	nilai resources.type	Data APIs masuk ke CloudTrail
Outposts S3	`AWS::S3Outposts::Object`	• AbortMultipartUpload • CompleteMultipartUpload • CopyObject (hanya salinan wilayah yang sama) • CreateMultipartUpload • DeleteObject • DeleteObjectTagging • GetObject • GetObjectTagging • HeadObject • ListObjects • ListObjectsV2 • ListParts • PutObject • PutObjectTagging • UploadPart • UploadPartCopy

Anda dapat mengonfigurasi pemilih acara lanjutan untuk memfilter pada `eventName`,`readOnly`, dan `resources.ARN` bidang untuk mencatat hanya peristiwa yang penting bagi Anda. Untuk informasi selengkapnya tentang bidang ini, lihat AdvancedFieldSelectordi Referensi AWS CloudTrail API.

Acara manajemen Amazon S3 di CloudTrail

Amazon S3 mencatat semua operasi pesawat kontrol sebagai peristiwa manajemen. Untuk informasi selengkapnya tentang operasi API S3, lihat Referensi API Amazon S3.

Cara CloudTrail menangkap permintaan yang dibuat ke Amazon S3

Secara default, CloudTrail mencatat panggilan API tingkat ember S3 yang dibuat dalam 90 hari terakhir, tetapi tidak mencatat permintaan yang dibuat ke objek. Panggilan tingkat bucket mencakup

peristiwa seperti `CreateBucket`, `DeleteBucket`, `PutBucketLifecycle`, `PutBucketPolicy`, dan seterusnya. Anda dapat melihat peristiwa tingkat ember di konsol. CloudTrail Namun, Anda tidak dapat melihat peristiwa data (panggilan tingkat objek Amazon S3) di sana—Anda harus mengurai atau mengajukan kueri log CloudTrail untuk itu.

Jika Anda mencatat aktivitas data dengan AWS CloudTrail, catatan peristiwa untuk peristiwa `DeleteObjects` data Amazon S3 mencakup peristiwa dan `DeleteObjects` `DeleteObject` peristiwa untuk setiap objek yang dihapus sebagai bagian dari operasi tersebut. Anda dapat mengecualikan visibilitas tambahan tentang objek yang dihapus dari catatan peristiwa. Untuk informasi selengkapnya, lihat AWS CLI contoh untuk memfilter peristiwa data di Panduan AWS CloudTrail Pengguna.

Tindakan tingkat akun Amazon S3 dilacak dengan pencatatan CloudTrail

CloudTrail mencatat tindakan tingkat akun. Catatan Amazon S3 ditulis bersama dengan Layanan AWS catatan lain dalam file log. CloudTrail menentukan kapan harus membuat dan menulis ke file baru berdasarkan periode waktu dan ukuran file.

Tabel di bagian ini mencantumkan tindakan tingkat akun Amazon S3 yang didukung untuk pencatatan. CloudTrail

Tindakan API tingkat akun Amazon S3 yang dilacak dengan CloudTrail logging muncul sebagai nama peristiwa berikut. Nama CloudTrail acara berbeda dari nama tindakan API. Misalnya, DeletePublicAccessBlock adalah DeleteAccountPublicAccessBlock.

- DeleteAccountPublicAccessBlock
- GetAccountPublicAccessBlock
- PutAccountPublicAccessBlock

Tindakan tingkat ember Amazon S3 yang dilacak dengan logging CloudTrail

Secara default, CloudTrail mencatat tindakan tingkat ember untuk bucket tujuan umum. Catatan Amazon S3 ditulis bersama dengan catatan AWS layanan lain dalam file log. CloudTrail menentukan kapan harus membuat dan menulis ke file baru berdasarkan periode waktu dan ukuran file.

Bagian ini mencantumkan tindakan tingkat ember Amazon S3 yang didukung untuk pencatatan oleh. CloudTrail

Tindakan API tingkat ember Amazon S3 yang dilacak dengan CloudTrail logging muncul sebagai nama peristiwa berikut. Dalam beberapa kasus, nama CloudTrail acara berbeda dari nama tindakan API. Misalnya, `PutBucketLifecycleConfiguration` adalah`PutBucketLifecycle`.

- CreateBucket
- CreateBucketMetadataTableConfiguration
- DeleteBucket
- DeleteBucketAnalyticsConfiguration
- DeleteBucketCors
- DeleteBucketEncryption
- DeleteBucketIntelligentTieringConfiguration
- DeleteBucketInventoryConfiguration
- DeleteBucketLifecycle
- DeleteBucketMetadataTableConfiguration
- DeleteBucketMetricsConfiguration
- DeleteBucketOwnershipControls
- DeleteBucketPolicy
- DeleteBucketPublicAccessBlock
- DeleteBucketReplication
- DeleteBucketTagging
- GetAccelerateConfiguration
- GetBucketAcl
- GetBucketAnalyticsConfiguration
- GetBucketCors
- GetBucketEncryption
- GetBucketIntelligentTieringConfiguration
- GetBucketInventoryConfiguration
- GetBucketLifecycle
- GetBucketLocation
- GetBucketLogging
- GetBucketMetadataTableConfiguration

- GetBucketMetricsConfiguration
- GetBucketNotification
- GetBucketObjectLockConfiguration
- GetBucketOwnershipControls
- GetBucketPolicy
- GetBucketPolicyStatus
- GetBucketPublicAccessBlock
- GetBucketReplication
- GetBucketRequestPayment
- GetBucketTagging
- GetBucketVersioning
- GetBucketWebsite
- HeadBucket
- ListBuckets
- PutAccelerateConfiguration
- PutBucketAcl
- PutBucketAnalyticsConfiguration
- PutBucketCors
- PutBucketEncryption
- PutBucketIntelligentTieringConfiguration
- PutBucketInventoryConfiguration
- PutBucketLifecycle
- PutBucketLogging
- PutBucketMetricsConfiguration
- PutBucketNotification
- PutBucketObjectLockConfiguration
- PutBucketOwnershipControls
- PutBucketPolicy
- PutBucketPublicAccessBlock
- PutBucketReplication

- PutBucketRequestPayment
- PutBucketTagging
- PutBucketVersioning
- PutBucketWebsite

Selain operasi API ini, Anda juga dapat menggunakan tindakan tingkat objek OPTIONS objek. Tindakan ini diperlakukan seperti tindakan tingkat ember dalam CloudTrail logging karena tindakan memeriksa konfigurasi CORS bucket.

Tindakan tingkat ember Amazon S3 Express One Zone (titik akhir API Regional) yang dilacak dengan pencatatan CloudTrail

Secara default, CloudTrail mencatat tindakan tingkat ember untuk bucket direktori sebagai peristiwa manajemen. Acara `eventsource` untuk CloudTrail manajemen untuk S3 Express One Zone adalah`s3express.amazonaws.com`.

Operasi API titik akhir Regional berikut ini dicatat. CloudTrail

- CreateBucket
- DeleteBucket
- DeleteBucketPolicy
- GetBucketPolicy
- PutBucketPolicy
- ListDirectoryBuckets
- PutBucketEncryption
- GetBucketEncryption
- DeleteBucketEncryption

Untuk informasi selengkapnya, lihat Logging with AWS CloudTrail untuk S3 Express One Zone

Tindakan tingkat objek Amazon S3 dalam skenario lintas akun

Berikut ini adalah kasus penggunaan khusus yang melibatkan panggilan API tingkat objek dalam skenario lintas akun dan bagaimana CloudTrail log dilaporkan. CloudTrail mengirimkan log ke pemohon (akun yang melakukan panggilan API), kecuali dalam beberapa kasus akses ditolak di

mana entri log disunting atau dihilangkan. Saat mengatur akses lintas akun, pertimbangkan contoh di bagian ini.

> **Note**
>
> Contoh mengasumsikan bahwa CloudTrail log dikonfigurasi dengan tepat.

Contoh 1: CloudTrail mengirimkan log ke pemilik bucket

CloudTrail mengirimkan log ke pemilik bucket meskipun pemilik bucket tidak memiliki izin untuk operasi API objek yang sama. Pertimbangkan skenario lintas akun berikut ini:

- Akun A adalah pemilik bucket.
- Akun-B (peminta) mencoba mengakses objek dalam bucket tersebut.
- Akun-C memiliki objek. Akun C mungkin atau mungkin bukan akun yang sama dengan Akun A.

> **Note**
>
> CloudTrail selalu mengirimkan log API tingkat objek ke pemohon (Akun B). Selain itu, CloudTrail juga mengirimkan log yang sama ke pemilik bucket (Akun A) bahkan ketika pemilik bucket tidak memiliki objek (Akun C) atau memiliki izin untuk operasi API yang sama pada objek tersebut.

Contoh 2: CloudTrail tidak memperbanyak alamat email yang digunakan dalam pengaturan objek ACLs

Pertimbangkan skenario lintas akun berikut ini:

- Akun A adalah pemilik bucket.
- Akun B (pemohon) mengirimkan permintaan untuk menetapkan pemberian izin ACL objek dengan menggunakan alamat surel. Untuk informasi lebih lanjut tentang ACLs, lihatGambaran umum daftar kontrol akses (ACL).

Pemohon mendapatkan log beserta informasi surel. Namun, pemilik bucket — jika mereka memenuhi syarat untuk menerima log, seperti pada contoh 1—mendapatkan log yang melaporkan peristiwa

tersebut CloudTrail . Namun, pemilik bucket tidak mendapatkan konfigurasi ACL, kustomnya alamat email penerima izin dan izinnya. Satu-satunya informasi yang diberitahukan oleh log kepada pemilik bucket adalah bahwa panggilan API ACL dilakukan oleh Akun-B.

CloudTrail entri file log untuk Amazon S3 dan S3 di Outposts

> **Important**
>
> Amazon S3 sudah menerapkan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3) sebagai tingkat dasar enkripsi bagi setiap bucket di Amazon S3. Mulai 5 Januari 2023, semua unggahan objek baru ke Amazon S3 secara otomatis akan dienkripsi tanpa biaya tambahan dan tidak akan berdampak pada kinerja. Status enkripsi otomatis untuk konfigurasi enkripsi default bucket S3 dan untuk unggahan objek baru tersedia di AWS CloudTrail log, S3 Inventory, S3 Storage Lens, konsol Amazon S3, dan sebagai header respons API Amazon S3 tambahan di dan. AWS Command Line Interface AWS SDKs Untuk informasi selengkapnya, lihat FAQ enkripsi default.

Peristiwa mewakili permintaan tunggal dari sumber manapun dan mencakup informasi tentang operasi API yang diminta, tanggal dan waktu operasi, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, sehingga peristiwa tidak muncul dalam urutan tertentu.

> **Note**
>
> Untuk melihat contoh file CloudTrail log untuk Amazon S3 Express One Zone, lihat contoh file CloudTrail log untuk S3 Express One Zone.

Untuk informasi selengkapnya, lihat contoh berikut ini.

Topik

- Contoh: entri file CloudTrail log untuk Amazon S3

Contoh: entri file CloudTrail log untuk Amazon S3

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan GETLayanan, PutBucketAcl, dan GetBucketVersioningtindakan.

```
{
    "Records": [
    {
        "eventVersion": "1.03",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "111122223333",
            "arn": "arn:aws:iam::111122223333:user/myUserName",
            "accountId": "111122223333",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "userName": "myUserName"
        },
        "eventTime": "2019-02-01T03:18:19Z",
        "eventSource": "s3.amazonaws.com",
        "eventName": "ListBuckets",
        "awsRegion": "us-west-2",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "[]",
        "requestParameters": {
            "host": [
                "s3.us-west-2.amazonaws.com"
            ]
        },
        "responseElements": null,
        "additionalEventData": {
            "SignatureVersion": "SigV2",
            "AuthenticationMethod": "QueryString",
            "aclRequired": "Yes"
    },
        "requestID": "47B8E8D397DCE7A6",
        "eventID": "cdc4b7ed-e171-4cef-975a-ad829d4123e8",
        "eventType": "AwsApiCall",
        "recipientAccountId": "444455556666",
        "tlsDetails": {
            "tlsVersion": "TLSv1.2",
            "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
            "clientProvidedHostHeader": "s3.amazonaws.com"
    }
    },
    {
       "eventVersion": "1.03",
       "userIdentity": {
            "type": "IAMUser",
```

```
            "principalId": "111122223333",
            "arn": "arn:aws:iam::111122223333:user/myUserName",
            "accountId": "111122223333",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "userName": "myUserName"
        },
    "eventTime": "2019-02-01T03:22:33Z",
    "eventSource": "s3.amazonaws.com",
    "eventName": "PutBucketAcl",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "",
    "userAgent": "[]",
    "requestParameters": {
        "bucketName": "",
        "AccessControlPolicy": {
            "AccessControlList": {
                "Grant": {
                    "Grantee": {
                        "xsi:type": "CanonicalUser",
                        "xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",
                        "ID":
"d25639fbe9c19cd30a4c0f43fbf00e2d3f96400a9aa8dabfbbebe1906Example"
                    },
                    "Permission": "FULL_CONTROL"
                }
            },
            "xmlns": "http://s3.amazonaws.com/doc/2006-03-01/",
            "Owner": {
                "ID":
"d25639fbe9c19cd30a4c0f43fbf00e2d3f96400a9aa8dabfbbebe1906Example"
            }
        },
        "host": [
            "s3.us-west-2.amazonaws.com"
        ],
        "acl": [
            ""
        ]
    },
    "responseElements": null,
    "additionalEventData": {
        "SignatureVersion": "SigV4",
        "CipherSuite": "ECDHE-RSA-AES128-SHA",
        "AuthenticationMethod": "AuthHeader"
```

```
      },
      "requestID": "BD8798EACDD16751",
      "eventID": "607b9532-1423-41c7-b048-ec2641693c47",
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333",
      "tlsDetails": {
            "tlsVersion": "TLSv1.2",
            "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
            "clientProvidedHostHeader": "s3.amazonaws.com"
    }
    },
    {
      "eventVersion": "1.03",
      "userIdentity": {
          "type": "IAMUser",
          "principalId": "111122223333",
          "arn": "arn:aws:iam::111122223333:user/myUserName",
          "accountId": "111122223333",
          "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
          "userName": "myUserName"
        },
      "eventTime": "2019-02-01T03:26:37Z",
      "eventSource": "s3.amazonaws.com",
      "eventName": "GetBucketVersioning",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "",
      "userAgent": "[]",
      "requestParameters": {
          "host": [
              "s3.us-west-2.amazonaws.com"
          ],
          "bucketName": "amzn-s3-demo-bucket1",
          "versioning": [
              ""
          ]
      },
      "responseElements": null,
      "additionalEventData": {
          "SignatureVersion": "SigV4",
          "CipherSuite": "ECDHE-RSA-AES128-SHA",
          "AuthenticationMethod": "AuthHeader"
    },
      "requestID": "07D681279BD94AED",
      "eventID": "f2b287f3-0df1-4961-a2f4-c4bdfed47657",
```

```
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333",
      "tlsDetails": {
            "tlsVersion": "TLSv1.2",
            "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
            "clientProvidedHostHeader": "s3.amazonaws.com"
    }
    }
  ]
}
```

Mengaktifkan pencatatan CloudTrail peristiwa untuk bucket dan objek S3

Anda dapat menggunakan peristiwa CloudTrail data untuk mendapatkan informasi tentang bucket dan permintaan tingkat objek di Amazon S3. Untuk mengaktifkan peristiwa CloudTrail data untuk semua bucket Anda atau untuk daftar bucket tertentu, Anda harus membuat jejak secara manual. CloudTrail

> **Note**
>
> - Pengaturan default untuk CloudTrail adalah untuk menemukan hanya peristiwa manajemen. Periksa untuk memastikan bahwa Anda mengaktifkan peristiwa data untuk akun Anda.
> - Dengan bucket S3 yang menghasilkan beban kerja tinggi, Anda dapat dengan cepat menghasilkan ribuan log dalam waktu singkat. Perhatikan berapa lama Anda memilih untuk mengaktifkan peristiwa CloudTrail data untuk bucket yang sibuk.

CloudTrail menyimpan log peristiwa data Amazon S3 dalam ember S3 pilihan Anda. Pertimbangkan untuk menggunakan bucket secara terpisah Akun AWS untuk mengatur acara dengan lebih baik dari beberapa bucket yang mungkin Anda miliki ke tempat sentral untuk memudahkan kueri dan analisis. AWS Organizations membantu Anda membuat akun Akun AWS yang ditautkan ke akun yang memiliki bucket yang Anda pantau. Untuk informasi lebih lanjut, lihat Apa itu AWS Organizations? dalam AWS Organizations User Guide.

Saat Anda mencatat peristiwa data untuk jejak CloudTrail, Anda dapat memilih untuk menggunakan pemilih peristiwa lanjutan atau pemilih peristiwa dasar untuk mencatat peristiwa data untuk objek yang disimpan dalam bucket tujuan umum. Untuk mencatat peristiwa data untuk objek yang disimpan

dalam bucket direktori, Anda harus menggunakan pemilih peristiwa lanjutan. Untuk informasi selengkapnya, lihat Logging with AWS CloudTrail for S3 Express One Zone.

Saat membuat jejak di CloudTrail konsol menggunakan pemilih peristiwa lanjutan, di bagian peristiwa data, Anda dapat memilih Log semua peristiwa untuk templat pemilih Log untuk mencatat semua peristiwa tingkat objek. Saat Anda membuat jejak di CloudTrail konsol menggunakan pemilih acara dasar, di bagian peristiwa data, Anda dapat memilih kotak centang Pilih semua ember S3 di akun Anda untuk mencatat semua peristiwa tingkat objek.

> **Note**
>
> - Ini adalah praktik terbaik untuk membuat konfigurasi siklus hidup untuk bucket peristiwa AWS CloudTrail data Anda. Konfigurasikan konfigurasi siklus aktif untuk menghapus file log secara berkala setelah periode saat Anda yakin perlu mengauditnya. Hal ini mengurangi jumlah data yang dianalisis oleh Athena untuk setiap kueri. Untuk informasi selengkapnya, lihat Menyetel konfigurasi Siklus Hidup S3 pada bucket.
> - Untuk informasi selengkapnya tentang format pencatatan, lihat Mencatat panggilan API Amazon S3 menggunakan AWS CloudTrail.
> - Untuk contoh cara menanyakan CloudTrail log, lihat posting Blog AWS Big Data Menganalisis Keamanan, Kepatuhan, dan Penggunaan Aktivitas Operasional AWS CloudTrail dan Amazon Athena.

Mengaktifkan pencatatan untuk objek dalam bucket menggunakan konsol tersebut

Anda dapat menggunakan konsol Amazon S3 untuk mengonfigurasi AWS CloudTrail jejak untuk mencatat peristiwa data untuk objek dalam bucket S3. CloudTrail mendukung pencatatan operasi API tingkat objek Amazon S3 seperti`GetObject`,, dan. `DeleteObject` `PutObject` Peristiwa ini disebut peristiwa data.

Secara default, CloudTrail jejak tidak mencatat peristiwa data, tetapi Anda dapat mengonfigurasi jejak untuk mencatat peristiwa data untuk bucket S3 yang Anda tentukan, atau untuk mencatat peristiwa data untuk semua bucket Amazon S3 di bucket Anda. Akun AWS Untuk informasi selengkapnya, lihat Mencatat panggilan API Amazon S3 menggunakan AWS CloudTrail.

CloudTrail tidak mengisi peristiwa data dalam riwayat CloudTrail acara. Selain itu, tidak semua tindakan tingkat ember diisi dalam riwayat acara. CloudTrail Untuk informasi selengkapnya tentang tindakan API tingkat bucket Amazon S3 yang dilacak dengan logging, lihat. CloudTrail Tindakan

tingkat ember Amazon S3 yang dilacak dengan logging CloudTrail Untuk informasi selengkapnya tentang cara menanyakan CloudTrail log, lihat artikel Pusat AWS Pengetahuan tentang penggunaan pola filter CloudWatch Log Amazon dan Amazon Athena untuk menanyakan CloudTrail log.

> **Note**
>
> Jika Anda mencatat aktivitas data dengan AWS CloudTrail, catatan peristiwa untuk peristiwa `DeleteObjects` data Amazon S3 mencakup peristiwa dan `DeleteObjects` `DeleteObject` peristiwa untuk setiap objek yang dihapus sebagai bagian dari operasi tersebut. Anda dapat mengecualikan visibilitas tambahan tentang objek yang dihapus dari catatan peristiwa. Untuk informasi selengkapnya, lihat AWS CLI contoh untuk memfilter peristiwa data di Panduan AWS CloudTrail Pengguna.

Untuk mengonfigurasi jejak guna mencatat peristiwa data bagi bucket S3, Anda dapat menggunakan konsol AWS CloudTrail tersebut atau konsol Amazon S3. Jika Anda mengonfigurasi jejak untuk mencatat peristiwa data untuk semua bucket Amazon S3 di Akun AWS Anda, lebih mudah menggunakan konsol. CloudTrail Untuk informasi tentang menggunakan CloudTrail konsol untuk mengonfigurasi jejak untuk mencatat peristiwa data S3, lihat Peristiwa data di Panduan AWS CloudTrail Pengguna.

> **Important**
>
> Biaya tambahan berlaku untuk peristiwa data. Untuk informasi selengkapnya, lihat harga AWS CloudTrail.

Prosedur berikut menunjukkan cara menggunakan konsol Amazon S3 untuk mengonfigurasi CloudTrail jejak untuk mencatat peristiwa data untuk bucket S3.

Untuk mengaktifkan pencatatan peristiwa CloudTrail data untuk objek di bucket tujuan umum S3 atau di bucket direktori S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di daftar Bucket, pilih nama bucket.
3. Pilih Properti.
4. Di bawah peristiwa AWS CloudTrail data, pilih Konfigurasikan dalam CloudTrail.

Anda dapat membuat CloudTrail jejak baru atau menggunakan kembali jejak yang ada dan mengonfigurasi peristiwa data Amazon S3 untuk dicatat di jejak Anda. Untuk informasi tentang cara membuat jejak di CloudTrail konsol, lihat Membuat dan memperbarui jejak dengan konsol di Panduan AWS CloudTrail Pengguna. Untuk informasi tentang cara mengonfigurasi pencatatan peristiwa data Amazon S3 di CloudTrail konsol, lihat Mencatat peristiwa data untuk Objek Amazon S3 di AWS CloudTrail Panduan Pengguna.

> **Note**
>
> Jika Anda menggunakan CloudTrail konsol atau konsol Amazon S3 untuk mengonfigurasi jejak untuk mencatat peristiwa data untuk bucket S3, konsol Amazon S3 menunjukkan bahwa pencatatan tingkat objek diaktifkan untuk bucket.

Untuk menonaktifkan pencatatan peristiwa CloudTrail data untuk objek dalam bucket S3

1. Masuk ke AWS Management Console dan buka CloudTrail konsol di https://console.aws.amazon.com/cloudtrail/.
2. Di panel navigasi kiri, pilih Jalur.
3. Pilih nama jejak yang Anda buat untuk peristiwa log untuk bucket Anda.
4. Di halaman detail untuk jejak Anda, pilih Hentikan pencatatan di sudut kanan atas.
5. Pada kotak dialog yang muncul, pilih Hentikan pencatatan.

Untuk informasi tentang mengaktifkan pencatatan tingkat objek saat Anda membuat bucket S3, lihat Membuat ember tujuan umum.

Untuk informasi selengkapnya tentang CloudTrail logging dengan bucket S3, lihat topik berikut:

- Melihat properti untuk bucket tujuan umum S3
- Mencatat panggilan API Amazon S3 menggunakan AWS CloudTrail
- Bekerja dengan File CloudTrail Log di Panduan AWS CloudTrail Pengguna

Mengidentifikasi permintaan Amazon S3 menggunakan CloudTrail

Di Amazon S3, Anda dapat mengidentifikasi permintaan menggunakan log AWS CloudTrail peristiwa. AWS CloudTrail adalah cara yang lebih disukai untuk mengidentifikasi permintaan Amazon S3, tetapi

jika Anda menggunakan log akses server Amazon S3, lihat. the section called “Mengidentifikasi permintaan S3”

Topik

- Mengidentifikasi permintaan yang dibuat ke Amazon S3 dalam log CloudTrail
- Mengidentifikasi permintaan Amazon S3 Signature Version 2 dengan menggunakan CloudTrail
- Mengidentifikasi akses ke objek S3 dengan menggunakan CloudTrail

Mengidentifikasi permintaan yang dibuat ke Amazon S3 dalam log CloudTrail

Setelah mengatur CloudTrail untuk mengirimkan acara ke bucket, Anda akan mulai melihat objek masuk ke bucket tujuan Anda di konsol Amazon S3. Ini diformat sebagai berikut:

```
s3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/Region/yyyy/mm/dd
```

Peristiwa yang dicatat oleh CloudTrail disimpan sebagai objek gzipped JSON terkompresi di bucket S3 Anda. Untuk menemukan permintaan secara efisien, Anda harus menggunakan layanan seperti Amazon Athena untuk mengindeks dan menanyakan log. CloudTrail

Untuk informasi selengkapnya tentang CloudTrail dan Athena, lihat Membuat tabel untuk AWS CloudTrail log di Athena menggunakan proyeksi partisi di Panduan Pengguna Amazon Athena.

Mengidentifikasi permintaan Amazon S3 Signature Version 2 dengan menggunakan CloudTrail

Anda dapat menggunakan log CloudTrail peristiwa untuk mengidentifikasi versi tanda tangan API mana yang digunakan untuk menandatangani permintaan di Amazon S3. Kemampuan ini penting karena dukungan untuk Signature Version 2 akan dinonaktifkan (dihentikan). Setelah itu, Amazon S3 tidak akan menerima lagi permintaan yang menggunakan Signature Version 2, dan semua permintaan harus menggunakan penandatanganan Signature Version 4.

Kami sangat menyarankan agar Anda menggunakannya CloudTrail untuk membantu menentukan apakah alur kerja Anda menggunakan penandatanganan Signature Version 2. Perbaiki dengan meningkatkan pustaka dan kode untuk menggunakan Signature Version 4 guna mencegah dampaknya pada bisnis Anda.

Untuk informasi selengkapnya, lihat Pengumuman: AWS CloudTrail untuk Amazon S3 menambahkan bidang baru untuk audit keamanan yang ditingkatkan. AWS re:Post

> **Note**
>
> CloudTrail peristiwa untuk Amazon S3 menyertakan versi tanda tangan dalam detail permintaan dengan nama kunci '. `additionalEventData` Untuk menemukan versi tanda tangan pada permintaan yang dibuat untuk objek di Amazon S3 sepertiGET,, dan DELETE permintaanPUT, Anda harus mengaktifkan peristiwa CloudTrail data. (Fitur ini dimatikan secara default.)

AWS CloudTrail adalah metode yang disukai untuk mengidentifikasi permintaan Signature Version 2. Jika Anda menggunakan pencatatan akses server Amazon S3, lihat Mengidentifikasi permintaan Signature Version 2 menggunakan pencatatan akses Amazon S3.

Topik

- Contoh kueri Athena untuk mengidentifikasi permintaan Tanda Tangan versi 2 Amazon S3
- Mempartisikan data Signature Version 2

Contoh kueri Athena untuk mengidentifikasi permintaan Tanda Tangan versi 2 Amazon S3

Example — Pilih semua peristiwa Signature Version 2, dan hanya cetak **`EventTime`**, **`S3_Action`**, **`Request_Parameters`**, **`Region`**, **`SourceIP`**, dan **`UserAgent`**

Pada kueri Athena berikut, ganti *`s3_cloudtrail_events_db.cloudtrail_table`* dengan detail Athena Anda, dan menambahkan atau menghapus batasnya sesuai dengan kebutuhan.

```
SELECT EventTime, EventName as S3_Action, requestParameters as Request_Parameters,
 awsregion as AWS_Region, sourceipaddress as Source_IP, useragent as User_Agent
FROM s3_cloudtrail_events_db.cloudtrail_table
WHERE eventsource='s3.amazonaws.com'
AND json_extract_scalar(additionalEventData, '$.SignatureVersion')='SigV2'
LIMIT 10;
```

Example — Pilih semua peminta yang mengirimkan lalu lintas Signature Version 2

```
SELECT useridentity.arn, Count(requestid) as RequestCount
FROM s3_cloudtrail_events_db.cloudtrail_table
```

```
WHERE eventsource='s3.amazonaws.com'
    and json_extract_scalar(additionalEventData, '$.SignatureVersion')='SigV2'
Group by useridentity.arn
```

Mempartisikan data Signature Version 2

Jika Anda memiliki data dalam jumlah large untuk kueri, Anda dapat meredam biaya dan runtime Athena dengan membuat tabel partisi.

Untuk melakukannya, buat tabel baru dengan partisi sebagai berikut.

```
CREATE EXTERNAL TABLE s3_cloudtrail_events_db.cloudtrail_table_partitioned(
     eventversion STRING,
     userIdentity STRUCT<
         type:STRING,
         principalid:STRING,
         arn:STRING,
         accountid:STRING,
         invokedby:STRING,
         accesskeyid:STRING,
         userName:STRING,
       sessioncontext:STRUCT<
                  attributes:STRUCT<
                  mfaauthenticated:STRING,
                  creationdate:STRING>,
                  sessionIssuer:STRUCT<
                  type:STRING,
                  principalId:STRING,
                  arn:STRING,
                  accountId:STRING,
                  userName:STRING>
             >
          >,
     eventTime STRING,
     eventSource STRING,
     eventName STRING,
     awsRegion STRING,
     sourceIpAddress STRING,
     userAgent STRING,
     errorCode STRING,
     errorMessage STRING,
     requestParameters STRING,
```

```
        responseElements STRING,
        additionalEventData STRING,
        requestId STRING,
        eventId STRING,
        resources ARRAY<STRUCT<ARN:STRING,accountId: STRING,type:STRING>>,
        eventType STRING,
        apiVersion STRING,
        readOnly STRING,
        recipientAccountId STRING,
        serviceEventDetails STRING,
        sharedEventID STRING,
        vpcEndpointId STRING
    )
    PARTITIONED BY (region string, year string, month string, day string)
    ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.orc.OrcSerde'
    STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.SymlinkTextInputFormat'
    OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
    LOCATION 's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/';
```

Lalu, buat partisi secara terpisah. Anda tidak bisa mendapatkan hasil dari tanggal yang belum Anda buat.

```
ALTER TABLE s3_cloudtrail_events_db.cloudtrail_table_partitioned ADD
    PARTITION (region= 'us-east-1', year= '2019', month= '02', day= '19') LOCATION
 's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/us-east-1/2019/02/19/'
    PARTITION (region= 'us-west-1', year= '2019', month= '02', day= '19') LOCATION
 's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/us-west-1/2019/02/19/'
    PARTITION (region= 'us-west-2', year= '2019', month= '02', day= '19') LOCATION
 's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/us-west-2/2019/02/19/'
    PARTITION (region= 'ap-southeast-1', year= '2019', month= '02', day= '19') LOCATION
 's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/ap-southeast-1/2019/02/19/'
    PARTITION (region= 'ap-southeast-2', year= '2019', month= '02', day= '19') LOCATION
 's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/ap-southeast-2/2019/02/19/'
    PARTITION (region= 'ap-northeast-1', year= '2019', month= '02', day= '19') LOCATION
 's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/ap-northeast-1/2019/02/19/'
    PARTITION (region= 'eu-west-1', year= '2019', month= '02', day= '19') LOCATION
 's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/eu-west-1/2019/02/19/'
    PARTITION (region= 'sa-east-1', year= '2019', month= '02', day= '19') LOCATION
 's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/sa-east-1/2019/02/19/';
```

Kemudian, Anda dapat membuat permintaan berdasarkan partisi ini, dan Anda tidak perlu memuat seluruh bucket.

```
SELECT useridentity.arn,
Count(requestid) AS RequestCount
FROM s3_cloudtrail_events_db.cloudtrail_table_partitioned
WHERE eventsource='s3.amazonaws.com'
AND json_extract_scalar(additionalEventData, '$.SignatureVersion')='SigV2'
AND region='us-east-1'
AND year='2019'
AND month='02'
AND day='19'
Group by useridentity.arn
```

Mengidentifikasi akses ke objek S3 dengan menggunakan CloudTrail

Anda dapat menggunakan log AWS CloudTrail peristiwa untuk mengidentifikasi permintaan akses objek Amazon S3 untuk peristiwa data sepertiGetObject,, dan DeleteObjectPutObject, dan menemukan informasi tambahan tentang permintaan tersebut.

> **Note**
>
> Jika Anda mencatat aktivitas data dengan AWS CloudTrail, catatan peristiwa untuk peristiwa DeleteObjects data Amazon S3 mencakup peristiwa dan DeleteObjects DeleteObject peristiwa untuk setiap objek yang dihapus sebagai bagian dari operasi tersebut. Anda dapat mengecualikan visibilitas tambahan tentang objek yang dihapus dari catatan peristiwa. Untuk informasi selengkapnya, lihat AWS CLI contoh untuk memfilter peristiwa data di Panduan AWS CloudTrail Pengguna.

Contoh berikut menunjukkan cara mendapatkan semua permintaan PUT objek untuk Amazon S3 dari log AWS CloudTrail peristiwa.

Topik

- Contoh kueri Athena untuk mengidentifikasi permintaan akses objek Amazon S3

Contoh kueri Athena untuk mengidentifikasi permintaan akses objek Amazon S3

Pada kueri Athena berikut, ganti *s3_cloudtrail_events_db.cloudtrail_table* dengan detail Athena Anda, dan modifikasi rentang tanggal sesuai kebutuhan.

Example — Pilih semua peristiwa yang memiliki permintaan akses **PUT** objek, dan hanya cetak **EventTime**, **EventSource**, **SourceIP**, **UserAgent**, **BucketName**, **object**, dan **UserARN**

```
SELECT
  eventTime,
  eventName,
  eventSource,
  sourceIpAddress,
  userAgent,
  json_extract_scalar(requestParameters, '$.bucketName') as bucketName,
  json_extract_scalar(requestParameters, '$.key') as object,
  userIdentity.arn as userArn
FROM
  s3_cloudtrail_events_db.cloudtrail_table
WHERE
  eventName = 'PutObject'
  AND eventTime BETWEEN '2019-07-05T00:00:00Z' and '2019-07-06T00:00:00Z'
```

Example — Pilih semua peristiwa yang memiliki permintaan akses objek **GET**, kemudian hanya cetak untuk **EventTime**, **EventSource**, **SourceIP**, **UserAgent**, **BucketName**, **object**, dan **UserARN**

```
SELECT
  eventTime,
  eventName,
  eventSource,
  sourceIpAddress,
  userAgent,
  json_extract_scalar(requestParameters, '$.bucketName') as bucketName,
  json_extract_scalar(requestParameters, '$.key') as object,
  userIdentity.arn as userArn
FROM
  s3_cloudtrail_events_db.cloudtrail_table
WHERE
  eventName = 'GetObject'
  AND eventTime BETWEEN '2019-07-05T00:00:00Z' and '2019-07-06T00:00:00Z'
```

Example — Pilih semua peristiwa pemohon anonim ke bucket dalam periode tertentu dan hanya cetak **`EventTime`**, **`EventName`**, **`EventSource`**, **`SourceIP`**, **`UserAgent`**, **`BucketName`**, **`UserARN`**, dan **`AccountID`**

```
SELECT
  eventTime,
  eventName,
  eventSource,
  sourceIpAddress,
  userAgent,
  json_extract_scalar(requestParameters, '$.bucketName') as bucketName,
  userIdentity.arn as userArn,
  userIdentity.accountId
FROM
  s3_cloudtrail_events_db.cloudtrail_table
WHERE
  userIdentity.accountId = 'anonymous'
  AND eventTime BETWEEN '2019-07-05T00:00:00Z' and '2019-07-06T00:00:00Z'
```

Example — Identifikasi semua permintaan yang memerlukan ACL untuk otorisasi

Contoh kueri Amazon Athena berikut menunjukkan cara mengidentifikasi semua permintaan ke bucket S3 Anda yang memerlukan daftar kontrol akses (ACL) untuk otorisasi. Jika permintaan memerlukan ACL untuk otorisasi, `aclRequired` nilai dalam `additionalEventData` adalah `Yes`. Jika ACLs tidak diperlukan, `aclRequired` tidak ada. Anda dapat menggunakan informasi ini untuk memigrasikan izin ACL tersebut ke kebijakan bucket yang sesuai. Setelah membuat kebijakan bucket ini, Anda dapat ACLs menonaktifkan bucket ini. Untuk informasi selengkapnya tentang menonaktifkan ACLs, lihat. Prasyarat untuk menonaktifkan ACLs

```
SELECT
  eventTime,
  eventName,
  eventSource,
  sourceIpAddress,
  userAgent,
  userIdentity.arn as userArn,
  json_extract_scalar(requestParameters, '$.bucketName') as bucketName,
  json_extract_scalar(requestParameters, '$.key') as object,
  json_extract_scalar(additionalEventData, '$.aclRequired') as aclRequired
FROM
  s3_cloudtrail_events_db.cloudtrail_table
```

```
WHERE
  json_extract_scalar(additionalEventData, '$.aclRequired') = 'Yes'
  AND eventTime BETWEEN '2022-05-10T00:00:00Z' and '2022-08-10T00:00:00Z'
```

> **Note**
>
> - Contoh kueri ini juga dapat berguna untuk pemantauan keamanan. Anda dapat meninjau hasil untuk panggilan `PutObject` atau `GetObject` dari alamat IP yang tidak terduga atau tidak sah atau pemohon, dan untuk mengidentifikasi permintaan anonim ke bucket Anda.
> - Kueri ini hanya mengambil informasi dari waktu saat pencatatan log diaktifkan.

Jika Anda menggunakan log akses server Amazon S3, lihat Mengidentifikasi permintaan akses objek dengan menggunakan pencatatan akses Amazon S3.

Pencatatan permintaan dengan pencatatan akses server

Pencatatan akses server menyediakan catatan terperinci untuk permintaan yang dilakukan ke bucket. Log akses server bermanfaat untuk berbagai macam aplikasi. Misalnya, informasi log akses dapat berguna dalam audit keamanan dan akses. Informasi ini juga dapat membantu Anda untuk mempelajari basis pelanggan Anda, serta memahami tagihan Amazon S3.

> **Note**
>
> Log akses server tidak mencatat informasi tentang kesalahan pengalihan wilayah yang salah untuk Wilayah yang diluncurkan setelah 20 Maret 2019. Kesalahan pengalihan wilayah yang salah terjadi saat permintaan objek atau bucket dibuat di luar Wilayah tempat bucket berada.

Bagaimana cara mengaktifkan log pengiriman?

Untuk mengaktifkan log pengiriman, lakukan langkah-langkah basic berikut. Lihat perinciannya di Mengaktifkan pencatatan akses server Amazon S3.

1. Berikan nama bucket tujuan (juga dikenal sebagai bucket target). Bucket ini adalah tempat Anda ingin Amazon S3 menyimpan pencatatan akses sebagai objek. Bucket sumber dan tujuan harus berada di Wilayah AWS yang sama, dan dimiliki oleh akun yang sama. Bucket tujuan tidak

boleh memiliki konfigurasi periode retensi default Kunci Objek S3. Bucket tujuan juga harus tidak mengaktifkan Requester Pays.

Anda dapat mengirimkan log ke bucket mana pun milik Anda yang berada di Wilayah yang sama dengan bucket sumber, termasuk bucket sumber itu sendiri. Tetapi untuk manajemen log yang lebih sederhana, kami sarankan agar Anda menyimpan log akses dalam bucket yang berbeda.

Saat bucket sumber dan bucket tujuan Anda merupakan bucket yang sama, akan dibuat log tambahan untuk log yang ditulis ke bucket, yang akan membuat loop log tak terbatas. Kami tidak menyarankan melakukan hal ini karena dapat menyebabkan sedikit peningkatan dalam penagihan penyimpanan Anda. Selain itu, catatan tambahan tentang log mungkin akan menyulitkan Anda menemukan log yang Anda cari.

Jika Anda memilih untuk menyimpan log akses di bucket sumber, sebaiknya Anda menentukan prefiks tujuan (juga dikenal sebagai prefiks target) untuk semua kunci objek log. Saat Anda menentukan prefiks, semua nama objek log dimulai dengan string umum, yang membuat objek log lebih mudah diidentifikasi.

2. (Opsional) Tetapkan prefiks tujuan ke semua kunci objek log Amazon S3. Awalan tujuan (juga dikenal sebagai prefiks target) menjadikan pencarian objek log lebih mudah. Misalnya, jika Anda menentukan nilai prefiks`logs/`, setiap objek log yang dibuat Amazon S3 dimulai dengan prefiks `logs/` di dalam kuncinya, misalnya:

```
logs/2013-11-01-21-32-16-E568B2907131C0C0
```

Jika Anda menentukan nilai prefiks `logs`, objek log muncul sebagai berikut:

```
logs2013-11-01-21-32-16-E568B2907131C0C0
```

Prefiks juga berguna untuk membedakan bucket sumber jika beberapa bucket mencatat ke bucket tujuan yang sama.

Prefiks ini juga dapat membantu saat Anda menghapus pencatatan. Misalnya, Anda dapat menetapkan aturan konfigurasi siklus aktif bagi Amazon S3 untuk menghapus objek dengan prefiks kunci tertentu. Untuk informasi selengkapnya, lihat Menghapus file log Amazon S3.

3. (Opsional) Atur izin sehingga orang lain dapat mengakses log yang dihasilkan. Secara default, hanya pemilik bucket yang selalu memiliki akses penuh ke objek log. Jika bucket tujuan menggunakan setelan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek S3 untuk

menonaktifkan daftar kontrol akses (ACLs), Anda tidak dapat memberikan izin dalam hibah tujuan (juga dikenal sebagai hibah target) yang digunakan. ACLs Namun, Anda dapat memperbarui kebijakan bucket untuk bucket tujuan untuk memberikan akses kepada orang lain. Untuk informasi selengkapnya, lihat Identity and Access Management untuk Amazon S3 dan Izin untuk pengiriman log.

4. (Opsional) Tetapkan format kunci objek log untuk file log. Anda memiliki dua opsi untuk format kunci objek log (juga dikenal sebagai format kunci objek target):

   - Non-date-based partisi - Ini adalah format kunci objek log asli. Jika Anda memilih format ini, format kunci file log muncul sebagai berikut:

     ```
     [DestinationPrefix][YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
     ```

     Misalnya, jika Anda menentukan `logs/` sebagai prefiks, objek log Anda diberi nama seperti ini:

     ```
     logs/2013-11-01-21-32-16-E568B2907131C0C0
     ```

   - Partisi berbasis tanggal–Jika Anda memilih partisi berbasis tanggal, Anda dapat memilih waktu peristiwa atau waktu pengiriman untuk file log sebagai sumber tanggal yang digunakan dalam format log. Format ini membuatnya lebih mudah untuk mengkueri log.

     Jika Anda memilih partisi berbasis tanggal, format kunci file log akan muncul sebagai berikut:

     ```
     [DestinationPrefix][SourceAccountId]/[SourceRegion]/[SourceBucket]/[YYYY]/[MM]/
     [DD]/[YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
     ```

     Misalnya, jika Anda menentukan `logs/` sebagai prefiks target, objek log Anda diberi nama seperti ini:

     ```
     logs/123456789012/us-west-2/amzn-s3-demo-source-
     bucket/2023/03/01/2023-03-01-21-32-16-E568B2907131C0C0
     ```

     Untuk pengiriman waktu pengiriman, waktu dalam nama file log sesuai dengan waktu pengiriman untuk file log.

     Untuk pengiriman waktu peristiwa, tahun, bulan, dan hari sesuai dengan hari di mana peristiwa itu terjadi, dan jam, menit dan detik diatur ke `00` dalam kunci. Log yang dikirimkan dalam file log ini hanya untuk hari tertentu.

Jika Anda mengonfigurasi log Anda melalui AWS Command Line Interface (AWS CLI) AWS SDKs, atau Amazon S3 REST API, `TargetObjectKeyFormat` gunakan untuk menentukan format kunci objek log. Untuk menentukan non-date-based partisi, gunakan. `SimplePrefix` Untuk menentukan partisi berbasis data, gunakan `PartitionedPrefix`. Jika Anda menggunakan`PartitionedPrefix`, gunakan `PartitionDateSource` untuk menentukan antara `EventTime`, atau `DeliveryTime`.

Untuk `SimplePrefix`, format kunci file log muncul sebagai berikut:

```
[TargetPrefix][YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
```

Untuk `PartitionedPrefix` dengan waktu peristiwa atau waktu pengiriman, format kunci file log muncul sebagai berikut:

```
[TargetPrefix][SourceAccountId]/[SourceRegion]/[SourceBucket]/[YYYY]/[MM]/[DD]/
[YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
```

Format kunci objek log

Amazon S3 menggunakan format kunci objek berikut untuk objek log yang diunggahnya di bucket tujuan:

- Non-date-based partisi - Ini adalah format kunci objek log asli. Jika Anda memilih format ini, format kunci file log muncul sebagai berikut:

  ```
  [DestinationPrefix][YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
  ```

- Partisi berbasis tanggal–Jika Anda memilih partisi berbasis tanggal, Anda dapat memilih waktu peristiwa atau waktu pengiriman untuk file log sebagai sumber tanggal yang digunakan dalam format log. Format ini membuatnya lebih mudah untuk mengkueri log.

  Jika Anda memilih partisi berbasis tanggal, format kunci file log akan muncul sebagai berikut:

  ```
  [DestinationPrefix][SourceAccountId]/[SourceRegion]/[SourceBucket]/[YYYY]/[MM]/[DD]/
  [YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
  ```

Dalam kunci objek log, YYYY, MM, DD, hh, mm, dan ss merupakan digit tahun, bulan, hari, jam, menit, dan detik (masing-masing). Tanggal dan waktu ini berada dalam Waktu Universal Terkoordinasi (UTC).

Berkas log yang dikirimkan pada waktu tertentu dapat berisi catatan yang ditulis kapan pun sebelum waktu tersebut. Tidak ada cara untuk mengetahui apakah semua catatan log untuk interval waktu tertentu telah dikirim atau tidak.

Komponen UniqueString pada kunci ada untuk mencegah penimpaan file. Tidak memiliki makna, dan perangkat lunak pemroses log harus mengabaikannya.

Bagaimana log dikirimkan?

Amazon S3 secara berkala mengumpulkan catatan akses, mengonsolidasikan catatan dalam file log, kemudian mengunggah file log ke bucket tujuan sebagai objek log. Jika Anda mengaktifkan pencatatan log ke beberapa bucket sumber yang mengidentifikasi bucket tujuan yang sama, bucket tujuan akan memiliki catatan akses untuk semua bucket sumber tersebut. Namun demikian, setiap objek catatan melaporkan arsip log akses untuk bucket sumber spesifik.

Amazon S3 menggunakan akun pengiriman log kustom untuk menulis log akses server. Penulisan ini tunduk pada pembatasan kontrol akses biasa. Kami menyarankan Anda memperbarui kebijakan bucket pada bucket tujuan untuk memberikan akses ke pengguna utama layanan pencatatan (logging.s3.amazonaws.com) kepada pengiriman log akses. Anda juga dapat memberikan akses kepada pengiriman log akses ke grup pengiriman log S3 melalui daftar kontrol akses (ACL) bucket Anda. Namun, pemberian akses ke grup pengiriman log S3 dengan menggunakan bucket ACL Anda tidak disarankan.

Saat mengaktifkan pencatatan akses server dan memberikan akses untuk pengiriman log akses melalui kebijakan bucket tujuan, Anda harus memperbarui kebijakan untuk mengizinkan akses s3:PutObject bagi pengguna utama layanan pencatatan. Jika Anda menggunakan konsol Amazon S3 untuk mengaktifkan pencatatan akses server, konsol akan secara otomatis memperbarui kebijakan bucket tujuan untuk memberikan izin ini kepada pengguna utama layanan logging. Untuk informasi selengkapnya tentang pemberian izin untuk pengiriman log akses server, lihat Izin untuk pengiriman log.

> **Note**
>
> S3 tidak mendukung pengiriman CloudTrail log atau log akses server ke pemohon atau pemilik bucket untuk permintaan titik akhir VPC saat kebijakan titik akhir VPC menyangkalnya atau untuk permintaan yang gagal sebelum kebijakan VPC dievaluasi.

Pengaturan yang diberlakukan pemilik bucket untuk Kepemilikan Objek S3

Jika bucket tujuan menggunakan setelan diberlakukan pemilik Bucket untuk Kepemilikan Objek, akan ACLs dinonaktifkan dan tidak lagi memengaruhi izin. Anda harus memperbarui kebijakan bucket pada bucket tujuan untuk memberikan akses ke pengguna utama layanan logging. Untuk informasi selengkapnya tentang Kepemilikan Objek, lihat Berikan akses ke grup pengiriman log S3 untuk pencatatan akses server.

Pengiriman log server dengan upaya terbaik

Catatan log akses server disampaikan atas dasar upaya terbaik. Sebagian besar permintaan bucket yang dikonfigurasi dengan benar untuk mencatat hasil dalam catatan log yang dikirim. Sebagian besar catatan log dikirim dalam beberapa jam setelah log dicatat, tetapi dapat dikirimkan lebih sering.

Kelengkapan dan ketepatan waktu pencatatan server tidak dijamin. Catatan log untuk permintaan tertentu mungkin dikirim dalam waktu lama setelah permintaan diproses, atau mungkin tidak dikirimkan sama sekali. Ada kemungkinan bahwa Anda bahkan mungkin melihat duplikasi catatan log. Tujuan log server adalah untuk memberikan Anda ide tentang sifat lalu lintas terhadap bucket Anda. Meskipun catatan akuntasi log jarang hilang atau digandakan, perlu diketahui bahwa pencatatan akuntasi server tidak dimaksudkan untuk menjadi pencatatan akuntasi lengkap atas semua permintaan.

Karena sifat upaya terbarik dari pencatatan log, laporan penggunaan Anda dapat mencakup satu permintaan akses atau lebih yang tidak muncul di log server yang dikirim. Anda dapat menemukan laporan penggunaan ini di bawah Laporan biaya & penggunaan di konsol AWS Manajemen Penagihan dan Biaya .

Perubahan status pencatatan log bucket memerlukan waktu

Perubahan status pencatatan log pada bucket memerlukan waktu untuk benar-benar memengaruhi pengiriman file log. Misalnya, jika Anda mengaktifkan pencatatan log untuk bucket, beberapa

permintaan yang dilakukan di jam berikutnya mungkin akan dicatat, dan yang lainnya mungkin tidak. Misalkan Anda mengubah bucket tujuan untuk pencatatan log dari bucket A ke bucket B. Untuk jam berikutnya, beberapa catatan mungkin akan terus dikirimkan ke bucket A, sedangkan yang lain mungkin dikirimkan ke bucket tujuan B baru. Dalam semua kasus, pengaturan baru tersebut pada akhirnya akan berlaku tanpa tindakan lebih lanjut dari pihak Anda.

Untuk informasi lebih lanjut tentang pencatatan dan file log, lihat bagian berikut:

Topik

- Mengaktifkan pencatatan akses server Amazon S3
- Server Amazon S3 mengakses format log
- Menghapus file log Amazon S3
- Menggunakan log akses server Amazon S3 untuk mengidentifikasi permintaan
- Memecahkan masalah pencatatan akses server

Mengaktifkan pencatatan akses server Amazon S3

Pencatatan akses server menyediakan catatan terperinci untuk permintaan yang dilakukan ke bucket Amazon S3 Anda. Log akses server bermanfaat untuk berbagai macam aplikasi. Misalnya, informasi log akses dapat berguna dalam audit keamanan dan akses. Informasi ini juga dapat membantu Anda untuk mempelajari basis pelanggan Anda, serta memahami tagihan Amazon S3.

Secara default, Amazon S3 tidak mengumpulkan log akses server. Saat Anda mengaktifkan pencatatan log, Amazon S3 mengirimkan pencatatan akses untuk bucket sumber ke bucket tujuan (juga dikenal sebagai bucket target) yang Anda pilih. Bucket tujuan harus sama Wilayah AWS dan Akun AWS sebagai ember sumber.

Catatan log akses berisi detail tentang permintaan yang dilakukan ke bucket. Informasi ini dapat mencakup jenis permintaan, sumber daya yang ditentukan dalam permintaan, dan waktu serta tanggal pemrosesan permintaan. Untuk informasi lebih lanjut tentang dasar pencatatan log, lihat Pencatatan permintaan dengan pencatatan akses server.

> **Important**
>
> - Tidak ada biaya tambahan untuk mengaktifkan pencatatan akses server di bucket Amazon S3. Namun, semua file log yang dikirim sistem kepada Anda akan dikenakan biaya penyimpanan yang biasa. (Anda dapat menghapus file log ini kapan saja.) Kami tidak

menilai biaya transfer data untuk pengiriman file log, tetapi kami memang mengenakan tarif transfer data normal untuk mengakses file log.

- bucket tujuan Anda seharusnya tidak memiliki akses server pencatatan yang diaktifkan. Anda dapat mengirimkan log ke bucket mana pun milik Anda yang berada di Wilayah yang sama dengan bucket sumber, termasuk bucket sumber itu sendiri. Namun, mengirimkan log ke bucket sumber akan menyebabkan loop log yang tak terbatas dan tidak disarankan. Tetapi untuk manajemen log yang lebih sederhana, kami sarankan agar Anda menyimpan log akses dalam bucket yang berbeda. Untuk informasi selengkapnya, lihat Bagaimana cara mengaktifkan log pengiriman?
- Bucket S3 yang mengaktifkan Kunci Objek S3 tidak dapat digunakan sebagai bucket tujuan untuk pencatatan akses server. Bucket tujuan Anda tidak boleh memiliki konfigurasi periode retensi default.
- Bucket tujuan tidak boleh mengaktifkan Pembayaran oleh Pemohon.
- Anda dapat menggunakan enkripsi bucket default pada bucket tujuan hanya jika Anda menggunakan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3), yang menggunakan Standar Enkripsi Lanjutan 256 bit (AES-256). Enkripsi sisi server default dengan kunci AWS Key Management Service (AWS KMS) (SSE-KMS) tidak didukung.

Anda dapat mengaktifkan atau menonaktifkan pencatatan akses server dengan menggunakan konsol Amazon S3, Amazon S3 API,AWS CLI(), AWS Command Line Interface atau. AWS SDKs

Izin untuk pengiriman log

Amazon S3 menggunakan akun pengiriman log kustom untuk menulis log akses server. Penulisan ini tunduk pada pembatasan kontrol akses biasa. Untuk pengiriman log akses, Anda harus memberikan akses kepada pengguna utama layanan pencatatan (`logging.s3.amazonaws.com`) ke bucket tujuan Anda.

Untuk memberikan izin ke Amazon S3 untuk pengiriman log, Anda dapat menggunakan kebijakan bucket atau daftar kontrol akses bucket ACLs (), tergantung pada setelan Kepemilikan Objek S3 bucket tujuan. Namun, kami menyarankan Anda menggunakan kebijakan bucket sebagai gantinya ACLs.

Pengaturan yang diberlakukan pemilik bucket untuk Kepemilikan Objek S3

Jika bucket tujuan menggunakan setelan diberlakukan pemilik Bucket untuk Kepemilikan Objek, akan ACLs dinonaktifkan dan tidak lagi memengaruhi izin. Dalam hal ini, Anda harus memperbarui

kebijakan bucket untuk bucket tujuan untuk memberikan akses ke pengguna utama layanan logging. Anda tidak dapat memperbarui ACL bucket untuk memberikan akses ke grup pengiriman log S3. Anda juga tidak dapat menyertakan pemberian tujuan (juga dikenal sebagai pemberian target) dalam konfigurasi PutBucketLogging Anda.

Untuk informasi tentang memigrasi bucket yang ada ACLs untuk pengiriman log akses ke kebijakan bucket, lihatBerikan akses ke grup pengiriman log S3 untuk pencatatan akses server. Untuk informasi selengkapnya tentang Kepemilikan Objek, lihat Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda. Saat Anda membuat bucket baru, dinonaktifkan ACLs secara default.

memberikan akses dengan menggunakan kebijakan bucket

Untuk memberikan akses dengan menggunakan kebijakan bucket di bucket tujuan, perbarui kebijakan bucket untuk memberikan izin `s3:PutObject` kepada pengguna utama layanan logging. Jika Anda menggunakan konsol Amazon S3 untuk mengaktifkan pencatatan akses server, konsol secara otomatis memperbarui kebijakan bucket pada bucket tujuan untuk memberikan izin ini kepada pengguna utama layanan pencatatan. Jika Anda mengaktifkan pencatatan akses server secara terprogram, Anda harus memperbarui kebijakan bucket secara manual untuk bucket tujuan untuk memberikan akses ke pengguna utama layanan logging.

Untuk contoh kebijakan bucket yang memberikan akses ke pengguna utama layanan logging, lihat the section called “Berikan izin kepada pengguna utama layanan logging dengan menggunakan kebijakan bucket”.

Memberikan akses dengan menggunakan bucket ACLs

Anda dapat menggunakan bucket secara bergantian ACLs untuk memberikan akses untuk pengiriman log akses. Anda menambahkan entri pemberian ke ACL bucket yang memberikan izin `WRITE` dan `READ_ACP` ke grup pengiriman log S3. Namun, pemberian akses ke grup pengiriman log S3 dengan menggunakan bucket tidak ACLs disarankan. Untuk informasi selengkapnya, lihat Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda. Untuk informasi tentang memigrasi bucket yang ada ACLs untuk pengiriman log akses ke kebijakan bucket, lihatBerikan akses ke grup pengiriman log S3 untuk pencatatan akses server. Untuk contoh ACL yang memberikan akses ke pengguna utama layanan logging, lihat the section called “Berikan izin ke grup pengiriman log dengan menggunakan bucket ACL”.

Berikan izin kepada pengguna utama layanan logging dengan menggunakan kebijakan bucket

Contoh kebijakan bucket ini memberikan `s3:PutObject` izin kepada pengguna utama layanan logging (`logging.s3.amazonaws.com`). Untuk menggunakan kebijakan bucket ini, ganti *`user`*

input placeholders dengan informasi Anda sendiri. Dalam kebijakan berikut, *amzn-s3-demo-destination-bucket* adalah bucket tujuan tempat log akses server akan dikirimkan, dan *amzn-s3-demo-source-bucket* merupakan bucket sumber. *EXAMPLE-LOGGING-PREFIX*adalah awalan tujuan opsional (juga dikenal sebagai awalan target) yang ingin Anda gunakan untuk objek log Anda. *SOURCE-ACCOUNT-ID*adalah Akun AWS yang memiliki ember sumber.

> **Note**
>
> Jika terdapat pernyataan Deny dalam kebijakan bucket Anda, pastikan pernyataan tersebut tidak mencegah Amazon S3 mengirimkan pencatatan akses.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3ServerAccessLogsPolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "logging.s3.amazonaws.com"
            },
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket/EXAMPLE-LOGGING-
PREFIX*",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:::amzn-s3-demo-source-bucket"
                },
                "StringEquals": {
                    "aws:SourceAccount": "SOURCE-ACCOUNT-ID"
                }
            }
        }
    ]
}
```

Berikan izin ke grup pengiriman log dengan menggunakan bucket ACL

> **Note**
>
> Sebagai praktik keamanan terbaik, Amazon S3 menonaktifkan daftar kontrol akses (ACLs) secara default di semua bucket baru. Untuk informasi selengkapnya tentang izin ACL di konsol Amazon S3, lihat Mengkonfigurasi ACLs.

Meskipun kami tidak merekomendasikan pendekatan ini, Anda dapat memberikan izin ke grup pengiriman log dengan menggunakan bucket ACL. Namun, jika bucket tujuan menggunakan setelan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek, Anda tidak dapat menyetel bucket atau objek ACLs. Anda juga tidak dapat menyertakan pemberian tujuan (juga dikenal sebagai pemberian target) dalam konfigurasi PutBucketLogging Anda. Sebagai gantinya, Anda harus menggunakan kebijakan bucket untuk memberikan akses ke pengguna utama layanan logging (`logging.s3.amazonaws.com`). Untuk informasi selengkapnya, lihat Izin untuk pengiriman log.

Di bucket ACL, grup pengiriman log direpresentasikan dengan URL berikut:

```
http://acs.amazonaws.com/groups/s3/LogDelivery
```

Untuk memberikan izin `WRITE` dan `READ_ACP` (ACL baca), menambahkan pemberian berikut ini ke ACL bucket tujuan:

```
<Grant>
    <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:type="Group">
        <URI>http://acs.amazonaws.com/groups/s3/LogDelivery</URI>
    </Grantee>
    <Permission>WRITE</Permission>
</Grant>
<Grant>
    <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:type="Group">
        <URI>http://acs.amazonaws.com/groups/s3/LogDelivery</URI>
    </Grantee>
    <Permission>READ_ACP</Permission>
</Grant>
```

Untuk contoh penambahan pemberian ACL secara terprogram, lihat Mengkonfigurasi ACLs.

> **Important**
>
> Saat Anda mengaktifkan pencatatan akses server Amazon S3 dengan menggunakan AWS CloudFormation pada bucket dan Anda gunakan ACLs untuk memberikan akses ke grup pengiriman log S3, Anda juga harus menambahkan `"AccessControl": "LogDeliveryWrite"`ke template Anda. CloudFormation Melakukannya penting karena Anda dapat memberikan izin tersebut hanya dengan membuat ACL untuk bucket, tetapi Anda tidak dapat membuat kustom ACLs untuk bucket. CloudFormation Anda hanya dapat menggunakan kalengan ACLs dengan CloudFormation.

Untuk mengaktifkan pencatatan log akses server

Untuk mengaktifkan pencatatan akses server menggunakan konsol Amazon S3, Amazon S3 REST API AWS SDKs, AWS CLI dan, gunakan prosedur berikut.

Menggunakan konsol S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Dalam daftar bucket, pilih nama bucket yang ingin Anda aktifkan pencatatan akses server.
4. Pilih Properti.
5. Di Pencatatan akses server bagian, pilih Edit.
6. Di bawah Pencatatan akses server, pilih Aktifkan.
7. Di Bucket tujuan, tentukan bucket dan prefiks opsional. Jika Anda menentukan prefiks, sebaiknya sertakan garis miring ke depan (/) setelah prefiks agar lebih mudah untuk menemukan log Anda.

   > **Note**
   >
   > Menentukan prefiks dengan garis miring (/) membuatnya lebih mudah bagi Anda untuk menemukan objek log. Misalnya, jika Anda menentukan nilai prefiks`logs/`, setiap objek log yang dibuat Amazon S3 dimulai dengan prefiks `logs/` pada kuncinya, sebagai berikut:
   >
   > ```
   > logs/2013-11-01-21-32-16-E568B2907131C0C0
   > ```

Jika Anda menentukan nilai prefiks `logs`, objek log muncul sebagai berikut:

```
logs2013-11-01-21-32-16-E568B2907131C0C0
```

8. Di bawah Format kunci objek log , lakukan salah satu langkah berikut:

   - Untuk memilih non-date-based partisi, pilih [DestinationPrefix] [YYYY] - [MM] - [DD] - [hh] - [mm] - [ss] - []. UniqueString
   - Untuk memilih partisi berbasis tanggal, pilih [DestinationPrefix] []/[]/[SourceAccountId]/[SourceRegionYYYYSourceBucket]/[MM]/[DD]/[YYYY] - [MM] - [DD] - [hh] - [mm] - [ss] - [], lalu pilih waktu acara S3 atau Waktu pengiriman file log. UniqueString

9. Pilih Simpan perubahan.

   Saat Anda mengaktifkan pencatatan log akses server pada bucket, konsol keduanya memungkinkan pencatatan log pada bucket sumber dan memperbarui kebijakan bucket untuk bucket tujuan guna memberikan izin `s3:PutObject` kepada pengguna utama layanan pencatatan log (`logging.s3.amazonaws.com`). Untuk informasi lebih lanjut tentang kebijakan bucket ini, lihat Berikan izin kepada pengguna utama layanan logging dengan menggunakan kebijakan bucket.

   Anda dapat melihat pencatatan di dalam bucket tujuan. Setelah Anda mengaktifkan pencatatan akses server, mungkin perlu beberapa jam sebelum pencatatan dikirim ke bucket target. Untuk informasi selengkapnya tentang bagaimana dan kapan log dikirim, lihat Bagaimana log dikirimkan?

Untuk informasi selengkapnya, lihat Melihat properti untuk bucket tujuan umum S3.

Penggunaan API REST

Untuk mengaktifkan pencatatan log, Anda mengirimkan permintaan PutBucketLogging untuk menambahkan konfigurasi pencatatan log di bucket sumber. Permintaan menentukan bucket tujuan (juga dikenal sebagai bucket target) dan, secara opsional, prefiks yang akan digunakan dengan semua kunci objek log.

Contoh berikut mengidentifikasi *`amzn-s3-demo-destination-bucket`* sebagai bucket tujuan, dan *`logs/`*sebagai prefiks.

```
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">
```

```
  <LoggingEnabled>
    <TargetBucket>amzn-s3-demo-destination-bucket</TargetBucket>
    <TargetPrefix>logs/</TargetPrefix>
  </LoggingEnabled>
</BucketLoggingStatus>
```

Contoh berikut mengidentifikasi *amzn-s3-demo-destination-bucket* sebagai bucket tujuan, *logs/* sebagai prefiks, dan `EventTime` sebagai format kunci objek log.

```
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <LoggingEnabled>
    <TargetBucket>amzn-s3-demo-destination-bucket</TargetBucket>
    <TargetPrefix>logs/</TargetPrefix>
    <TargetObjectKeyFormat>
      <PartitionedPrefix>
         <PartitionDateSource>EventTime</PartitionDateSource>
      </PartitionedPrefix>
  </TargetObjectKeyFormat>
  </LoggingEnabled>
</BucketLoggingStatus>
```

Objek log ditulis dan dimiliki oleh akun pengiriman log S3, dan pemilik bucket diberi izin penuh pada objek log. Anda dapat secara opsional menggunakan pemberian tujuan (juga dikenal sebagai pemberian target) untuk memberikan izin kepada pengguna lain sehingga mereka dapat mengakses log. Untuk informasi selengkapnya, lihat PutBucketLogging.

> **Note**
>
> Jika bucket tujuan menggunakan pengaturan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek, Anda tidak dapat menggunakan pemberian tujuan untuk memberikan izin kepada pengguna lain. Untuk memberikan izin kepada orang lain, Anda dapat memperbarui kebijakan bucket di bucket tujuan. Untuk informasi selengkapnya, lihat Izin untuk pengiriman log.

Untuk mengambil konfigurasi logging pada bucket, gunakan operasi API GetBucketLogging.

Untuk menghapus konfigurasi pencatatan log, Anda mengirimkan `PutBucketLogging` permintaan dengan `BucketLoggingStatus` yang kosong:

```
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">
```

```
</BucketLoggingStatus>
```

Untuk mengaktifkan logging pada bucket, Anda dapat menggunakan Amazon S3 API atau pustaka pembungkus AWS SDK.

Menggunakan AWS SDKs

Contoh berikut mengaktifkan pencatatan log ke bucket. Anda harus membuat dua bucket, satu bucket sumber, dan satu bucket tujuan (target). Contoh ini memperbarui bucket ACL di bucket tujuan terlebih dahulu. Mereka kemudian memberikan ke grup pengiriman log izin yang diperlukan untuk menulis log ke bucket tujuan, dan kemudian mereka mengaktifkan pencatatan log ke bucket sumber.

Contoh ini tidak akan berfungsi pada bucket tujuan yang menggunakan pengaturan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek.

Jika bucket tujuan (target) menggunakan setelan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek, Anda tidak dapat menyetel bucket atau objek ACLs. Anda juga tidak dapat menyertakan hibah tujuan (target) dalam PutBucketLoggingkonfigurasi Anda. Anda harus menggunakan kebijakan bucket untuk memberikan akses ke pengguna utama layanan pencatatan (`logging.s3.amazonaws.com`). Untuk informasi selengkapnya, lihat Izin untuk pengiriman log.

.NET

SDK untuk .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
using System;
using System.IO;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;
using Microsoft.Extensions.Configuration;

/// <summary>
/// This example shows how to enable logging on an Amazon Simple Storage
/// Service (Amazon S3) bucket. You need to have two Amazon S3 buckets for
```

```
    /// this example. The first is the bucket for which you wish to enable
    /// logging, and the second is the location where you want to store the
    /// logs.
    /// </summary>
    public class ServerAccessLogging
    {
        private static IConfiguration _configuration = null!;

        public static async Task Main()
        {
            LoadConfig();

            string bucketName = _configuration["BucketName"];
            string logBucketName = _configuration["LogBucketName"];
            string logObjectKeyPrefix = _configuration["LogObjectKeyPrefix"];
            string accountId = _configuration["AccountId"];

            // If the AWS Region defined for your default user is different
            // from the Region where your Amazon S3 bucket is located,
            // pass the Region name to the Amazon S3 client object's constructor.
            // For example: RegionEndpoint.USWest2 or RegionEndpoint.USEast2.
            IAmazonS3 client = new AmazonS3Client();

            try
            {
                // Update bucket policy for target bucket to allow delivery of
logs to it.
                await SetBucketPolicyToAllowLogDelivery(
                    client,
                    bucketName,
                    logBucketName,
                    logObjectKeyPrefix,
                    accountId);

                // Enable logging on the source bucket.
                await EnableLoggingAsync(
                    client,
                    bucketName,
                    logBucketName,
                    logObjectKeyPrefix);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine($"Error: {e.Message}");
```

```
            }
        }

        /// <summary>
        /// This method grants appropriate permissions for logging to the
        /// Amazon S3 bucket where the logs will be stored.
        /// </summary>
        /// <param name="client">The initialized Amazon S3 client which will be
used
        /// to apply the bucket policy.</param>
        /// <param name="sourceBucketName">The name of the source bucket.</param>
        /// <param name="logBucketName">The name of the bucket where logging
        /// information will be stored.</param>
        /// <param name="logPrefix">The logging prefix where the logs should be
delivered.</param>
        /// <param name="accountId">The account id of the account where the
source bucket exists.</param>
        /// <returns>Async task.</returns>
        public static async Task SetBucketPolicyToAllowLogDelivery(
            IAmazonS3 client,
            string sourceBucketName,
            string logBucketName,
            string logPrefix,
            string accountId)
        {
            var resourceArn = @"""arn:aws:s3:::" + logBucketName + "/" +
logPrefix + @"*""";

            var newPolicy = @"{
                                ""Statement"":[{
                                ""Sid"": ""S3ServerAccessLogsPolicy"",
                                ""Effect"": ""Allow"",
                                ""Principal"": { ""Service"":
""logging.s3.amazonaws.com"" },
                                ""Action"": [""s3:PutObject""],
                                ""Resource"": [" + resourceArn + @"],
                                ""Condition"": {
                                ""ArnLike"": { ""aws:SourceArn"":
""arn:aws:s3:::" + sourceBucketName + @""" },
                                ""StringEquals"": { ""aws:SourceAccount"": """ +
accountId + @""" }
                                        }
                                    }]
                                }";
```

```
            Console.WriteLine($"The policy to apply to bucket {logBucketName} to
 enable logging:");
            Console.WriteLine(newPolicy);

            PutBucketPolicyRequest putRequest = new PutBucketPolicyRequest
            {
                BucketName = logBucketName,
                Policy = newPolicy,
            };
            await client.PutBucketPolicyAsync(putRequest);
            Console.WriteLine("Policy applied.");
        }

        /// <summary>
        /// This method enables logging for an Amazon S3 bucket. Logs will be
 stored
        /// in the bucket you selected for logging. Selected prefix
        /// will be prepended to each log object.
        /// </summary>
        /// <param name="client">The initialized Amazon S3 client which will be
 used
        /// to configure and apply logging to the selected Amazon S3 bucket.</
param>
        /// <param name="bucketName">The name of the Amazon S3 bucket for which
 you
        /// wish to enable logging.</param>
        /// <param name="logBucketName">The name of the Amazon S3 bucket where
 logging
        /// information will be stored.</param>
        /// <param name="logObjectKeyPrefix">The prefix to prepend to each
        /// object key.</param>
        /// <returns>Async task.</returns>
        public static async Task EnableLoggingAsync(
            IAmazonS3 client,
            string bucketName,
            string logBucketName,
            string logObjectKeyPrefix)
        {
            Console.WriteLine($"Enabling logging for bucket {bucketName}.");
            var loggingConfig = new S3BucketLoggingConfig
            {
                TargetBucketName = logBucketName,
                TargetPrefix = logObjectKeyPrefix,
            };
```

```
            var putBucketLoggingRequest = new PutBucketLoggingRequest
            {
                BucketName = bucketName,
                LoggingConfig = loggingConfig,
            };
            await client.PutBucketLoggingAsync(putBucketLoggingRequest);
            Console.WriteLine($"Logging enabled.");
        }

        /// <summary>
        /// Loads configuration from settings files.
        /// </summary>
        public static void LoadConfig()
        {
            _configuration = new ConfigurationBuilder()
                .SetBasePath(Directory.GetCurrentDirectory())
                .AddJsonFile("settings.json") // Load settings from .json file.
                .AddJsonFile("settings.local.json", true) // Optionally, load
  local settings.
                .Build();
        }
    }
```

- Untuk detail API, lihat PutBucketLoggingdi Referensi AWS SDK untuk .NET API.

Java

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.BucketLoggingStatus;
import software.amazon.awssdk.services.s3.model.LoggingEnabled;
import software.amazon.awssdk.services.s3.model.PartitionedPrefix;
import software.amazon.awssdk.services.s3.model.PutBucketLoggingRequest;
import software.amazon.awssdk.services.s3.model.TargetObjectKeyFormat;

// Class to set a bucket policy on a target S3 bucket and enable server access
 logging on a source S3 bucket.
public class ServerAccessLogging {
    private static S3Client s3Client;
```

```
    public static void main(String[] args) {
        String sourceBucketName = "SOURCE-BUCKET";
        String targetBucketName = "TARGET-BUCKET";
        String sourceAccountId = "123456789012";
        String targetPrefix = "logs/";

        // Create S3 Client.
        s3Client = S3Client.builder().
                region(Region.US_EAST_2)
                .build();

        // Set a bucket policy on the target S3 bucket to enable server access
 logging by granting the
        // logging.s3.amazonaws.com principal permission to use the PutObject
 operation.
        ServerAccessLogging serverAccessLogging = new ServerAccessLogging();
        serverAccessLogging.setTargetBucketPolicy(sourceAccountId, sourceBucketName,
 targetBucketName);

        // Enable server access logging on the source S3 bucket.
        serverAccessLogging.enableServerAccessLogging(sourceBucketName,
 targetBucketName,
                targetPrefix);

    }

    // Function to set a bucket policy on the target S3 bucket to enable server
 access logging by granting the
    // logging.s3.amazonaws.com principal permission to use the PutObject operation.
    public void setTargetBucketPolicy(String sourceAccountId, String
 sourceBucketName, String targetBucketName) {
        String policy = "{\n" +
                "    \"Version\": \"2012-10-17\",\n" +
                "    \"Statement\": [\n" +
                "        {\n" +
                "            \"Sid\": \"S3ServerAccessLogsPolicy\",\n" +
                "            \"Effect\": \"Allow\",\n" +
                "            \"Principal\": {\"Service\": \"logging.s3.amazonaws.com
\"},\n" +
                "            \"Action\": [\n" +
                "                \"s3:PutObject\"\n" +
                "            ],\n" +
                "            \"Resource\": \"arn:aws:s3:::" + targetBucketName + "/*
\",\n" +
```

```
                "            \"Condition\": {\n" +
                "                \"ArnLike\": {\n" +
                "                    \"aws:SourceArn\": \"arn:aws:s3:::" +
 sourceBucketName + "\"\n" +
                "                },\n" +
                "                \"StringEquals\": {\n" +
                "                    \"aws:SourceAccount\": \"" + sourceAccountId +
 "\"\n" +
                "                }\n" +
                "            }\n" +
                "        }\n" +
                "    ]\n" +
                "}";
        s3Client.putBucketPolicy(b -> b.bucket(targetBucketName).policy(policy));
    }

    // Function to enable server access logging on the source S3 bucket.
    public void enableServerAccessLogging(String sourceBucketName, String
 targetBucketName,
            String targetPrefix) {
        TargetObjectKeyFormat targetObjectKeyFormat =
 TargetObjectKeyFormat.builder()

 .partitionedPrefix(PartitionedPrefix.builder().partitionDateSource("EventTime").build())
                .build();
        LoggingEnabled loggingEnabled = LoggingEnabled.builder()
                .targetBucket(targetBucketName)
                .targetPrefix(targetPrefix)
                .targetObjectKeyFormat(targetObjectKeyFormat)
                .build();
        BucketLoggingStatus bucketLoggingStatus = BucketLoggingStatus.builder()
                .loggingEnabled(loggingEnabled)
                .build();
        s3Client.putBucketLogging(PutBucketLoggingRequest.builder()
                .bucket(sourceBucketName)
                .bucketLoggingStatus(bucketLoggingStatus)
                .build());
    }

}
```

Menggunakan AWS CLI

Kami menyarankan Anda membuat bucket logging khusus di setiap bucket S3 Wilayah AWS yang Anda miliki. Kemudian, mintalah log akses Amazon S3 dikirimkan ke bucket S3 tersebut. Untuk informasi dan contoh selengkapnya, lihat put-bucket-logging di Referensi AWS CLI .

Jika bucket tujuan (target) menggunakan setelan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek, Anda tidak dapat menyetel bucket atau objek ACLs. Anda juga tidak dapat menyertakan hibah tujuan (target) dalam PutBucketLoggingkonfigurasi Anda. Anda harus menggunakan kebijakan bucket untuk memberikan akses ke pengguna utama layanan pencatatan (`logging.s3.amazonaws.com`). Untuk informasi selengkapnya, lihat Izin untuk pengiriman log.

Example — Mengaktifkan pencatatan akses dengan lima bucket di dua Wilayah

Dalam contoh ini, Anda memiliki lima bucket berikut:

- *amzn-s3-demo-source-bucket*-us-east-1
- *amzn-s3-demo-source-bucket1*-us-east-1
- *amzn-s3-demo-source-bucket2*-us-east-1
- *amzn-s3-demo-bucket1*-us-west-2
- *amzn-s3-demo-bucket2*-us-west-2

> **Note**
>
> Langkah terakhir dari prosedur berikut memberikan contoh skrip bash yang dapat Anda gunakan untuk membuat bucket logging Anda dan mengaktifkan log akses server pada bucket ini. Untuk menggunakan skrip tersebut, Anda harus membuat file `policy.json` dan `logging.json`, seperti yang dijelaskan dalam prosedur berikut.

1. Buat dua bucket pencatatan di Wilayah AS Barat (Oregon) dan AS Timur (Virginia Utara), dan beri mereka nama-nama berikut ini:
   - *amzn-s3-demo-destination-bucket*-logs-us-east-1
   - *amzn-s3-demo-destination-bucket1*-logs-us-west-2
2. Kemudian dalam langkah-langkah ini, Anda akan mengaktifkan pencatatan akses server sebagai berikut:

- *amzn-s3-demo-source-bucket*-us-east-1 menyimpan log ke bucket S3 *amzn-s3-demo-destination-bucket*-logs-us-east-1 dengan prefiks *amzn-s3-demo-source-bucket*-us-east-1
- *amzn-s3-demo-source-bucket1*-us-east-1 menyimpan log ke bucket S3 *amzn-s3-demo-destination-bucket*-logs-us-east-1 dengan prefiks *amzn-s3-demo-source-bucket1*-us-east-1
- *amzn-s3-demo-source-bucket2*-us-east-1 menyimpan log ke bucket S3 *amzn-s3-demo-destination-bucket*-logs-us-east-1 dengan prefiks *amzn-s3-demo-source-bucket2*-us-east-1
- *amzn-s3-demo-bucket1*-us-west-2 menyimpan log ke bucket S3 *amzn-s3-demo-destination-bucket1*-logs-us-west-2 dengan prefiks *amzn-s3-demo-bucket1*-us-west-2
- *amzn-s3-demo-bucket2*-us-west-2 menyimpan log ke bucket S3 *amzn-s3-demo-destination-bucket1*-logs-us-west-2 dengan prefiks *amzn-s3-demo-bucket2*-us-west-2

3. Untuk setiap bucket logging tujuan, berikan izin untuk pengiriman log akses server dengan menggunakan bucket ACL atau kebijakan bucket:

   - Perbarui kebijakan bucket (Disarankan)–Untuk memberikan izin kepada pengguna utama layanan logging, gunakan perintah `put-bucket-policy` berikut. Ganti *amzn-s3-demo-destination-bucket*-logs dengan nama bucket tujuan Anda.

     ```
     aws s3api put-bucket-policy --bucket amzn-s3-demo-destination-bucket-logs --
     policy file://policy.json
     ```

     `Policy.json` adalah dokumen JSON di dalam folder saat ini, yang berisi kebijakan bucket berikut. Untuk menggunakan kebijakan bucket ini, ganti *user input placeholders* dengan informasi Anda sendiri. Dalam kebijakan berikut, *amzn-s3-demo-destination-bucket-logs*adalah bucket tujuan tempat log akses server akan dikirimkan, dan *amzn-s3-demo-source-bucket* merupakan bucket sumber. *SOURCE-ACCOUNT-ID* adalah Akun AWS yang memiliki bucket sumber.

     ```
     {
         "Version": "2012-10-17",
         "Statement": [
     ```

```
        {
            "Sid": "S3ServerAccessLogsPolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "logging.s3.amazonaws.com"
            },
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket-logs/*",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:::amzn-s3-demo-source-bucket"
                },
                "StringEquals": {
                    "aws:SourceAccount": "SOURCE-ACCOUNT-ID"
                }
            }
        }
    ]
}
```

- Perbarui bucket ACL—Untuk memberikan izin ke grup pengiriman log S3, gunakan perintah `put-bucket-acl` berikut. Ganti *amzn-s3-demo-destination-bucket-logs* dengan nama bucket tujuan (target) Anda.

```
aws s3api put-bucket-acl --bucket amzn-s3-demo-destination-bucket-logs  --
grant-write URI=http://acs.amazonaws.com/groups/s3/LogDelivery --grant-read-acp
 URI=http://acs.amazonaws.com/groups/s3/LogDelivery
```

4. Kemudian, buat file `logging.json` yang berisi konfigurasi logging Anda (berdasarkan salah satu dari tiga contoh berikut). Setelah Anda membuat file `logging.json`, Anda dapat menerapkan konfigurasi logging dengan menggunakan perintah `put-bucket-logging` berikut. Ganti *amzn-s3-demo-destination-bucket-logs* dengan nama bucket tujuan (target) Anda.

```
aws s3api put-bucket-logging --bucket amzn-s3-demo-destination-bucket-logs --
bucket-logging-status file://logging.json
```

> **Note**
>
> Alih-alih menggunakan perintah `put-bucket-logging` ini untuk menerapkan konfigurasi logging pada setiap bucket tujuan, Anda dapat menggunakan salah satu skrip bash yang disediakan di langkah berikutnya. Untuk menggunakan skrip tersebut, Anda harus membuat file `policy.json` dan `logging.json`, seperti yang dijelaskan dalam prosedur ini.

File `logging.json` ini adalah dokumen JSON di folder saat ini yang berisi konfigurasi pencatatan log. Jika bucket tujuan menggunakan pengaturan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek, konfigurasi logging Anda tidak dapat berisi pemberian tujuan (target). Untuk informasi selengkapnya, lihat Izin untuk pengiriman log.

Example – **`logging.json`** tanpa pemberian tujuan (target)

File contoh `logging.json` berikut tidak berisi pemberian tujuan (target). Oleh karena itu, Anda dapat menerapkan konfigurasi ini ke bucket tujuan (target) yang menggunakan pengaturan yang diterapkan pemilik Bucket untuk Kepemilikan Objek.

```
{
    "LoggingEnabled": {
        "TargetBucket": "amzn-s3-demo-destination-bucket-logs",
        "TargetPrefix": "amzn-s3-demo-destination-bucket/"
     }
 }
```

Example – **`logging.json`** dengan pemberian tujuan (target)

File contoh `logging.json` berikut berisi pemberian tujuan (target).

Jika bucket tujuan menggunakan pengaturan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek, Anda tidak dapat menyertakan pemberian tujuan (target) dalam konfigurasi PutBucketLogging Anda. Untuk informasi selengkapnya, lihat Izin untuk pengiriman log.

```
{
    "LoggingEnabled": {
        "TargetBucket": "amzn-s3-demo-destination-bucket-logs",
        "TargetPrefix": "amzn-s3-demo-destination-bucket/",
        "TargetGrants": [
             {
                "Grantee": {
                    "Type": "AmazonCustomerByEmail",
                    "EmailAddress": "user@example.com"
                 },
                "Permission": "FULL_CONTROL"
             }
         ]
     }
 }
```

Nilai penerima hibah

Anda dapat menentukan orang (penerima hibah) kepada siapa Anda menetapkan hak akses (dengan menggunakan elemen permintaan) dengan cara berikut:

- Dengan ID orang tersebut:

```
{
  "Grantee": {
    "Type": "CanonicalUser",
    "ID": "ID",
    "DisplayName": "GranteesEmail"
  }
}
```

  `DisplayName`bersifat opsional dan diabaikan dalam permintaan.

- Melalui alamat email:

```
{
  "Grantee": {
    "Type": "AmazonCustomerByEmail",
    "EmailAddress": "username@example.com"
```

```
  }
}
```

Penerima hibah diselesaikan untuk `CanonicalUser` dan, sebagai tanggapan `GetObjectAcl` atas permintaan, muncul sebagai. `CanonicalUser`

> **Note**
>
> Menggunakan alamat email untuk menentukan penerima hibah hanya didukung di beberapa. Wilayah AWS Untuk informasi selengkapnya, lihat Grantee di Referensi API Amazon S3.

- Oleh URI:

```
{
  "Grantee": {
    "Type": "Group",
    "URI": "http://acs.amazonaws.com/groups/global/AuthenticatedUsers"
  }
}
```

Example - **`logging.json`** dengan format kunci objek log yang diatur ke waktu peristiwa S3

File `logging.json` berikut mengubah format kunci objek log ke waktu peristiwa S3. Untuk informasi selengkapnya tentang pengaturan format kunci objek log, lihat the section called "Bagaimana cara mengaktifkan log pengiriman?"

```
  {
    "LoggingEnabled": {
        "TargetBucket": "amzn-s3-demo-destination-bucket-logs",
        "TargetPrefix": "amzn-s3-demo-destination-bucket/",
        "TargetObjectKeyFormat": {
            "PartitionedPrefix": {
                "PartitionDateSource": "EventTime"
            }
         }
    }
}
```

5. Gunakan salah satu skrip bash berikut untuk menambahkan pencatatan log akses untuk semua bucket dalam akun Anda. Ganti *amzn-s3-demo-destination-bucket-logs* dengan nama bucket tujuan (target) Anda, dan ganti *us-west-2* dengan nama Wilayah tempat bucket Anda berada.

> **Note**
>
> Script ini hanya berfungsi jika semua bucket Anda berada di Wilayah yang sama. Jika Anda memiliki bucket di beberapa Wilayah, Anda harus menyesuaikan skrip-nya.

Example – Memberikan akses dengan kebijakan bucket dan menambahkan logging untuk bucket di akun Anda

```
loggingBucket='amzn-s3-demo-destination-bucket-logs'
region='us-west-2'


# Create the logging bucket.
aws s3 mb s3://$loggingBucket --region $region

aws s3api put-bucket-policy --bucket $loggingBucket --policy file://policy.json

# List the buckets in this account.
buckets="$(aws s3 ls | awk '{print $3}')"

# Put a bucket logging configuration on each bucket.
for bucket in $buckets
    do
      # This if statement excludes the logging bucket.
      if [ "$bucket" == "$loggingBucket" ] ; then
          continue;
      fi
      printf '{
        "LoggingEnabled": {
          "TargetBucket": "%s",
          "TargetPrefix": "%s/"
      }
    }' "$loggingBucket" "$bucket"  > logging.json
```

```
      aws s3api put-bucket-logging --bucket $bucket --bucket-logging-status file://
logging.json
      echo "$bucket done"
  done

  rm logging.json

  echo "Complete"
```

Example — Berikan akses dengan bucket ACLs dan tambahkan logging untuk bucket di akun Anda

```
  loggingBucket='amzn-s3-demo-destination-bucket-logs'
  region='us-west-2'


  # Create the logging bucket.
  aws s3 mb s3://$loggingBucket --region $region

  aws s3api put-bucket-acl --bucket $loggingBucket --grant-write URI=http://
acs.amazonaws.com/groups/s3/LogDelivery --grant-read-acp URI=http://
acs.amazonaws.com/groups/s3/LogDelivery

  # List the buckets in this account.
  buckets="$(aws s3 ls | awk '{print $3}')"

  # Put a bucket logging configuration on each bucket.
  for bucket in $buckets
      do
        # This if statement excludes the logging bucket.
        if [ "$bucket" == "$loggingBucket" ] ; then
            continue;
        fi
        printf '{
          "LoggingEnabled": {
            "TargetBucket": "%s",
            "TargetPrefix": "%s/"
        }
      }' "$loggingBucket" "$bucket"  > logging.json
      aws s3api put-bucket-logging --bucket $bucket --bucket-logging-status file://
logging.json
      echo "$bucket done"
```

```
done

rm logging.json

echo "Complete"
```

Memverifikasi penyiapan log akses server Anda

Setelah Anda mengaktifkan pencatatan akses server, lakukan langkah berikut:

- Akses bucket tujuan dan verifikasi bahwa file log sedang dikirim. Setelah log akses disiapkan, Amazon S3 segera mulai menangkap permintaan dan mencatatnya. Namun, mungkin perlu beberapa jam sebelum log dikirim ke ember tujuan. Untuk informasi selengkapnya, lihat the section called “Perubahan status pencatatan log bucket memerlukan waktu” dan the section called “Pengiriman log server dengan upaya terbaik”.

  Anda juga dapat memverifikasi pengiriman log secara otomatis dengan menggunakan metrik permintaan Amazon S3 dan menyiapkan CloudWatch alarm Amazon untuk metrik ini. Untuk informasi selengkapnya, lihat Memantau metrik dengan Amazon CloudWatch.
- Verifikasi bahwa Anda dapat membuka dan membaca isi file log.

Untuk informasi pemecahan masalah pencatatan akses server, lihat Memecahkan masalah pencatatan akses server.

Server Amazon S3 mengakses format log

Pencatatan akses server menyediakan catatan terperinci untuk permintaan yang dilakukan ke bucket Amazon S3 Anda. Anda dapat menggunakan log akses server untuk tujuan berikut:

- Melakukan audit keamanan dan akses
- Belajar tentang basis pelanggan Anda
- Memahami tagihan Amazon S3 Anda

Bagian ini menjelaskan format dan detail lain tentang file log akses server Amazon S3.

Berkas log akses server terdiri atas urutan catatan log yang dibatasi di baris baru. Setiap catatan log mewakili satu permintaan dan terdiri dari bidang dengan ruang terbatas.

Berikut ini adalah contoh log yang terdiri dari lima catatan log.

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 amzn-s3-demo-bucket1 [06/Feb/2019:00:00:38 +0000] 192.0.2.3
 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 3E57427F3EXAMPLE
 REST.GET.VERSIONING - "GET /amzn-s3-demo-bucket1?versioning HTTP/1.1" 200 - 113 - 7 -
 "-" "S3Console/0.4" - s9lzHYrFp76ZVxRcpX9+5cjAnEH2ROuNkd2BHfIa6UkFVdtjf5mKR3/eTPFvsiP/
XV/VLi31234= SigV4 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader amzn-s3-demo-bucket1.s3.us-
west-1.amazonaws.com TLSV1.2 arn:aws:s3:us-west-1:123456789012:accesspoint/example-AP
 Yes
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 amzn-s3-demo-bucket1 [06/Feb/2019:00:00:38 +0000] 192.0.2.3
 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 891CE47D2EXAMPLE
 REST.GET.LOGGING_STATUS - "GET /amzn-s3-demo-bucket1?logging HTTP/1.1" 200 -
 242 - 11 - "-" "S3Console/0.4" - 9vKBE6vMhrNiWHZmb2L0mXOcqPGzQOI5XLnCtZNPxev+Hf
+7tpT6sxDwDty4LHBUOZJG96N1234= SigV4 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader amzn-s3-
demo-bucket1.s3.us-west-1.amazonaws.com TLSV1.2 - -
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 amzn-s3-demo-bucket1 [06/Feb/2019:00:00:38 +0000] 192.0.2.3
 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be A1206F460EXAMPLE
 REST.GET.BUCKETPOLICY - "GET /amzn-s3-demo-bucket1?policy HTTP/1.1" 404
 NoSuchBucketPolicy 297 - 38 - "-" "S3Console/0.4" - BNaBsXZQQDbssi6xMBdBU2sLt
+Yf5kZDmeBUP35sFoKa3sLLeMC78iwEIWxs99CRUrbS4n11234= SigV4 ECDHE-RSA-AES128-GCM-SHA256
 AuthHeader amzn-s3-demo-bucket1.s3.us-west-1.amazonaws.com TLSV1.2 - Yes
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 amzn-s3-demo-bucket1 [06/Feb/2019:00:01:00 +0000] 192.0.2.3
 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 7B4A0FABBEXAMPLE
 REST.GET.VERSIONING - "GET /amzn-s3-demo-bucket1?versioning HTTP/1.1" 200 -
 113 - 33 - "-" "S3Console/0.4" - Ke1bUcazaN1jWuUlPJaxF64cQVpUEhoZKEG/hmy/gijN/
I1DeWqDfFvnpybfEseEME/u7ME1234= SigV4 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader amzn-s3-
demo-bucket1.s3.us-west-1.amazonaws.com TLSV1.2 - -
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 amzn-s3-demo-bucket1 [06/Feb/2019:00:01:57 +0000] 192.0.2.3
 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 DD6CC733AEXAMPLE REST.PUT.OBJECT s3-dg.pdf "PUT /amzn-s3-demo-bucket1/
s3-dg.pdf HTTP/1.1" 200 - - 4406583 41754 28 "-" "S3Console/0.4" -
 10S62Zv81kBW7BB6SX4XJ48o6kpcl6LPwEoizZQQxJd5qDSCTLX0TgS37kYUBKQW3+bPdrg1234= SigV4
 ECDHE-RSA-AES128-SHA AuthHeader amzn-s3-demo-bucket1.s3.us-west-1.amazonaws.com
 TLSV1.2 - Yes
```

> **Note**
>
> Bidang apa pun dapat diatur menjadi - untuk menunjukkan bahwa data tidak diketahui atau tidak tersedia, atau bahwa bidang tidak berlaku untuk permintaan ini.

Topik

- Bidang catatan log
- Pencatatan Tambahan untuk operasi penyalinan
- Informasi log akses kustom
- Pertimbangan pemrograman untuk format log akses server yang dapat diperluas

Bidang catatan log

Daftar berikut menjelaskan bidang catatan log.

Pemilik Bucket

ID pengguna resmi dari pemilik bucket sumber. ID pengguna kanonik adalah bentuk lain dari ID. Akun AWS Untuk informasi selengkapnya tentang ID pengguna kanonik, lihat Akun AWS pengidentifikasi di Referensi Umum AWS. Untuk informasi tentang cara menemukan ID pengguna kanonik untuk akun Anda, lihat Menemukan ID Pengguna kanonik Akun AWS Anda.

Entri contoh

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Bucket

Nama bucket tempat permintaan diproses untuk dibandingkan. Jika sistem menerima permintaan yang salah bentuknya dan tidak dapat menentukan bucket, permintaan tidak akan muncul di log akses server apa pun.

Entri contoh

```
amzn-s3-demo-bucket1
```

Waktu

Waktu pada saat permintaan diterima; tanggal dan waktu ini berada dalam Waktu Universal Terkoordinasi (UTC). Formatnya, menggunakan terminologi `strftime()`, yaitu sebagai berikut: `[%d/%b/%Y:%H:%M:%S %z]`

Entri contoh

```
[06/Feb/2019:00:00:38 +0000]
```

IP Jarak Jauh

Alamat IP yang jelas dari pemohon. Proxy perantara dan firewall mungkin mengaburkan alamat IP aktual perangkat yang membuat permintaan.

Entri contoh

```
192.0.2.3
```

Peminta

ID pengguna kanonik pemohon, atau - untuk permintaan yang tidak terautentikasi. Jika pemohon adalah pengguna IAM, bidang ini mengembalikan nama pengguna IAM pemohon bersama dengan Akun AWS yang dimiliki pengguna IAM. Pengidentifikasi ini sama dengan yang digunakan untuk tujuan kontrol akses.

Entri contoh

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Jika pemohon menggunakan peran yang diasumsikan, bidang ini mengembalikan peran IAM yang diasumsikan.

Entri contoh

```
arn:aws:sts::123456789012:assumed-role/roleName/test-role
```

ID Permintaan

String yang dibuat oleh Amazon S3 untuk mengidentifikasi setiap permintaan secara unik.

Entri contoh

```
3E57427F33A59F07
```

Operasi

Operasi yang tercantum di sini dinyatakan sebagai `SOAP.operation`, `REST.HTTP_method.resource_type`, `WEBSITE.HTTP_method.resource_type`, atau `BATCH.DELETE.OBJECT`, atau `S3.action.resource_type` untuk Siklus hidup S3 dan pencatatan.

Entri contoh

```
REST.PUT.OBJECT
```

Kunci

Kunci (nama objek) bagian dari permintaan.

Entri contoh

```
/photos/2019/08/puppy.jpg
```

Request-URI

Bagian `Request-URI` dari pesan permintaan HTTP.

Entri Contoh

```
"GET /amzn-s3-demo-bucket1/photos/2019/08/puppy.jpg?x-foo=bar HTTP/1.1"
```

Status HTTP

Kode status HTTP numerik dari respons.

Entri contoh

```
200
```

Kode Kesalahan

Respons Kesalahan Amazon S3, atau - jika tidak terjadi kesalahan.

Entri contoh

```
NoSuchBucket
```

Byte Dikirim

Jumlah byte respons yang dikirim, tidak termasuk overhead protokol HTTP, atau - jika nol.

Entri contoh

```
2662992
```

Ukuran Objek

Ukuran total objek yang dimaksud.

Entri contoh

```
3462992
```

Total Waktu

Jumlah milidetik permintaan dalam proses pengiriman dari perspektif server. Nilai ini diukur sejak permintaan Anda diterima hingga byte terakhir respons dikirimkan. Pengukuran yang dibuat dari perspektif klien mungkin lebih lama oleh sebab latensi jaringan.

Entri contoh

```
70
```

Waktu Perputaran

Jumlah milidetik yang dihabiskan Amazon S3 untuk memproses permintaan Anda. Nilai ini diukur sejak byte terakhir permintaan Anda diterima hingga byte pertama respons dikirim.

Entri contoh

```
10
```

Referer

Nilai dari header HTTP `Referer`, jika ada. Agen pengguna HTTP (misalnya, browser) biasanya mengatur header ini ke URL dari halaman penautan atau penyematan saat membuat permintaan.

Entri contoh

```
"http://www.example.com/webservices"
```

User-Agent

Nilai dari header HTTP `User-Agent`.

Entri contoh

```
"curl/7.15.1"
```

ID Versi

ID versi dalam permintaan, atau `-` jika operasi tidak mengambil `versionId` parameter.

Entri contoh

```
3HL4kqtJvjVBH40Nrjfkd
```

Id Host

ID permintaan `x-amz-id-2` yang diperluas atau Amazon S3.

Entri contoh

```
s9lzHYrFp76ZVxRcpX9+5cjAnEH2ROuNkd2BHfIa6UkFVdtjf5mKR3/eTPFvsiP/XV/VLi31234=
```

Versi Tanda Tangan

Versi tanda tangan, `SigV2` atau `SigV4`, yang digunakan untuk mengautentikasi permintaan atau `-` untuk permintaan tidak terautentikasi.

Entri contoh

```
SigV2
```

Rangkaian Penyandian

Cipher Transport Layer Security (TLS) yang dinegosiasikan untuk permintaan HTTPS atau untuk HTTP. `-`

Entri contoh

```
ECDHE-RSA-AES128-GCM-SHA256
```

Jenis Autentikasi

Jenis autentikasi permintaan yang digunakan, `AuthHeader` untuk header autentikasi, `QueryString` untuk string kueri (URL yang ditandatangani sebelumnya) atau `-` untuk permintaan tidak terautentikasi.

Entri contoh

```
AuthHeader
```

Header Host

Titik akhir yang digunakan untuk terhubung ke Amazon S3.

Entri contoh

```
s3.us-west-2.amazonaws.com
```

Beberapa Wilayah sebelumnya mendukung titik akhir lama. Anda mungkin melihat titik akhir ini di log atau AWS CloudTrail log akses server Anda. Untuk informasi selengkapnya, lihat Titik akhir legasi. Untuk daftar lengkap titik akhir dan Wilayah Amazon S3, lihat Titik akhir dan kuota Amazon S3 di Referensi Umum Amazon Web Services.

Versi TLS

Versi Keamanan Lapisan Pengangkutan (TLS) yang dinegosiasikan oleh klien. Nilai tersebut adalah salah satu dari berikut ini:`TLSv1.1`, `TLSv1.2`, `TLSv1.3`; atau `-` jika TLS tidak digunakan.

Entri contoh

```
TLSv1.2
```

Titik Akses ARN

Amazon Resource Name (ARN) dari titik akses permintaan. Jika titik akses ARN mengalami kecacatan atau tidak digunakan, bidang akan berisi `-`. Untuk informasi lebih lanjut tentang titik akses, lihat Menggunakan jalur akses Amazon S3 untuk bucket tujuan umum. Untuk informasi selengkapnya ARNs, lihat Amazon Resource Name (ARN) di Panduan AWS Referensi.

Entri contoh

```
arn:aws:s3:us-east-1:123456789012:accesspoint/example-AP
```

aclRequired

String yang menunjukkan apakah permintaan memerlukan daftar kontrol akses (ACL) untuk otorisasi. Jika permintaan memerlukan ACL untuk otorisasi, stringnya adalah `Yes`. Jika ACLs tidak diperlukan, string adalah`-`. Untuk informasi lebih lanjut tentang ACLs, lihatGambaran umum daftar kontrol akses (ACL). Untuk informasi selengkapnya tentang menggunakan `aclRequired` bidang untuk menonaktifkan ACLs, lihatMengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.

Entri contoh

```
Yes
```

Pencatatan Tambahan untuk operasi penyalinan

Sebuah operasi penyalinan melibatkan `GET` dan sebuah `PUT`. Karena alasan tersebut, kami mencatat dua catatan saat melakukan operasi penyalinan. Bagian sebelumnya menguraikan bidang yang terkait dengan bagian `PUT` dari operasi. Daftar berikut menjelaskan kolom dalam catatan yang berhubungan dengan bagian `GET` dari operasi penyalinan.

Pemilik Bucket

ID pengguna resmi dari bucket yang menyimpan objek yang disalin. ID pengguna kanonik adalah bentuk lain dari ID. Akun AWS Untuk informasi selengkapnya tentang ID pengguna kanonik, lihat Akun AWS pengidentifikasi di Referensi Umum AWS. Untuk informasi tentang cara menemukan ID pengguna kanonik untuk akun Anda, lihat Menemukan ID Pengguna kanonik Akun AWS Anda.

Entri contoh

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Bucket

Nama bucket yang menyimpan objek yang disalin.

Entri contoh

```
amzn-s3-demo-bucket1
```

Waktu

Waktu pada saat permintaan diterima; tanggal dan waktu ini berada dalam Waktu Universal Terkoordinasi (UTC). Formatnya, menggunakan terminologi `strftime()`, yaitu sebagai berikut: `[%d/%B/%Y:%H:%M:%S %z]`

Entri contoh

```
[06/Feb/2019:00:00:38 +0000]
```

IP Jarak Jauh

Alamat IP yang jelas dari pemohon. Proxy perantara dan firewall mungkin mengaburkan alamat IP aktual perangkat yang membuat permintaan.

Entri contoh

```
192.0.2.3
```

Peminta

ID pengguna kanonik pemohon, atau - untuk permintaan yang tidak terautentikasi. Jika pemohon adalah pengguna IAM, bidang ini akan mengembalikan nama pengguna IAM pemohon bersama dengan Pengguna root akun AWS yang dimiliki pengguna IAM. Pengidentifikasi ini sama dengan yang digunakan untuk tujuan kontrol akses.

Entri contoh

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Jika pemohon menggunakan peran yang diasumsikan, bidang ini mengembalikan peran IAM yang diasumsikan.

Entri contoh

```
arn:aws:sts::123456789012:assumed-role/roleName/test-role
```

ID Permintaan

String yang dibuat oleh Amazon S3 untuk mengidentifikasi setiap permintaan secara unik.

Entri contoh

```
3E57427F33A59F07
```

Operasi

Operasi yang tercantum di sini dinyatakan sebagai `SOAP.operation`, `REST.HTTP_method.resource_type`, `WEBSITE.HTTP_method.resource_type`, atau `BATCH.DELETE.OBJECT`.

Entri contoh

```
REST.COPY.OBJECT_GET
```

Kunci

Kunci (nama objek) dari objek yang disalin, atau `-` jika operasi tidak mengambil parameter kunci.

Entri contoh

```
/photos/2019/08/puppy.jpg
```

Request-URI

Bagian `Request-URI` dari pesan permintaan HTTP.

Entri contoh

```
"GET /amzn-s3-demo-bucket1/photos/2019/08/puppy.jpg?x-foo=bar"
```

Status HTTP

Kode status HTTP numerik dari bagian `GET` dari operasi penyalinan.

Entri contoh

```
200
```

Kode Kesalahan

Respons Kesalahan Amazon S3 dari GET bagian operasi penyalinan, atau - jika tidak terjadi kesalahan.

Entri contoh

```
NoSuchBucket
```

Byte Dikirim

Jumlah byte respons yang dikirim, tidak termasuk overhead protokol HTTP, atau - jika nol.

Entri contoh

```
2662992
```

Ukuran Objek

Ukuran total objek yang dimaksud.

Entri contoh

```
3462992
```

Total Waktu

Jumlah milidetik permintaan dalam proses pengiriman dari perspektif server. Nilai ini diukur sejak permintaan Anda diterima hingga byte terakhir respons dikirimkan. Pengukuran yang dibuat dari perspektif klien mungkin lebih lama oleh sebab latensi jaringan.

Entri contoh

```
70
```

Waktu Perputaran

Jumlah milidetik yang dihabiskan Amazon S3 untuk memproses permintaan Anda. Nilai ini diukur sejak byte terakhir permintaan Anda diterima hingga byte pertama respons dikirim.

Entri contoh

```
10
```

Referer

Nilai dari header HTTP `Referer`, jika ada. Agen pengguna HTTP (misalnya, browser) biasanya mengatur header ini ke URL dari halaman penautan atau penyematan saat membuat permintaan.

Entri contoh

```
"http://www.example.com/webservices"
```

User-Agent

Nilai dari header HTTP `User-Agent`.

Entri contoh

```
"curl/7.15.1"
```

ID Versi

ID versi objek yang disalin, atau `-` jika header `x-amz-copy-source` tidak menentukan parameter `versionId` sebagai bagian dari sumber salinan.

Entri Contoh

```
3HL4kqtJvjVBH40Nrjfkd
```

Id Host

ID permintaan `x-amz-id-2` yang diperluas atau Amazon S3.

Entri contoh

```
s9lzHYrFp76ZVxRcpX9+5cjAnEH2ROuNkd2BHfIa6UkFVdtjf5mKR3/eTPFvsiP/XV/VLi31234=
```

Versi Tanda Tangan

Versi tanda tangan, `SigV2` atau `SigV4`, yang digunakan untuk mengautentikasi permintaan atau `-` untuk permintaan tidak terautentikasi.

Entri contoh

```
SigV4
```

Rangkaian Penyandian

Cipher Transport Layer Security (TLS) yang dinegosiasikan untuk permintaan HTTPS, atau untuk HTTP. -

Entri contoh

```
ECDHE-RSA-AES128-GCM-SHA256
```

Jenis Autentikasi

Jenis otentikasi permintaan yang digunakan: `AuthHeader` untuk header otentikasi, untuk string kueri (presigned URLs), atau `QueryString` untuk permintaan yang tidak diautentikasi. -

Entri contoh

```
AuthHeader
```

Header Host

Titik akhir yang digunakan untuk terhubung ke Amazon S3.

Entri contoh

```
s3.us-west-2.amazonaws.com
```

Beberapa Wilayah sebelumnya mendukung titik akhir lama. Anda mungkin melihat titik akhir ini di log atau AWS CloudTrail log akses server Anda. Untuk informasi selengkapnya, lihat Titik akhir legasi. Untuk daftar lengkap titik akhir dan Wilayah Amazon S3, lihat Titik akhir dan kuota Amazon S3 di Referensi Umum Amazon Web Services.

Versi TLS

Versi Keamanan Lapisan Pengangkutan (TLS) yang dinegosiasikan oleh klien. Nilai tersebut adalah salah satu dari berikut ini:`TLSv1.1`, `TLSv1.2`, `TLSv1.3`; atau - jika TLS tidak digunakan.

Entri contoh

```
TLSv1.2
```

Titik Akses ARN

Amazon Resource Name (ARN) dari titik akses permintaan. Jika titik akses ARN mengalami kecacatan atau tidak digunakan, bidang akan berisi `-`. Untuk informasi lebih lanjut tentang titik akses, lihat Menggunakan jalur akses Amazon S3 untuk bucket tujuan umum. Untuk informasi selengkapnya ARNs, lihat Amazon Resource Name (ARN) di Panduan AWS Referensi.

Entri contoh

```
arn:aws:s3:us-east-1:123456789012:accesspoint/example-AP
```

aclRequired

String yang menunjukkan apakah permintaan memerlukan daftar kontrol akses (ACL) untuk otorisasi. Jika permintaan memerlukan ACL untuk otorisasi, stringnya adalah `Yes`. Jika ACLs tidak diperlukan, string adalah`-`. Untuk informasi lebih lanjut tentang ACLs, lihatGambaran umum daftar kontrol akses (ACL). Untuk informasi selengkapnya tentang menggunakan `aclRequired` bidang untuk menonaktifkan ACLs, lihatMengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.

Entri contoh

```
Yes
```

Informasi log akses kustom

Anda dapat menyertakan informasi kustom untuk disimpan dalam catatan log akses untuk permintaan. Untuk melakukannya, menambahkan parameter string kueri kustom ke URL untuk permintaan tersebut. Amazon S3 mengabaikan parameter query-string yang dimulai dengan`x-`, tetapi mencakup parameter tersebut dalam catatan akses untuk permintaan, sebagai bagian dari `Request-URI` bidang catatan log.

Contohnya, permintaan `GET` untuk `"s3.amazonaws.com/amzn-s3-demo-bucket1/photos/2019/08/puppy.jpg?x-user=johndoe"` bekerja dengan cara yang sama seperti permintaan untuk `"s3.amazonaws.com/amzn-s3-demo-bucket1/photos/2019/08/puppy.jpg"`, kecuali bahwa string `"x-user=johndoe"` termasuk dalam bidang `Request-URI` untuk catatan log terkait. Fungsi ini tersedia di antarmuka REST saja.

Pertimbangan pemrograman untuk format log akses server yang dapat diperluas

Dari waktu ke waktu, kami dapat memperluas format catatan log akses dengan menambahkan bidang baru pada akhir setiap baris. Oleh karena itu, pastikan kode apa pun yang menguraikan pencatatan akses server dapat menangani bidang jejak yang mungkin tidak dipahaminya.

Menghapus file log Amazon S3

Bucket Amazon S3 dengan pencatatan akses server diaktifkan dapat mengakumulasi banyak objek log server dari waktu ke waktu. Aplikasi Anda mungkin memerlukan log akses ini untuk periode tertentu setelah pembuatan, dan setelah itu, Anda mungkin ingin menghapusnya. Anda dapat menggunakan konfigurasi siklus aktif Amazon S3 untuk menetapkan aturan sehingga Amazon S3 secara otomatis membuat antrean objek ini untuk dihapus di akhir masa aktifnya.

Anda dapat menentukan konfigurasi siklus aktif untuk bagian objek dalam bucket S3 Anda dengan menggunakan prefiks bersama. Jika Anda menentukan prefiks di konfigurasi pencatatan log akses server Anda, Anda dapat mengatur aturan konfigurasi siklus aktif untuk menghapus objek log yang memiliki prefiks tersebut.

Misalnya, anggaplah objek log Anda memiliki prefiks `logs/`. Anda dapat mengatur aturan konfigurasi siklus aktif untuk menghapus semua objek dalam bucket yang memiliki prefiks `logs/` setelah periode waktu tertentu.

Untuk informasi lebih lanjut tentang konfigurasi siklus aktif, lihat Mengelola siklus hidup objek.

Untuk informasi selengkapnya tentang pencatatan akses server, lihat Pencatatan permintaan dengan pencatatan akses server.

Menggunakan log akses server Amazon S3 untuk mengidentifikasi permintaan

Anda dapat mengidentifikasi permintaan Amazon S3 dengan pencatatan akses server Amazon S3.

> **Note**
>
> - Untuk mengidentifikasi permintaan Amazon S3, sebaiknya gunakan peristiwa AWS CloudTrail data alih-alih log akses server Amazon S3. CloudTrail peristiwa data lebih mudah diatur dan berisi lebih banyak informasi. Untuk informasi selengkapnya, lihat Mengidentifikasi permintaan Amazon S3 menggunakan CloudTrail.

- Bergantung pada berapa banyak permintaan akses yang Anda dapatkan, menganalisis log Anda mungkin memerlukan lebih banyak sumber daya atau waktu daripada menggunakan peristiwa CloudTrail data.

Topik

- Melakukan kueri pencatatan akses untuk permintaan dengan menggunakan Amazon Athena
- Mengidentifikasi permintaan Signature Version 2 menggunakan pencatatan akses Amazon S3
- Mengidentifikasi permintaan akses objek dengan menggunakan pencatatan akses Amazon S3

Melakukan kueri pencatatan akses untuk permintaan dengan menggunakan Amazon Athena

Anda dapat mengidentifikasi permintaan Amazon S3 dengan log akses Amazon S3 menggunakan Amazon Athena.

Amazon S3 menyimpan log akses server sebagai objek dalam bucket S3. Sering kali lebih mudah menggunakan alat yang dapat menganalisis log di Amazon S3. Athena mendukung analisis S3 Object dan dapat digunakan untuk mencari log akses Amazon S3.

Example

Contoh berikut menunjukkan bagaimana Anda dapat melakukan kueri log akses server Amazon S3 di Amazon Athena. Ganti yang *`user input placeholders`* digunakan dalam contoh berikut dengan informasi Anda sendiri.

> **Note**
>
> Untuk menentukan lokasi Amazon S3 dalam kueri Athena, Anda harus memberikan URI S3 untuk bucket tempat log Anda dikirimkan. URI ini harus menyertakan nama bucket dan prefiks dalam format berikut: `s3://`*`amzn-s3-demo-bucket1-logs/prefix`*`/`

1. Buka konsol Athena di https://console.aws.amazon.com/athena/.
2. Dalam Editor Kueri, jalankan perintah yang mirip dengan berikut ini. Ganti *`s3_access_logs_db`* dengan nama yang ingin Anda berikan ke basis data Anda.

```
CREATE DATABASE s3_access_logs_db
```

> **Note**
>
> Ini adalah praktik terbaik untuk membuat database yang Wilayah AWS sama dengan bucket S3 Anda.

3. Dalam Query Editor, jalankan perintah yang mirip dengan perintah berikut untuk membuat skema tabel di basis data yang Anda buat di langkah 2. Ganti *s3_access_logs_db.mybucket_logs* dengan nama yang ingin Anda berikan ke meja Anda. Nilai jenis data `STRING` dan `BIGINT` adalah properti log akses. Anda dapat mencari properti ini di Athena. Untuk `LOCATION`, masukkan bucket S3 dan jalur prefiks seperti yang disebutkan sebelumnya.

   Date-based partitioning

```
CREATE EXTERNAL TABLE s3_access_logs_db.mybucket_logs(
 `bucketowner` STRING,
 `bucket_name` STRING,
 `requestdatetime` STRING,
 `remoteip` STRING,
 `requester` STRING,
 `requestid` STRING,
 `operation` STRING,
 `key` STRING,
 `request_uri` STRING,
 `httpstatus` STRING,
 `errorcode` STRING,
 `bytessent` BIGINT,
 `objectsize` BIGINT,
 `totaltime` STRING,
 `turnaroundtime` STRING,
 `referrer` STRING,
 `useragent` STRING,
 `versionid` STRING,
 `hostid` STRING,
 `sigv` STRING,
 `ciphersuite` STRING,
 `authtype` STRING,
 `endpoint` STRING,
 `tlsversion` STRING,
```

```
 `accesspointarn` STRING,
 `aclrequired` STRING)
 PARTITIONED BY (
   `timestamp` string)
ROW FORMAT SERDE
 'org.apache.hadoop.hive.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
 'input.regex'='([^ ]*) ([^ ]*) \\[(.*?)\\] ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*)
 ([^ ]*) (\"[^\"]*\"|-) (-|[0-9]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*)
 ([^ ]*) (\"[^\"]*\"|-) ([^ ]*)(?: ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*)
 ([^ ]*) ([^ ]*) ([^ ]*))?.*$')
STORED AS INPUTFORMAT
 'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
 's3://bucket-name/prefix-name/account-id/region/source-bucket-name/'
 TBLPROPERTIES (
  'projection.enabled'='true',
  'projection.timestamp.format'='yyyy/MM/dd',
  'projection.timestamp.interval'='1',
  'projection.timestamp.interval.unit'='DAYS',
  'projection.timestamp.range'='2024/01/01,NOW',
  'projection.timestamp.type'='date',
  'storage.location.template'='s3://bucket-name/prefix-name/account-id/region/
source-bucket-name/${timestamp}')
```

Non-date-based partitioning

```
CREATE EXTERNAL TABLE `s3_access_logs_db.mybucket_logs`(
  `bucketowner` STRING,
  `bucket_name` STRING,
  `requestdatetime` STRING,
  `remoteip` STRING,
  `requester` STRING,
  `requestid` STRING,
  `operation` STRING,
  `key` STRING,
  `request_uri` STRING,
  `httpstatus` STRING,
  `errorcode` STRING,
  `bytessent` BIGINT,
```

```
  `objectsize` BIGINT,
  `totaltime` STRING,
  `turnaroundtime` STRING,
  `referrer` STRING,
  `useragent` STRING,
  `versionid` STRING,
  `hostid` STRING,
  `sigv` STRING,
  `ciphersuite` STRING,
  `authtype` STRING,
  `endpoint` STRING,
  `tlsversion` STRING,
  `accesspointarn` STRING,
  `aclrequired` STRING)
ROW FORMAT SERDE
  'org.apache.hadoop.hive.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
  'input.regex'='([^ ]*) ([^ ]*) \\[(.*?)\\] ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*)
 ([^ ]*) (\"[^\"]*\"|-) (-|[0-9]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*)
 ([^ ]*) (\"[^\"]*\"|-) ([^ ]*)(?: ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*)
 ([^ ]*) ([^ ]*) ([^ ]*))?.*$')
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
  's3://amzn-s3-demo-bucket1-logs/prefix/'
```

4. Dalam panel navigasi, pada Basis Data, pilih basis data Anda.
5. Pada Tabel, pilih Tabel pratinjau di sebelah nama tabel Anda.

   Di panel Hasil, Anda akan melihat data dari log akses server, seperti `bucketowner`, `bucket`, `requestdatetime`, dan sebagainya. Ini berarti Anda berhasil membuat tabel Athena. Sekarang Anda dapat mengkueri log akses server Amazon S3.

Example — Menunjukkan siapa yang menghapus objek dan kapan (tanda waktu, alamat IP, dan pengguna IAM)

```
SELECT requestdatetime, remoteip, requester, key
FROM s3_access_logs_db.mybucket_logs
WHERE key = 'images/picture.jpg' AND operation like '%DELETE%';
```

Example — Menunjukkan semua operasi yang dilakukan oleh pengguna IAM

```
SELECT *
FROM s3_access_logs_db.mybucket_logs
WHERE requester='arn:aws:iam::123456789123:user/user_name';
```

Example — Menunjukkan semua operasi yang dilakukan pada objek dalam periode waktu tertentu

```
SELECT *
FROM s3_access_logs_db.mybucket_logs
WHERE Key='prefix/images/picture.jpg'
    AND parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z')
    BETWEEN parse_datetime('2017-02-18:07:00:00','yyyy-MM-dd:HH:mm:ss')
    AND parse_datetime('2017-02-18:08:00:00','yyyy-MM-dd:HH:mm:ss');
```

Example — Menunjukkan berapa banyak data yang dipindahkan ke alamat IP tertentu dalam periode waktu tertentu

```
SELECT coalesce(SUM(bytessent), 0) AS bytessenttotal
FROM s3_access_logs_db.mybucket_logs
WHERE remoteip='192.0.2.1'
AND parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2022-06-01','yyyy-MM-dd')
AND parse_datetime('2022-07-01','yyyy-MM-dd');
```

Example — Temukan permintaan IDs untuk kesalahan HTTP 5xx dalam periode waktu tertentu

```
SELECT requestdatetime, key, httpstatus, errorcode, requestid, hostid
FROM s3_access_logs_db.mybucket_logs
WHERE httpstatus like '5%' AND timestamp
BETWEEN '2024/01/29'
AND '2024/01/30'
```

> **Note**
>
> Untuk mengurangi waktu dalam mempertahankan pencatatan, Anda dapat membuat konfigurasi Siklus Hidup S3 untuk bucket pencatatan akses server Anda. Buat aturan konfigurasi siklus aktif untuk menghapus file log secara berkala. Hal ini mengurangi jumlah data yang dianalisis oleh Athena untuk setiap kueri. Untuk informasi selengkapnya, lihat Menyetel konfigurasi Siklus Hidup S3 pada bucket.

Mengidentifikasi permintaan Signature Version 2 menggunakan pencatatan akses Amazon S3

Dukungan Amazon S3 untuk Signature Version 2 akan dinonaktifkan (dihentikan). Setelah itu, Amazon S3 tidak akan menerima lagi permintaan yang menggunakan Signature Version 2, dan semua permintaan harus menggunakan penandatanganan Signature Version 4. Anda dapat mengidentifikasi permintaan akses Signature Version 2 menggunakan log akses Amazon S3.

> **Note**
>
> Untuk mengidentifikasi permintaan Signature Version 2, sebaiknya gunakan peristiwa AWS CloudTrail data, bukan log akses server Amazon S3. CloudTrail peristiwa data lebih mudah diatur dan berisi lebih banyak informasi daripada log akses server. Untuk informasi selengkapnya, lihat Mengidentifikasi permintaan Amazon S3 Signature Version 2 dengan menggunakan CloudTrail.

Example — Menunjukkan semua peminta yang mengirimkan lalu lintas Signature Version 2

```
SELECT requester, sigv, Count(sigv) as sigcount
FROM s3_access_logs_db.mybucket_logs
GROUP BY requester, sigv;
```

Mengidentifikasi permintaan akses objek dengan menggunakan pencatatan akses Amazon S3

Anda dapat menggunakan kueri di log akses server Amazon S3 untuk mengidentifikasi permintaan akses objek Amazon S3, untuk operasi seperti `GET`, `PUT`, dan `DELETE`, lalu menemukan informasi lebih lanjut tentang permintaan tersebut.

Contoh kueri Amazon Athena berikut menunjukkan cara mendapatkan semua permintaan objek `PUT` untuk Amazon S3 dari log akses server.

Example — Menunjukkan semua peminta yang mengirimkan permintaan objek **`PUT`** dalam periode tertentu

```
SELECT bucket_name, requester, remoteip, key, httpstatus, errorcode, requestdatetime
FROM s3_access_logs_db
WHERE operation='REST.PUT.OBJECT' AND
parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2019-07-01:00:42:42','yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2019-07-02:00:42:42','yyyy-MM-dd:HH:mm:ss')
```

Contoh kueri Amazon Athena berikut menunjukkan cara mendapatkan semua permintaan objek `GET` untuk Amazon S3 dari log akses server.

Example — Menunjukkan semua peminta yang mengirimkan permintaan objek **`GET`** dalam periode tertentu

```
SELECT bucket_name, requester, remoteip, key, httpstatus, errorcode, requestdatetime
FROM s3_access_logs_db
WHERE operation='REST.GET.OBJECT' AND
parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2019-07-01:00:42:42','yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2019-07-02:00:42:42','yyyy-MM-dd:HH:mm:ss')
```

Contoh kueri Amazon Athena berikut menunjukkan cara mendapatkan semua permintaan anonim ke bucket S3 Anda dari log akses server.

Example — Menunjukkan semua pemohon anonim yang membuat permintaan ke bucket selama periode tertentu

```
SELECT bucket_name, requester, remoteip, key, httpstatus, errorcode, requestdatetime
FROM s3_access_logs_db.mybucket_logs
WHERE requester IS NULL AND
parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2019-07-01:00:42:42','yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2019-07-02:00:42:42','yyyy-MM-dd:HH:mm:ss')
```

Kueri Amazon Athena berikut menunjukkan cara mengidentifikasi semua permintaan ke bucket S3 Anda yang memerlukan daftar kontrol akses (ACL) untuk otorisasi. Anda dapat menggunakan informasi ini untuk memigrasikan izin ACL tersebut ke kebijakan bucket yang sesuai dan menonaktifkannya. ACLs Setelah membuat kebijakan bucket ini, Anda dapat ACLs menonaktifkan bucket ini. Untuk informasi selengkapnya tentang menonaktifkan ACLs, lihat. Prasyarat untuk menonaktifkan ACLs

Example — Identifikasi semua permintaan yang memerlukan ACL untuk otorisasi

```
SELECT bucket_name, requester, key, operation, aclrequired, requestdatetime
FROM s3_access_logs_db
WHERE aclrequired = 'Yes' AND
parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2022-05-10:00:00:00','yyyy-MM-dd:HH:mm:ss')
AND parse_datetime('2022-08-10:00:00:00','yyyy-MM-dd:HH:mm:ss')
```

Note

- Anda dapat mengubah rentang tanggal sesuai kebutuhan.
- Contoh kueri ini juga dapat berguna untuk pemantauan keamanan. Anda dapat meninjau hasil untuk panggilan `PutObject` atau `GetObject` dari alamat IP yang tidak terduga atau tidak sah atau pemohon, dan untuk mengidentifikasi permintaan anonim ke bucket Anda.
- Kueri ini hanya mengambil informasi dari waktu saat pencatatan log diaktifkan.
- Jika Anda menggunakan AWS CloudTrail log, lihatMengidentifikasi akses ke objek S3 dengan menggunakan CloudTrail.

Memecahkan masalah pencatatan akses server

Topik berikut dapat membantu Anda memecahkan masalah yang mungkin Anda alami saat menyiapkan pencatatan dengan Amazon S3.

Topik

- Pesan kesalahan umum saat mengatur pencatatan
- Memecahkan masalah kegagalan pengiriman

Pesan kesalahan umum saat mengatur pencatatan

Pesan galat umum berikut dapat muncul saat Anda mengaktifkan logging melalui AWS Command Line Interface (AWS CLI) dan AWS SDKs:

Kesalahan: Pencatatan lokasi Cross S3 tidak diizinkan

Jika bucket tujuan (juga dikenal sebagai bucket target) berada di Region yang berbeda dari bucket sumber, kesalahan Pencatatan lokasi Cross S3 tidak diizinkan akan terjadi. Untuk mengatasi kesalahan ini, pastikan bucket tujuan yang dikonfigurasi untuk menerima log akses sama Wilayah AWS dan Akun AWS sebagai bucket sumber.

Kesalahan: Pemilik bucket yang akan dicatat dan bucket target harus sama

Saat Anda mengaktifkan pencatatan akses server, kesalahan ini terjadi jika bucket tujuan yang ditentukan dimiliki akun lain. Untuk mengatasi kesalahan ini, pastikan bucket tujuan Akun AWS sama dengan bucket sumber.

> **Note**
>
> Kami menyarankan Anda memilih bucket tujuan yang berbeda dari bucket sumber. Ketika bucket sumber dan bucket tujuan sama, log tambahan dibuat untuk log yang ditulis ke bucket, yang dapat menambah tagihan penyimpanan Anda. Log tambahan tentang log ini juga dapat menyulitkan untuk menemukan log tertentu yang Anda cari. Untuk manajemen pencatatan yang lebih sederhana, kami sarankan untuk menyimpan log akses dalam bucket yang berbeda. Untuk informasi selengkapnya, lihat the section called “Bagaimana cara mengaktifkan log pengiriman?”.

Kesalahan: Bucket target untuk pencatatan tidak ada

Bucket tujuan harus ada sebelum menyetel konfigurasi. Kesalahan ini menunjukkan bahwa bucket tujuan tidak ada atau tidak dapat ditemukan. Pastikan bahwa nama bucket dieja dengan benar, lalu coba lagi.

Kesalahan: Pemberian target tidak diizinkan untuk bucket yang diberlakukan pemilik bucket

Kesalahan ini menunjukkan bahwa bucket tujuan menggunakan pengaturan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek S3. Pengaturan yang diberlakukan pemilik Bucket tidak mendukung pemberian tujuan (target). Untuk informasi selengkapnya, lihat Izin untuk pengiriman log.

Memecahkan masalah kegagalan pengiriman

Untuk menghindari masalah pencatatan akses server, pastikan Anda mengikuti praktik terbaik berikut ini:

- Grup pengiriman log S3 memiliki akses tulis ke bucket tujuan–Grup pengiriman log S3 mengirimkan log akses server ke bucket tujuan. Kebijakan bucket atau bucket daftar kontrol akses (ACL) dapat digunakan untuk memberikan akses tulis ke bucket tujuan. Namun, kami menyarankan Anda menggunakan kebijakan bucket, bukan ACL. Untuk informasi selengkapnya tentang cara memberikan akses tulis ke bucket tujuan, lihat Izin untuk pengiriman log.

  > **Note**
  >
  > Jika bucket tujuan menggunakan setelan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek, perhatikan hal-hal berikut:
  >
  > - ACLs dinonaktifkan dan tidak lagi memengaruhi izin. Ini berarti Anda tidak dapat memperbarui ACL bucket untuk memberikan akses ke grup pengiriman log S3. Sebagai gantinya, untuk memberikan akses ke pengguna utama layanan pencatatan, Anda harus memperbarui kebijakan bucket untuk bucket tujuan.
  > - Anda tidak dapat menyertakan pemberian tujuan dalam konfigurasi `PutBucketLogging` Anda.

- Kebijakan bucket untuk bucket tujuan memungkinkan akses ke log–Periksa kebijakan bucket dari bucket tujuan. Cari kebijakan bucket untuk setiap pernyataan yang berisi `"Effect": "Deny"`. Kemudian, verifikasi bahwa pernyataan `Deny` tersebut tidak mencegah log akses ditulis ke bucket.
- Kunci Objek S3 tidak diaktifkan di bucket tujuan–Periksa apakah bucket tujuan mengaktifkan Kunci Objek. Kunci Objek memblokir pengiriman log akses server. Anda harus memilih bucket tujuan yang tidak mengaktifkan Kunci Objek.

- Kunci terkelola Amazon S3 (SSE-S3) dipilih jika enkripsi default diaktifkan pada bucket tujuan–Anda dapat menggunakan enkripsi bucket default pada bucket tujuan hanya jika Anda menggunakan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3). Enkripsi sisi server default dengan kunci AWS Key Management Service (AWS KMS) (SSE-KMS) tidak didukung untuk bucket tujuan pencatatan akses server. Untuk informasi selengkapnya tentang cara mengaktifkan enkripsi default, lihatMengonfigurasi enkripsi default.
- Bucket tujuan tidak mengaktifkan Pembayaran Pemohon–Menggunakan bucket Pembayaran Pemohon sebagai bucket tujuan untuk pencatatan akses server tidak didukung. Untuk mengizinkan pengiriman log akses server, nonaktifkan opsi Pembayaran Pemohon di bucket tujuan.
- Tinjau kebijakan kontrol AWS Organizations layanan (SCPs) dan kebijakan kontrol sumber daya (RCPs) — Saat Anda menggunakan AWS Organizations, periksa kebijakan kontrol layanan dan kebijakan kontrol sumber daya untuk memastikan akses Amazon S3 diizinkan. Kebijakan ini menetapkan izin maksimum untuk prinsipal dan sumber daya di akun yang terpengaruh. Cari kebijakan untuk setiap pernyataan yang berisi `"Effect": "Deny"` dan verifikasi bahwa `Deny` pernyataan tidak mencegah log akses apa pun ditulis ke bucket. Untuk informasi selengkapnya, lihat Kebijakan otorisasi AWS Organizations di Panduan AWS Organizations Pengguna.
- Berikan beberapa waktu agar perubahan konfigurasi pencatatan terbaru diterapkan–Mengaktifkan pencatatan akses server untuk pertama kalinya, atau mengubah bucket tujuan untuk log, memerlukan waktu untuk sepenuhnya diterapkan. Mungkin diperlukan waktu lebih dari satu jam agar semua permintaan dicatat dan dikirim dengan benar.

  Untuk memeriksa kegagalan pengiriman log, aktifkan metrik permintaan di Amazon CloudWatch. Jika log tidak dikirim dalam beberapa jam, cari metrik `4xxErrors`, yang dapat menunjukkan kegagalan pengiriman log. Untuk informasi lebih lanjut tentang mengaktifkan metrik permintaan, lihat the section called “Membuat konfigurasi metrik untuk semua objek”.

Memantau metrik dengan Amazon CloudWatch

CloudWatch Metrik Amazon untuk Amazon S3 dapat membantu Anda memahami dan meningkatkan kinerja aplikasi yang menggunakan Amazon S3. Ada beberapa cara yang dapat Anda gunakan CloudWatch dengan Amazon S3.

Metrik penyimpanan harian untuk bucket

Pantau penyimpanan bucket menggunakan CloudWatch, yang mengumpulkan dan memproses data penyimpanan dari Amazon S3 menjadi metrik harian yang dapat dibaca. Metrik penyimpanan

untuk Amazon S3 ini dilaporkan satu kali sehari dan diberikan kepada semua pelanggan tanpa biaya tambahan.

Metrik permintaan

Memantau permintaan Amazon S3 untuk mengidentifikasi dan menindaklanjuti masalah operasional dengan cepat. Metrik tersedia dengan interval 1 menit setelah beberapa latensi pemrosesan. CloudWatch Metrik ini ditagih dengan tarif yang sama dengan metrik CloudWatch khusus Amazon. Untuk informasi tentang CloudWatch harga, lihat CloudWatch harga Amazon. Untuk mempelajari cara memilih mendapatkan metrik ini, lihat CloudWatch konfigurasi metrik.

Saat diaktifkan, metrik permintaan dilaporkan untuk semua operasi objek. Secara default, metrik 1 menit ini tersedia di tingkat bucket Amazon S3. Anda juga dapat menentukan filter untuk metrik menggunakan prefiks bersama, tag objek, atau titik akses:

- Titik akses–Titik akses diberi nama titik akhir jaringan yang dilampirkan ke bucket dan menyederhanakan pengelolaan akses data pada skala untuk set data bersama di S3. Dengan filter titik akses, Anda dapat memperoleh wawasan tentang penggunaan titik akses Anda. Untuk informasi lebih lanjut tentang titik akses, lihat Pemantauan dan pencatatan titik akses.
- Prefiks–Meskipun model data Amazon S3 adalah struktur datar, Anda dapat menggunakan prefiks untuk menyimpulkan hierarki. Prefiks mirip dengan nama direktori yang mengaktifkan Anda untuk membuat grup objek yang sama bersama-sama dalam bucket. Konsol S3 mendukung prefiks dengan konsep folder. Jika Anda memfilter menurut prefiks, objek yang memiliki prefiks yang sama akan disertakan dalam konfigurasi metrik. Untuk informasi selengkapnya tentang prefiks, lihat Organisasi objek menggunakan prefiks.
- Tag–Tag adalah pasangan nama nilai-kunci yang dapat Anda tambahkan ke objek. Tag membantu Anda untuk menemukan dan menata objek dengan mudah. Anda juga dapat menggunakan tag sebagai filter untuk konfigurasi metrik sehingga hanya objek dengan tag tersebut yang disertakan dalam konfigurasi metrik. Untuk informasi selengkapnya tentang tag objek, lihat Mengategorikan penyimpanan Anda menggunakan tag.

Untuk menyelaraskan metrik ini ke aplikasi bisnis, alur kerja, atau organisasi internal tertentu, Anda dapat memfilter pada prefiks bersama, tag objek, atau titik akses.

Metrik replikasi

Pantau jumlah total operasi S3 API yang menunggu replikasi, ukuran total objek yang menunggu replikasi, waktu replikasi maksimum ke tujuan Wilayah AWS, dan jumlah total operasi yang gagal replikasi. Aturan replikasi yang memiliki Kontrol Waktu Replikasi S3 (S3 RTC) atau metrik replikasi S3 yang diaktifkan akan menerbitkan metrik replikasi.

Untuk informasi lebih lanjut, lihat Memantau replikasi dengan metrik, pemberitahuan acara, dan status atau Memenuhi persyaratan kepatuhan dengan Kontrol Waktu Replikasi S3.

Metrik Lensa Penyimpanan Amazon S3

Anda dapat memublikasikan metrik penggunaan dan aktivitas Lensa Penyimpanan S3 CloudWatch ke Amazon untuk membuat tampilan terpadu kesehatan operasional Anda di dasbor. CloudWatch Metrik Lensa Penyimpanan S3 tersedia di namespace `AWS/S3/Storage-Lens`. Opsi CloudWatch penerbitan tersedia untuk dasbor Lensa Penyimpanan S3 yang ditingkatkan ke metrik dan rekomendasi lanjutan. Anda dapat mengaktifkan opsi CloudWatch penerbitan untuk konfigurasi dasbor baru atau yang sudah ada di S3 Storage Lens.

Untuk informasi selengkapnya, lihat Pantau metrik Lensa Penyimpanan S3 di CloudWatch.

Semua CloudWatch statistik disimpan untuk jangka waktu 15 bulan sehingga Anda dapat mengakses informasi historis dan mendapatkan perspektif yang lebih baik tentang kinerja aplikasi atau layanan web Anda. Untuk informasi selengkapnya CloudWatch, lihat Apa itu Amazon CloudWatch? di Panduan CloudWatch Pengguna Amazon. Anda mungkin memerlukan beberapa konfigurasi tambahan untuk CloudWatch alarm Anda, tergantung pada kasus penggunaan Anda. Misalnya, Anda dapat menggunakan ekspresi matematika metrik untuk membuat alarm. Untuk informasi selengkapnya, lihat Menggunakan CloudWatch metrik, Menggunakan matematika metrik, Menggunakan CloudWatch alarm Amazon, dan Membuat CloudWatch alarm berdasarkan ekspresi matematika metrik di CloudWatch Panduan Pengguna Amazon.

Pengiriman CloudWatch metrik upaya terbaik

CloudWatch metrik disampaikan atas dasar upaya terbaik. Sebagian besar permintaan untuk objek Amazon S3 yang memiliki metrik permintaan menghasilkan titik data yang dikirim. CloudWatch

Kelengkapan dan ketepatan waktu metrik tidak dijamin. Titik data untuk permintaan tertentu mungkin dikembalikan dengan stempel waktu yang lebih lambat daripada ketika permintaan tersebut sebenarnya diproses. Titik data selama satu menit mungkin tertunda sebelum tersedia CloudWatch, atau mungkin tidak dikirimkan sama sekali. CloudWatch metrik permintaan memberi Anda gambaran tentang sifat lalu lintas terhadap bucket Anda dalam waktu hampir nyata. Ini tidak dimaksudkan untuk menjadi pencatatan akuntasi yang lengkap atas semua permintaan.

Berdasarkan sifat upaya terbaik dari fitur ini, laporan yang tersedia di Dasbor Manajemen Penagihan & Biaya mungkin menyertakan satu atau lebih permintaan akses yang tidak muncul dalam metrik bucket.

Untuk informasi selengkapnya, lihat topik berikut.

Topik

- Metrik dan dimensi
- Mengakses metrik CloudWatch
- CloudWatch konfigurasi metrik

Metrik dan dimensi

Metrik dan dimensi penyimpanan yang dikirimkan Amazon S3 ke CloudWatch Amazon tercantum dalam tabel berikut.

Pengiriman CloudWatch metrik upaya terbaik

CloudWatch metrik disampaikan atas dasar upaya terbaik. Sebagian besar permintaan untuk objek Amazon S3 yang memiliki metrik permintaan menghasilkan titik data yang dikirim. CloudWatch

Kelengkapan dan ketepatan waktu metrik tidak dijamin. Titik data untuk permintaan tertentu mungkin dikembalikan dengan stempel waktu yang lebih lambat daripada ketika permintaan tersebut sebenarnya diproses. Titik data selama satu menit mungkin tertunda sebelum tersedia CloudWatch, atau mungkin tidak dikirimkan sama sekali. CloudWatch metrik permintaan memberi Anda gambaran tentang sifat lalu lintas terhadap bucket Anda dalam waktu hampir nyata. Ini tidak dimaksudkan untuk menjadi pencatatan akuntasi yang lengkap atas semua permintaan.

Berdasarkan sifat upaya terbaik dari fitur ini, laporan yang tersedia di Dasbor Manajemen Penagihan & Biaya mungkin menyertakan satu atau lebih permintaan akses yang tidak muncul dalam metrik bucket.

Topik

- Metrik penyimpanan harian Amazon S3 untuk bucket di CloudWatch
- Metrik permintaan Amazon S3 di CloudWatch
- Metrik replikasi S3 di CloudWatch
- Metrik Lensa Penyimpanan S3 di CloudWatch
- Metrik permintaan Lambda Objek S3 di CloudWatch
- Dimensi Amazon S3 di CloudWatch
- Dimensi replikasi S3 di CloudWatch
- Dimensi Lensa Penyimpanan S3 di CloudWatch

- Dimensi permintaan Lambda Objek S3 di CloudWatch
- Metrik penggunaan Amazon S3

Metrik penyimpanan harian Amazon S3 untuk bucket di CloudWatch

Namespace `AWS/S3` mencakup metrik penyimpanan harian berikut untuk bucket.

Metrik	Deskripsi
`BucketSizeBytes`	Jumlah data dalam byte yang disimpan dalam bucket di kelas penyimpanan berikut: • Reduced Redundancy Storage (RRS) (`REDUCED_REDUNDANCY`) • S3 Express One Zone (`EXPRESS_ONEZONE`) • S3 Glacier Deep Archive (`DEEP_ARCHIVE`) • S3 Glacier Flexible Retrieval (`GLACIER`) • S3 Glacier Instant Retrieval (`GLACIER_IR`) • S3 Intelligent-Tiering (`INTELLIGENT_TIERING`) • Akses S3 Satu Zona Jarang () `ONEZONE_IA` • S3 Standard (`STANDARD`) • S3 Standard-Infrequent Access (`STANDARD_IA`) Nilai ini dihitung dengan menjumlahkan ukuran semua objek dan metadata (seperti nama bucket) dalam bucket (baik objek saat ini maupun noncurrent), termasuk ukuran semua bagian untuk semua unggahan multipart yang tidak lengkap ke bucket. **Note** Kelas penyimpanan S3 Express One Zone hanya tersedia untuk bucket direktori. Filter tipe penyimpanan yang valid (lihat `StorageType` dimensinya):

Metrik	Deskripsi
	• Penyimpanan Redundansi Berkurang (RRS): `ReducedRedundancyStorage` • S3 Express Satu Zona: `ExpressOneZoneStorage` • Arsip Dalam Gletser S3:,,, `DeepArchiveObjectOverhead` `DeepArchiveS3ObjectOverhead` `DeepArchiveStagingStorage` `DeepArchiveStorage` • Pengambilan Fleksibel Gletser S3:,,, `GlacierObjectOverhead` `GlacierS3ObjectOverhead` `GlacierStagingStorage` `GlacierStorage` • Pengambilan Instan Gletser S3:, `GlacierInstantRetrievalStorage` `GlacierIRSizeOverhead` • Tingkat Cerdas S3:,,,, `IntelligentTieringAAStorage` `IntelligentTieringAIAStorage` `IntelligentTieringDAAStorage` `IntelligentTieringFAStorage` `IntelligentTieringIAStorage` • Akses S3 Satu Zona Jarang:, `OneZoneIASizeOverhead` `OneZoneIAStorage` • Standar S3: `StandardStorage` • Akses Jarang Standar S3:,, `StandardIAObjectOverhead` `StandardIASizeOverhead` `StandardIAStorage` Unit: Bita Statistik yang valid: Rata-rata Untuk informasi lebih lanjut tentang `StorageType` dimensi, lihatthe section called “Dimensi Amazon S3 di CloudWatch”.

Metrik	Deskripsi
`NumberOfObjects`	Total jumlah penyimpanan objek dalam bucket tujuan umum untuk semua kelas penyimpanan. Nilai ini dihitung dengan menghitung semua objek dalam bucket, yang mencakup objek saat ini dan yang tidak berjalan, penanda hapus, dan jumlah total bagian untuk semua unggahan multibagian yang tidak lengkap ke bucket. Untuk bucket direktori dengan objek di kelas penyimpanan S3 Express One Zone, nilai ini dihitung dengan menghitung semua objek dalam bucket, tetapi tidak termasuk beberapa unggahan yang tidak lengkap ke bucket. Filter jenis penyimpanan yang valid: `AllStorageTypes` (lihat dimensi `StorageType`) Unit: Hitungan Statistik yang valid: Rata-rata

Metrik permintaan Amazon S3 di CloudWatch

Namespace `AWS/S3` menyertakan metrik permintaan berikut. Metrik ini mencakup permintaan yang tidak dapat ditagih (dalam kasus `GET` permintaan dari `CopyObject` dan Replikasi).

> **Note**
>
> Metrik permintaan Amazon S3 CloudWatch tidak didukung untuk bucket direktori.

Metrik	Deskripsi
`AllRequests`	Total jumlah permintaan HTTP yang dilakukan ke bucket Amazon S3, apa pun jenisnya. Jika Anda menggunakan konfigurasi metrik dengan filter, maka metrik ini hanya mengembalikan permintaan HTTP yang memenuhi persyaratan filter. Unit: Hitungan Statistik yang valid: Jumlah

Metrik	Deskripsi
`GetRequests`	Jumlah `GET` permintaan HTTP yang dibuat untuk objek dalam bucket Amazon S3. Ini tidak termasuk operasi daftar. Metrik ini ditambahkan untuk sumber setiap `CopyObject` permintaan. Unit: Hitungan Statistik yang valid: Jumlah **Note** Permintaan berorientasi daftar paginasi, seperti ListMultipartUploads,, ListParts, dan lainnya ListObjectVersions, tidak termasuk dalam metrik ini.
`PutRequests`	Jumlah `PUT` permintaan HTTP yang dibuat untuk objek dalam bucket Amazon S3. Metrik ini ditambahkan untuk tujuan setiap `CopyObject` permintaan. Unit: Hitungan Statistik yang valid: Jumlah
`DeleteRequests`	Jumlah `DELETE` permintaan HTTP yang dibuat untuk objek dalam bucket Amazon S3. Metrik ini juga mencakup DeleteObjectspermintaan. Metrik ini menunjukkan jumlah permintaan yang dibuat, bukan jumlah objek yang dihapus. Unit: Hitungan Statistik yang valid: Jumlah
`HeadRequests`	Jumlah permintaan HTTP `HEAD` yang dibuat ke bucket Amazon S3. Unit: Hitungan Statistik yang valid: Jumlah

Metrik	Deskripsi
`PostRequests`	Jumlah permintaan HTTP `POST` yang dibuat ke bucket Amazon S3. Unit: Hitungan Statistik yang valid: Jumlah **Note** DeleteObjectsdan SelectObjectContentpermintaan tidak termasuk dalam metrik ini.
`SelectRequests`	Jumlah SelectObjectContentpermintaan Amazon S3 yang dibuat untuk objek di bucket Amazon S3. Unit: Hitungan Statistik yang valid: Jumlah
`SelectBytesScanned`	Jumlah byte data yang dipindai dengan permintaan Amazon S3 di SelectObjectContentbucket Amazon S3. Unit: Byte Statistik yang valid: Rata-rata (byte per permintaan), Jumlah (byte per periode), Jumlah Sampel, Min, Maks (sama dengan p100), persentil apa pun antara p0,0 dan p99,9
`SelectBytesReturned`	Jumlah byte data yang dikembalikan dengan permintaan Amazon SelectObjectContentS3 di bucket Amazon S3. Unit: Byte Statistik yang valid: Rata-rata (byte per permintaan), Jumlah (byte per periode), Jumlah Sampel, Min, Maks (sama dengan p100), persentil apa pun antara p0,0 dan p99,9

Metrik	Deskripsi
`ListRequests`	Jumlah permintaan HTTP yang mencantumkan daftar isi bucket. Unit: Hitungan Statistik yang valid: Jumlah
`BytesDownloaded`	Jumlah byte yang diunduh untuk permintaan yang dibuat ke bucket Amazon S3, yang tanggapannya mencakup satu isi. Unit: Byte Statistik yang valid: Rata-rata (byte per permintaan), Jumlah (byte per periode), Jumlah Sampel, Min, Maks (sama dengan p100), persentil apa pun antara p0,0 dan p99,9
`BytesUploaded`	Jumlah byte yang diunggah untuk permintaan yang dibuat ke bucket Amazon S3, yang permintaannya mencakup satu isi. Unit: Byte Statistik yang valid: Rata-rata (byte per permintaan), Jumlah (byte per periode), Jumlah Sampel, Min, Maks (sama dengan p100), persentil apa pun antara p0,0 dan p99,9
`4xxErrors`	Jumlah permintaan kode status kesalahan klien HTTP 4 xx yang dibuat ke bucket Amazon S3 dengan nilai 0 atau 1. Statistik rata-rata menunjukkan tingkat kesalahan, dan statistik Jumlah menunjukkan jumlah jenis kesalahan tersebut, selama setiap periode. Unit: Hitungan Statistik yang valid: Rata-rata (laporan per permintaan), Jumlah (laporan per periode), Min, Maks, Jumlah Sampel

Metrik	Deskripsi
`5xxErrors`	Jumlah permintaan kode status kesalahan server HTTP 5 xx yang dibuat ke bucket Amazon S3 dengan nilai 0 atau 1. Statistik rata-rata menunjukkan tingkat kesalahan, dan statistik Jumlah menunjukkan jumlah jenis kesalahan tersebut, selama setiap periode. Unit: Hitungan Statistik yang valid: Rata-rata (laporan per permintaan), Jumlah (laporan per periode), Min, Maks, Jumlah Sampel
`FirstByte Latency`	Waktu per permintaan dari permintaan lengkap yang diterima oleh bucket Amazon S3 saat respons mulai dikembalikan. Unit: Milidetik Statistik yang valid: Rata-rata, Jumlah, Min, Maks (sama seperti p100), Jumlah Sampel, persentil apa pun antara p0,0 dan p100
`TotalRequ estLatency`	Waktu per permintaan yang berlalu dari byte pertama yang diterima hingga byte terakhir yang dikirim ke bucket Amazon S3. Metrik ini mencakup waktu yang dibutuhkan untuk menerima isi permintaan dan mengirimkan isi respons, yang tidak termasuk dalam `FirstByte Latency` . Unit: Milidetik Statistik yang valid: Rata-rata, Jumlah, Min, Maks (sama seperti p100), Jumlah Sampel, persentil apa pun antara p0,0 dan p100

Metrik replikasi S3 di CloudWatch

Anda dapat memantau kemajuan replikasi dengan metrik replikasi S3 dengan melacak byte tertunda, operasi tertunda, dan latensi replikasi. Untuk informasi selengkapnya, lihat Memantau progres dengan metrik replikasi.

> **Note**
>
> Anda dapat mengaktifkan alarm untuk metrik replikasi Anda di Amazon. CloudWatch Saat Anda menyiapkan alarm untuk metrik replikasi Anda, tetapkan bidang Perawatan data yang hilang untuk Tangani data yang hilang sebagai diabaikan (jaga status alarm).

Metrik	Deskripsi
`ReplicationLatency`	Jumlah maksimum detik dimana tujuan replikasi Wilayah AWS berada di belakang sumber Wilayah AWS untuk aturan replikasi yang diberikan. Unit: detik Statistik yang valid: Maks
`BytesPendingReplication`	Jumlah total byte objek yang menunggu replikasi untuk aturan replikasi yang diberikan. Unit: Byte Statistik yang valid: Maks
`OperationsPendingReplication`	Jumlah operasi yang menunggu replikasi untuk aturan replikasi yang diberikan. Unit: Hitungan Statistik yang valid: Maks
`OperationsFailedReplication`	Jumlah operasi yang gagal untuk mereplikasi untuk aturan replikasi yang diberikan. Unit: Hitungan Statistik yang valid: Jumlah (jumlah total operasi yang gagal), Rata-rata (tingkat kegagalan), Jumlah Sampel (jumlah total operasi replikasi)

Metrik Lensa Penyimpanan S3 di CloudWatch

Anda dapat memublikasikan metrik penggunaan dan aktivitas Lensa Penyimpanan S3 CloudWatch ke Amazon untuk membuat tampilan terpadu kesehatan operasional Anda di dasbor. CloudWatch Metrik Lensa Penyimpanan S3 dipublikasikan ke `AWS/S3/Storage-Lens` namespace di. CloudWatch Opsi CloudWatch penerbitan tersedia untuk dasbor Lensa Penyimpanan S3 yang telah ditingkatkan ke metrik dan rekomendasi lanjutan.

Untuk daftar metrik Lensa Penyimpanan S3 yang dipublikasikan CloudWatch, lihat. Glosarium metrik Lensa Penyimpanan Amazon S3 Untuk daftar lengkap dimensi, lihat Dimensi.

Metrik permintaan Lambda Objek S3 di CloudWatch

S3 Lambda Objek menyertakan metrik permintaan berikut.

Metrik	Deskripsi
`AllRequests`	Total jumlah permintaan HTTP yang dibuat ke bucket Amazon S3 dengan menggunakan Titik Akses Lambda Object. Unit: Hitungan Statistik yang valid: Jumlah
`GetRequests`	Jumlah permintaan HTTP `GET` yang dibuat untuk objek dengan menggunakan Titik Akses Lambda Object. Metrik ini tidak mencakup operasi daftar. Unit: Hitungan Statistik yang valid: Jumlah
`BytesUploaded`	Jumlah byte yang diunggah ke bucket Amazon S3 menggunakan Titik Akses Lambda Object, yang permintaannya menyertakan isi. Unit: Byte Statistik yang valid: Rata-rata (byte per permintaan), Jumlah (byte per periode), Jumlah Sampel, Min, Maks (sama dengan p100), persentil apa pun antara p0,0 dan p99,9

Metrik	Deskripsi
`PostRequests`	Jumlah permintaan HTTP `POST` yang dibuat ke bucket Amazon S3 dengan menggunakan Titik Akses Lambda Object. Unit: Hitungan Statistik yang valid: Jumlah
`PutRequests`	Jumlah permintaan HTTP `PUT` yang dibuat untuk objek dalam bucket Amazon S3 menggunakan Titik Akses Lambda Object. Unit: Hitungan Statistik yang valid: Jumlah
`DeleteRequests`	Jumlah permintaan HTTP `DELETE` yang dibuat untuk objek dalam bucket Amazon S3 menggunakan Titik Akses Lambda Objek. Metrik ini mencakup DeleteObjectspermintaan. Metrik ini menunjukkan jumlah permintaan yang dibuat, bukan jumlah objek yang dihapus. Unit: Hitungan Statistik yang valid: Jumlah
`BytesDownloaded`	Jumlah byte yang diunduh untuk permintaan yang dibuat ke bucket Amazon S3 menggunakan Titik Akses Lambda Object, yang tanggapan nya mencakup satu isi. Unit: Byte Statistik yang valid: Rata-rata (byte per permintaan), Jumlah (byte per periode), Jumlah Sampel, Min, Maks (sama dengan p100), persentil apa pun antara p0,0 dan p99,9

Metrik	Deskripsi
`FirstByteLatency`	Waktu per permintaan dari permintaan lengkap yang diterima oleh bucket Amazon S3 melalui Titik Akses Lambda Objek hingga saat respons mulai dikembalikan. Metrik ini tergantung pada waktu berjalan pada fungsi AWS Lambda untuk mengubah objek sebelum fungsi mengembalikan byte ke Titik Akses Lambda Objek. Unit: Milidetik Statistik yang valid: Rata-rata, Jumlah, Min, Maks (sama seperti p100), Jumlah Sampel, persentil apa pun antara p0,0 dan p100
`TotalRequestLatency`	Waktu per permintaan yang berlalu dari byte pertama yang diterima hingga byte terakhir yang dikirim ke Titik Akses Lambda Objek. Metrik ini mencakup waktu yang dibutuhkan untuk menerima isi permintaan dan mengirimkan isi respons, yang tidak termasuk dalam `FirstByteLatency` . Unit: Milidetik Statistik yang valid: Rata-rata, Jumlah, Min, Maks (sama seperti p100), Jumlah Sampel, persentil apa pun antara p0,0 dan p100
`HeadRequests`	Jumlah permintaan HTTP `HEAD` yang dibuat ke bucket Amazon S3 dengan menggunakan Titik Akses Lambda Objek. Unit: Hitungan Statistik yang valid: Jumlah
`ListRequests`	Jumlah permintaan HTTP `GET` yang mencantumkan daftar isi bucket Amazon S3. Metrik ini mencakup kedua operasi `ListObjects` dan `ListObjectsV2` . Unit: Hitungan Statistik yang valid: Jumlah

Metrik	Deskripsi
`4xxErrors`	Jumlah permintaan kode status kesalahan klien HTTP 4 xx yang dibuat ke bucket Amazon S3 dengan menggunakan Object Lambda Access Point dengan nilai 0 atau 1. Statistik rata-rata menunjukkan tingkat kesalahan, dan statistik Jumlah menunjukkan jumlah jenis kesalahan tersebut, selama setiap periode. Unit: Hitungan Statistik yang valid: Rata-rata (laporan per permintaan), Jumlah (laporan per periode), Min, Maks, Jumlah Sampel
`5xxErrors`	Jumlah permintaan kode status kesalahan server HTTP 5 xx yang dibuat ke bucket Amazon S3 dengan menggunakan Object Lambda Access Point dengan nilai 0 atau 1. Statistik rata-rata menunjukkan tingkat kesalahan, dan statistik Jumlah menunjukkan jumlah jenis kesalahan tersebut, selama setiap periode. Unit: Hitungan Statistik yang valid: Rata-rata (laporan per permintaan), Jumlah (laporan per periode), Min, Maks, Jumlah Sampel
`ProxiedRequests`	Jumlah permintaan HTTP ke Titik Akses Lambda Objek yang mengembalikan respons API Amazon S3 standar. (Permintaan tersebut tidak memiliki fungsi Lambda yang dikonfigurasi.) Unit: Hitungan Statistik yang valid: Jumlah
`InvokedLambda`	Jumlah permintaan HTTP ke S3 Object di mana fungsi Lambda dipanggil . Unit: Hitungan Statistik yang valid: Jumlah

Metrik	Deskripsi
`LambdaResponseRequests`	Jumlah permintaan `WriteGetObjectResponse` yang dibuat oleh fungsi Lambda. Metrik ini hanya berlaku untuk permintaan `GetObject`.
`LambdaResponse4xx`	Jumlah kesalahan klien HTTP 4 xx yang terjadi saat memanggil `WriteGetObjectResponse` dari fungsi Lambda. Metrik ini memberikan informasi yang sama seperti `4xxErrors`, tetapi hanya untuk panggilan `WriteGetObjectResponse`.
`LambdaResponse5xx`	Jumlah kesalahan server HTTP 5 xx yang terjadi saat memanggil `WriteGetObjectResponse` dari fungsi Lambda. Metrik ini memberikan informasi yang sama seperti `5xxErrors`, tetapi hanya untuk panggilan `WriteGetObjectResponse`.

Dimensi Amazon S3 di CloudWatch

Dimensi berikut digunakan untuk memfilter metrik Amazon S3.

Dimensi	Deskripsi
`BucketName`	Dimensi ini memfilter data yang Anda minta untuk bucket yang diidentifikasi saja.
`StorageType`	Dimensi ini memfilter data yang telah Anda simpan dalam bucket dengan jenis penyimpanan berikut: • `DeepArchiveObjectOverhead` –Untuk setiap objek yang diarsipkan, S3 Glacier menambahkan penyimpanan 32 KB untuk indeks dan metadata terkait. Data ekstra ini diperlukan untuk mengidentifikasi dan memulihkan objek Anda. Anda dikenakan tarif biaya S3 Glacier Deep Archive untuk penyimpanan tambahan ini. • `DeepArchiveS3ObjectOverhead` –Untuk setiap objek yang diarsipkan ke S3 Glacier Deep Archive, Amazon S3 menggunakan penyimpanan 8 KB untuk nama objek dan

Dimensi	Deskripsi
	metadata lainnya. Anda dikenakan tarif Standar S3 untuk penyimpanan tambahan ini. • `DeepArchiveStagingStorage` — Jumlah byte yang digunakan untuk bagian dari objek upload multipart sebelum `CompleteMultipartUpload` permintaan selesai pada objek di kelas penyimpanan S3 Glacier Deep Archive. • `DeepArchiveStorage` –Jumlah byte yang digunakan untuk objek dalam kelas penyimpanan S3 Glacier Deep Archive. • `ExpressOneZoneStorage` –Jumlah byte yang digunakan untuk objek dalam kelas penyimpanan S3 Express One Zone. • `GlacierInstantRetrievalStorage` –Jumlah byte yang digunakan untuk objek dalam kelas penyimpanan S3 Glacier Instant Retrieval. • `GlacierIRSizeOverhead` –Jumlah byte yang digunakan untuk objek berukuran lebih kecil dari 128 KB di kelas penyimpanan Instant Retrieval S3 Glacier Instant Retrieval. • `GlacierObjectOverhead` –Untuk setiap objek yang diarsipkan, S3 Glacier menambahkan penyimpanan 32 KB untuk indeks dan metadata terkait. Data ekstra ini diperluka n untuk mengidentifikasi dan memulihkan objek Anda. Anda dikenakan tarif S3 Glacier Flexible Retrieval untuk penyimpan an tambahan ini. • `GlacierS3ObjectOverhead` –Untuk setiap objek yang diarsipkan ke S3 Glacier Flexible Retrieval, Amazon S3 menggunakan penyimpanan 8 KB untuk nama objek dan metadata lainnya. Anda dikenakan tarif Standar S3 untuk penyimpanan tambahan ini. • `GlacierStagingStorage` — Jumlah byte yang digunakan untuk bagian dari objek upload multipart sebelum `CompleteMultipartUpload` permintaan selesai pada objek di kelas penyimpanan S3 Glacier Flexible Retrieval.

Dimensi	Deskripsi
	• `GlacierStorage` –Jumlah byte yang digunakan untuk objek dalam kelas penyimpanan S3 Glacier Flexible Retrieval. • `IntAAObjectOverhead` –Untuk setiap objek dalam kelas penyimpanan `INTELLIGENT_TIERING` di tingkat Archive Access, S3 Glacier menambahkan 32 KB penyimpanan untuk indeks dan metadata terkait. Data ekstra ini diperlukan untuk mengidentifikasi dan memulihkan objek Anda. Anda dikenakan tarif S3 Glacier Flexible Retrieval untuk penyimpanan tambahan ini. • `IntAAS3ObjectOverhead` –Untuk setiap objek dalam kelas penyimpanan `INTELLIGENT_TIERING` di tingkat Archive Access, Amazon S3 menggunakan penyimpanan sebesar 8 KB untuk nama objek dan metadata lainnya. Anda dikenakan tarif Standar S3 untuk penyimpanan tambahan ini. • `IntDAAObjectOverhead` –Untuk setiap objek dalam kelas penyimpanan `INTELLIGENT_TIERING` di tingkat Deep Archive Access, S3 Glacier menambahkan penyimpanan sebesar 32 KB untuk indeks dan metadata terkait. Data ekstra ini diperlukan untuk mengidentifikasi dan memulihkan objek Anda. Anda dikenakan tarif biaya S3 Glacier Deep Archive untuk penyimpanan tambahan ini. • `IntDAAS3ObjectOverhead` –Untuk setiap objek dalam kelas penyimpanan `INTELLIGENT_TIERING` di tingkat Deep Archive Access, Amazon S3 menambahkan penyimpanan sebesar 8 KB untuk indeks dan metadata terkait. Data ekstra ini diperlukan untuk mengidentifikasi dan memulihkan objek Anda. Anda dikenakan tarif Standar S3 untuk penyimpanan tambahan ini. • `IntelligentTieringAAStorage` –Jumlah byte yang digunakan untuk objek di tingkat Archive Access dari kelas penyimpanan `INTELLIGENT_TIERING` .

Dimensi	Deskripsi
	• `IntelligentTieringAIAStorage` — Jumlah byte yang digunakan untuk objek di tingkat Archive Instant Access kelas penyimpanan `INTELLIGENT_TIERING` . • `IntelligentTieringDAAStorage` –Jumlah byte yang digunakan untuk objek di tingkat Deep Archive Access dari kelas penyimpanan `INTELLIGENT_TIERING` . • `IntelligentTieringFAStorage` –Jumlah byte yang digunakan untuk objek di tingkat Frequent Access dari kelas penyimpanan `INTELLIGENT_TIERING` . • `IntelligentTieringIAStorage` –Jumlah byte yang digunakan untuk objek di tingkat Infrequent Access dari kelas penyimpanan `INTELLIGENT_TIERING` . • `OneZoneIASizeOverhead` –Jumlah byte yang digunakan untuk objek berukuran lebih kecil dari 128 KB dalam kelas penyimpanan `ONEZONE_IA` . • `OneZoneIAStorage` –Jumlah byte yang digunakan untuk objek di kelas penyimpanan S3 One Zone-Infrequent Access (`ONEZONE_IA`). • `ReducedRedundancyStorage` –Jumlah byte yang digunakan untuk objek dalam kelas Reduced Redundancy Storage (RRS). • `StandardIASizeOverhead` –Jumlah byte yang digunakan untuk objek berukuran lebih kecil dari 128 KB dalam kelas penyimpanan `STANDARD_IA` . • `StandardIAStorage` — Jumlah byte yang digunakan untuk objek di kelas penyimpanan S3 Standard-Infrequent Access (). `STANDARD_IA` • `StandardStorage` –Jumlah byte yang digunakan untuk objek dalam kelas penyimpanan `STANDARD`.

Dimensi	Deskripsi
`FilterId`	Dimensi ini memfilter konfigurasi metrik yang Anda tentukan untuk metrik permintaan di bucket. Saat Anda membuat konfigurasi metrik, Anda menentukan ID filter (misalnya, prefiks, tag, atau titik akses). Untuk informasi lebih lanjut, lihat Membuat konfigurasi metrik.

Dimensi replikasi S3 di CloudWatch

Dimensi berikut digunakan untuk menyaring metrik Replikasi S3.

Dimensi	Deskripsi
`SourceBucket`	Nama objek ember direplikasi dari.
`DestinationBucket`	Nama objek bucket direplikasi ke.
`RuleId`	Pengidentifikasi unik untuk aturan yang memicu metrik replikasi ini diperbarui.

Dimensi Lensa Penyimpanan S3 di CloudWatch

Untuk daftar dimensi yang digunakan untuk memfilter metrik Lensa Penyimpanan S3 CloudWatch, lihat. Dimensi

Dimensi permintaan Lambda Objek S3 di CloudWatch

Dimensi berikut digunakan untuk memfilter data dari Titik Akses Lambda Objek.

Dimensi	Deskripsi
`AccessPointName`	Nama titik akses tempat permintaan sedang dibuat.
`DataSourceARN`	Sumber Titik Akses Lambda Objek mengambil data dari. Jika permintaan memanggil fungsi Lambda, ini merujuk ke Nama Sumber Daya Amazon Lambda (ARN). Kalau tidak, ini mengacu pada titik akses ARN.

Metrik penggunaan Amazon S3

Anda dapat menggunakan metrik CloudWatch penggunaan untuk memberikan visibilitas ke dalam penggunaan sumber daya akun Anda. Gunakan metrik ini untuk memvisualisasikan penggunaan layanan Anda saat ini pada CloudWatch grafik dan dasbor.

Metrik penggunaan Amazon S3 sesuai dengan kuota layanan. AWS Anda dapat mengonfigurasi alarm yang memberi tahu Anda saat penggunaan mendekati kuota layanan. Untuk informasi selengkapnya tentang CloudWatch integrasi dengan kuota layanan, lihat metrik AWS penggunaan di CloudWatch Panduan Pengguna Amazon.

Amazon S3 menerbitkan metrik berikut di namespace. `AWS/Usage`

Metrik	Deskripsi
`ResourceCount`	Jumlah sumber daya yang ditentukan yang berjalan di akun Anda. Sumber daya tersebut ditentukan oleh dimensi yang dikaitkan dengan metrik.

Dimensi berikut digunakan untuk menyempurnakan metrik penggunaan yang diterbitkan oleh Amazon S3.

Dimensi	Deskripsi
`Service`	Nama AWS layanan yang berisi sumber daya. Untuk metrik penggunaan Amazon S3, nilai untuk dimensi ini adalah. `S3`
`Type`	Tipe entitas yang dilaporkan. Saat ini, satu-satunya nilai yang valid untuk metrik penggunaan Amazon S3 adalah. `Resource`
`Resource`	Tipe sumber daya yang sedang berjalan. Saat ini, satu-satu nya nilai yang valid untuk metrik penggunaan Amazon S3 adalah`GeneralPurposeBuckets` , yang mengembalikan jumlah bucket tujuan umum dalam file. Akun AWS Bucket tujuan umum memungkinkan objek yang disimpan di semua kelas penyimpanan, kecuali S3 Express One Zone.

Mengakses metrik CloudWatch

Anda dapat menggunakan prosedur berikut untuk melihat metrik penyimpanan untuk Amazon S3. Untuk melibatkan metrik Amazon S3, Anda harus mengatur stempel waktu mulai dan selesai. Untuk metrik untuk periode 24 jam tertentu, atur periode waktu menjadi 86400 detik, jumlah detik dalam sehari. Selain itu, ingatlah untuk mengatur dimensi `BucketName` dan `StorageType`.

Menggunakan AWS CLI

Misalnya, jika Anda ingin menggunakan file AWS CLI untuk mendapatkan rata-rata ukuran bucket tertentu dalam byte, Anda dapat menggunakan perintah berikut:

```
aws cloudwatch get-metric-statistics --metric-name BucketSizeBytes --namespace AWS/S3
 --start-time 2016-10-19T00:00:00Z --end-time 2016-10-20T00:00:00Z --statistics Average
 --unit Bytes --region us-west-2 --dimensions Name=BucketName,Value=amzn-s3-demo-bucket
 Name=StorageType,Value=StandardStorage --period 86400 --output json
```

Contoh ini menghasilkan hasil berikut.

```
{
    "Datapoints": [
        {
            "Timestamp": "2016-10-19T00:00:00Z",
            "Average": 1025328.0,
            "Unit": "Bytes"
        }
    ],
    "Label": "BucketSizeBytes"
}
```

Menggunakan konsol S3

Untuk melihat metrik menggunakan konsol Amazon CloudWatch

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Metrik.
3. Pilih namespace S3.
4. (Opsional) Untuk melihat metrik, masukkan nama metrik dalam kotak pencarian.
5. (Opsional) Untuk memfilter berdasarkan StorageType , masukkan nama kelas penyimpanan di kotak pencarian.

Untuk melihat daftar metrik valid yang disimpan untuk Anda Akun AWS dengan menggunakan AWS CLI

- Pada prompt perintah, gunakan perintah berikut.

```
aws cloudwatch list-metrics --namespace "AWS/S3"
```

Untuk informasi selengkapnya tentang izin yang diperlukan untuk mengakses CloudWatch dasbor, lihat Izin CloudWatch dasbor Amazon di Panduan Pengguna Amazon CloudWatch .

CloudWatch konfigurasi metrik

Dengan metrik CloudWatch permintaan Amazon untuk Amazon S3, Anda dapat menerima metrik CloudWatch 1 menit, CloudWatch menyetel alarm, dan CloudWatch mengakses dasbor untuk near-real-time melihat operasi dan kinerja penyimpanan Amazon S3 Anda. Untuk aplikasi yang bergantung pada penyimpanan cloud, metrik ini memungkinkan Anda dengan cepat mengidentifikasi dan menindaklanjuti masalah operasional. Saat diaktifkan, metrik 1 menit ini tersedia di tingkat bucket Amazon S3, secara default.

Jika ingin mendapatkan metrik CloudWatch permintaan untuk objek dalam bucket, Anda harus membuat konfigurasi metrik untuk bucket. Untuk informasi selengkapnya, lihat Membuat konfigurasi CloudWatch metrik untuk semua objek di bucket.

Anda juga dapat menggunakan prefiks bersama, tag objek, atau titik akses untuk menentukan filter untuk metrik yang dikumpulkan. Metode mendefinisikan filter ini memungkinkan Anda menyelaraskan filter metrik ke aplikasi bisnis, alur kerja, atau organisasi internal tertentu. Untuk informasi selengkapnya, lihat Membuat konfigurasi metrik yang melakukan filter berdasarkan prefiks, tag objek, atau titik akses. Untuk informasi lebih lanjut tentang metrik CloudWatch yang tersedia dan perbedaan antara metrik penyimpanan dan permintaan, lihat Memantau metrik dengan Amazon CloudWatch.

Ingat hal berikut ini saat menggunakan konfigurasi metrik:

- Anda dapat memiliki maksimum 1.000 konfigurasi metrik per bucket.
- Anda dapat memilih objek mana di dalam bucket untuk disertakan dalam konfigurasi metrik dengan menggunakan filter. Anda dapat memfilter pada prefiks bersama, tag objek, atau titik akses untuk menyelaraskan filter metrik ke aplikasi bisnis, alur kerja, atau organisasi internal tertentu. Untuk meminta metrik untuk seluruh bucket, buat konfigurasi metrik tanpa filter.

- Konfigurasi metrik hanya diperlukan untuk mengaktifkan metrik permintaan. Metrik penyimpanan harian tingkat bucket selalu diaktifkan, dan disediakan tanpa biaya tambahan. Saat ini, tidak mungkin mendapatkan metrik penyimpanan harian untuk subset objek yang difilter.
- Setiap konfigurasi metrik memungkinkan serangkaian penuh metrik permintaan yang tersedia. Metrik spesifik operasi (seperti `PostRequests`) dilaporkan hanya jika ada permintaan jenis tersebut untuk bucket atau filter Anda.
- Metrik permintaan dilaporkan untuk operasi tingkat objek. Juga dilaporkan untuk operasi yang membuat daftar konten bucket, seperti GET Bucket (List Objects), GET Bucket Object Versions, dan List Multipart Uploads, tetapi tidak dilaporkan untuk operasi lain pada bucket.
- Minta pemfilteran yang mendukung metrik berdasarkan prefiks, tag objek, atau titik akses, tetapi metrik penyimpanan tidak demikian.

Pengiriman CloudWatch metrik upaya terbaik

CloudWatch metrik disampaikan atas dasar upaya terbaik. Sebagian besar permintaan untuk objek Amazon S3 yang memiliki metrik permintaan menghasilkan titik data yang dikirim. CloudWatch

Kelengkapan dan ketepatan waktu metrik tidak dijamin. Titik data untuk permintaan tertentu mungkin dikembalikan dengan stempel waktu yang lebih lambat daripada ketika permintaan tersebut sebenarnya diproses. Titik data selama satu menit mungkin tertunda sebelum tersedia CloudWatch, atau mungkin tidak dikirimkan sama sekali. CloudWatch metrik permintaan memberi Anda gambaran tentang sifat lalu lintas terhadap bucket Anda dalam waktu hampir nyata. Ini tidak dimaksudkan untuk menjadi pencatatan akuntasi yang lengkap atas semua permintaan.

Berdasarkan sifat upaya terbaik dari fitur ini, laporan yang tersedia di Dasbor Manajemen Penagihan & Biaya mungkin menyertakan satu atau lebih permintaan akses yang tidak muncul dalam metrik bucket.

Untuk informasi selengkapnya tentang bekerja dengan CloudWatch metrik di Amazon S3, lihat topik berikut.

Topik

- Membuat konfigurasi CloudWatch metrik untuk semua objek di bucket
- Membuat konfigurasi metrik yang melakukan filter berdasarkan prefiks, tag objek, atau titik akses
- Menghapus filter metrik

Membuat konfigurasi CloudWatch metrik untuk semua objek di bucket

Saat mengonfigurasi metrik permintaan, Anda dapat membuat konfigurasi CloudWatch metrik untuk semua objek di bucket, atau memfilter berdasarkan awalan, tag objek, atau titik akses. Prosedur dalam topik ini menunjukkan kepada Anda cara membuat konfigurasi untuk semua objek dalam bucket Anda. Untuk membuat konfigurasi yang melakukan filter berdasarkan tag objek, prefiks, atau titik akses, lihat Membuat konfigurasi metrik yang melakukan filter berdasarkan prefiks, tag objek, atau titik akses.

Ada tiga jenis metrik Amazon untuk Amazon S3: CloudWatch metrik penyimpanan, metrik permintaan, dan metrik replikasi. Metrik penyimpanan dilaporkan satu kali sehari, dan diberikan kepada semua pelanggan tanpa biaya tambahan. Metrik permintaan tersedia dalam interval satu menit setelah beberapa latensi untuk diproses. Metrik permintaan ditagih dengan tarif standar CloudWatch . Anda harus memilih untuk meminta metrik dengan mengonfigurasinya di dalam konsol, atau menggunakan API Amazon S3. Metrik Replikasi S3 memberikan metrik terperinci untuk aturan replikasi dalam konfigurasi replikasi Anda. Dengan metrik replikasi, Anda dapat memantau minute-by-minute kemajuan dengan melacak byte yang tertunda, operasi tertunda, operasi yang gagal replikasi, dan latensi replikasi.

Untuk informasi selengkapnya tentang CloudWatch metrik untuk Amazon S3, lihat. Memantau metrik dengan Amazon CloudWatch

Anda dapat menambahkan konfigurasi metrik ke bucket melalui konsol Amazon S3, dengan AWS Command Line Interface (AWS CLI), atau dengan API REST Amazon S3.

Menggunakan konsol S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Dalam daftar bucket, pilih nama bucket yang berisi objek yang ingin Anda minta metrik.
4. Pilih tab Metrik.
5. Di bagian bawah Metrik bucket, pilih Lihat bagan tambahan.
6. Pilih tab Minta metrik.
7. Pilih Buat filter.
8. Di dalam kotak Nama filter, masukkan nama filter Anda.

Nama hanya dapat berisi huruf, angka, periode, tanda hubung, dan garis bawah. Kami sarankan menggunakan nama `EntireBucket` untuk filter yang berlaku di semua objek.

9. Di bagian bawah Cakupan filter, pilih Filter ini berlaku untuk semua objek dalam bucket.

   Anda juga dapat menentukan filter sehingga metrik hanya dikumpulkan dan dilaporkan pada subset objek dalam bucket. Untuk informasi selengkapnya, lihat Membuat konfigurasi metrik yang melakukan filter berdasarkan prefiks, tag objek, atau titik akses.

10. Pilih Simpan perubahan.

11. Pada tab Minta metrik, di bagian bawah Filter, pilih filter yang baru saja Anda buat.

    Setelah sekitar 15 menit, CloudWatch mulai melacak metrik permintaan ini. Anda dapat melihatnya di tab Minta metrik. Anda dapat melihat grafik untuk metrik di Amazon CloudWatch S3 atau konsol. Metrik permintaan ditagih dengan tarif standar CloudWatch . Untuk informasi selengkapnya, lihat CloudWatch harga Amazon.

Penggunaan API REST

Anda juga dapat menambahkan konfigurasi metrik secara terprogram dengan API REST Amazon S3. Untuk informasi selengkapnya tentang menambahkan dan bekerja dengan konfigurasi metrik, lihat topik berikut di Referensi API Amazon Simple Storage Service:

- Konfigurasi Metrik Bucket PUT
- Konfigurasi Metrik Bucket GET
- Daftar Konfigurasi Metrik Bucket
- Konfigurasi Metrik Bucket DELETE

Menggunakan AWS CLI

1. Instal dan atur AWS CLI. Untuk instruksi, lihat Menginstal, memperbarui, dan menghapus AWS CLI di AWS Command Line Interface Panduan Pengguna.
2. Buka terminal.
3. Jalankan perintah berikut untuk menambahkan konfigurasi metrik.

```
aws s3api put-bucket-metrics-configuration --endpoint https://s3.us-
west-2.amazonaws.com --bucket bucket-name --id metrics-config-id --metrics-
configuration '{"Id":"metrics-config-id"}'
```

Membuat konfigurasi metrik yang melakukan filter berdasarkan prefiks, tag objek, atau titik akses

Ada tiga jenis metrik Amazon untuk Amazon S3: CloudWatch metrik penyimpanan, metrik permintaan, dan metrik replikasi. Metrik penyimpanan dilaporkan satu kali sehari, dan diberikan kepada semua pelanggan tanpa biaya tambahan. Metrik permintaan tersedia dalam interval satu menit setelah beberapa latensi untuk diproses. Metrik permintaan ditagih dengan tarif standar CloudWatch . Anda harus memilih untuk meminta metrik dengan mengonfigurasinya di dalam konsol, atau menggunakan API Amazon S3. Metrik Replikasi S3 memberikan metrik terperinci untuk aturan replikasi dalam konfigurasi replikasi Anda. Dengan metrik replikasi, Anda dapat memantau minute-by-minute kemajuan dengan melacak byte yang tertunda, operasi tertunda, operasi yang gagal replikasi, dan latensi replikasi.

Untuk informasi selengkapnya tentang CloudWatch metrik untuk Amazon S3, lihat. Memantau metrik dengan Amazon CloudWatch

Saat mengonfigurasi CloudWatch metrik, Anda dapat membuat filter untuk semua objek di bucket, atau memfilter konfigurasi ke dalam grup objek terkait dalam satu bucket. Anda dapat memfilter objek dalam bucket untuk disertakan dalam konfigurasi metrik berdasarkan satu atau beberapa jenis filter berikut ini:

- Prefiks nama kunci objek–Meskipun model data Amazon S3 adalah struktur datar, Anda dapat menyimpulkan hierarki dengan menggunakan prefiks. Konsol Amazon S3 mendukung prefiks ini dengan konsep folder. Jika Anda memfilter menurut prefiks, objek yang memiliki prefiks yang sama akan disertakan dalam konfigurasi metrik. Untuk informasi selengkapnya tentang prefiks, lihat Organisasi objek menggunakan prefiks.
- Tag–Anda dapat menambahkan tag, yang merupakan pasangan nama nilai-kunci, untuk objek. Tag membantu Anda untuk menemukan dan menata objek dengan mudah. Anda juga dapat menggunakan tag sebagai filter untuk konfigurasi metrik. Untuk informasi selengkapnya tentang tag objek, lihat Mengategorikan penyimpanan Anda menggunakan tag.
- Titik akses–Titik Akses S3 diberi nama titik akhir jaringan yang dilampirkan ke bucket dan menyederhanakan pengelolaan akses data pada skala untuk set data bersama di S3. Saat Anda

membuat filter titik akses, Amazon S3 menyertakan permintaan ke titik akses yang Anda tentukan dalam konfigurasi metrik. Untuk informasi selengkapnya, lihat Pemantauan dan pencatatan titik akses.

> **Note**
>
> Saat Anda membuat konfigurasi metrik yang memfilter berdasarkan titik akses, Anda harus menggunakan titik akses Amazon Resource Name (ARN), bukan alias titik akses. Pastikan Anda menggunakan ARN untuk titik akses itu sendiri, bukan ARN untuk objek tertentu. Untuk informasi selengkapnya tentang titik akses ARNs, lihatMenggunakan jalur akses Amazon S3 untuk bucket tujuan umum.

Jika Anda menetapkan filter, hanya permintaan yang beroperasi pada objek tunggal yang dapat cocok dengan filter dan disertakan dalam metrik yang dilaporkan. Permintaan suka DeleteObjectsdan `ListObjects` permintaan tidak menampilkan metrik apa pun untuk konfigurasi dengan filter.

Untuk meminta pemfilteran yang lebih kompleks, pilih dua atau lebih elemen. Hanya objek yang memiliki semua elemen tersebut yang disertakan dalam konfigurasi metrik. Jika Anda tidak mengatur filter, semua objek dalam bucket disertakan dalam konfigurasi metrik.

Menggunakan konsol S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum
3. Dalam daftar bucket, pilih nama bucket yang berisi objek yang ingin Anda minta metrik.
4. Pilih tab Metrik.
5. Di bagian bawah Metrik bucket, pilih Lihat bagan tambahan.
6. Pilih tab Minta metrik.
7. Pilih Buat filter.
8. Di dalam kotak Nama filter, masukkan nama filter Anda.

   Nama hanya dapat berisi huruf, angka, periode, tanda hubung, dan garis bawah.
9. Di bawah Cakupan filter, pilih Batasi cakupan filter ini menggunakan prefiks, tag objek, dan Titik Akses S3, atau kombinasi ketiganya.

10. Di bawah Jenis filter, pilih setidaknya satu jenis filter: Awalan, Tag objek, atau Titik akses.
11. Untuk menentukan filter prefiks dan membatasi cakupan filter ke satu jalur, di kotak Prefiks, masukkan prefiks.
12. Untuk menentukan filter tag objek, di bawah Tag objek, pilih Tambah tag, lalu masukkan tag Kunci dan Nilai.
13. Untuk menentukan filter titik akses, di bidang Titik Akses S3, masukkan titik akses ARN, atau pilih Telusuri S3 untuk menavigasi ke titik akses.

    > **Important**
    >
    > Anda tidak dapat memasukkan alias titik akses. Anda harus memasukkan ARN untuk titik akses itu sendiri, bukan ARN untuk objek tertentu.

14. Pilih Simpan perubahan.

    Amazon S3 membuat filter yang menggunakan prefiks, tag, atau titik akses yang Anda tentukan.
15. Pada tab Minta metrik, di bagian bawah Filter, pilih filter yang baru saja Anda buat.

    Sekarang Anda telah membuat filter yang membatasi cakupan metrik permintaan berdasarkan prefiks, tag objek, atau titik akses. Sekitar 15 menit setelah CloudWatch mulai melacak metrik permintaan ini, Anda dapat melihat bagan untuk metrik di Amazon CloudWatch S3 dan konsol. Metrik permintaan ditagih dengan tarif standar CloudWatch . Untuk informasi selengkapnya, lihat CloudWatch harga Amazon.

    Anda juga dapat mengonfigurasi metrik permintaan di tingkat bucket. Untuk informasi, lihat Membuat konfigurasi CloudWatch metrik untuk semua objek di bucket.

Menggunakan AWS CLI

1. Instal dan atur AWS CLI. Untuk instruksi, lihat Menginstal, memperbarui, dan menghapus AWS CLI di AWS Command Line Interface Panduan Pengguna.
2. Buka terminal.
3. Untuk menambahkan konfigurasi metrik, jalankan salah satu dari perintah berikut ini:

Example : Untuk memfilter berdasarkan prefiks

```
aws s3api put-bucket-metrics-configuration --bucket amzn-s3-demo-bucket --
id metrics-config-id --metrics-configuration '{"Id":"metrics-config-id", "Filter":
{"Prefix":"prefix1"}} '
```

Example : Untuk memfilter berdasarkan tag

```
aws s3api put-bucket-metrics-configuration --bucket amzn-s3-demo-bucket --
id metrics-config-id --metrics-configuration '{"Id":"metrics-config-id", "Filter":
{"Tag": {"Key": "string", "Value": "string"}} '
```

Example : Untuk memfilter berdasarkan titik akses

```
aws s3api put-bucket-metrics-configuration --bucket amzn-s3-demo-bucket --
id metrics-config-id --metrics-configuration '{"Id":"metrics-config-id", "Filter":
{"AccessPointArn":"arn:aws:s3:Region:account-id:accesspoint/access-point-name"}} '
```

Example : Untuk memfilter berdasarkan prefiks, tag, dan titik akses

```
aws s3api put-bucket-metrics-configuration --endpoint https://
s3.Region.amazonaws.com --bucket amzn-s3-demo-bucket --id metrics-config-id --
metrics-configuration '
{
    "Id": "metrics-config-id",
    "Filter": {
        "And": {
            "Prefix": "string",
            "Tags": [
                {
                    "Key": "string",
                    "Value": "string"
                }
            ],
            "AccessPointArn": "arn:aws:s3:Region:account-id:accesspoint/access-
point-name"
        }
    }
}'
```

Penggunaan API REST

Anda juga dapat menambahkan konfigurasi metrik secara terprogram dengan API REST Amazon S3. Untuk informasi selengkapnya tentang menambahkan dan bekerja dengan konfigurasi metrik, lihat topik berikut di Referensi API Amazon Simple Storage Service:

- Konfigurasi Metrik Bucket PUT
- Konfigurasi Metrik Bucket GET
- Daftar Konfigurasi Metrik Bucket
- Konfigurasi Metrik Bucket DELETE

Menghapus filter metrik

Anda dapat menghapus filter metrik CloudWatch permintaan Amazon jika Anda tidak lagi membutuhkannya. Saat menghapus filter, Anda tidak lagi dikenakan biaya untuk metrik permintaan yang menggunakan filter tersebut filter khusus. Namun, Anda akan terus dibebankan untuk konfigurasi filter lainnya yang ada.

Saat menghapus filter, Anda tidak lagi dapat menggunakan filter untuk metrik permintaan. Tindakan menghapus filter tidak dapat diurungkan.

Untuk informasi selengkapnya tentang membuat filter metrik permintaan, lihat topik-topik berikut:

- Membuat konfigurasi CloudWatch metrik untuk semua objek di bucket
- Membuat konfigurasi metrik yang melakukan filter berdasarkan prefiks, tag objek, atau titik akses

Menggunakan konsol S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Dalam daftar bucket, pilih nama bucket yang ingin Anda hapus filter metrik permintaan.
4. Pilih tab Metrik.
5. Di bagian bawah Metrik bucket, pilih Lihat bagan tambahan.
6. Pilih tab Minta metrik.
7. Pilih Kelola filter.

8. Pilih filter Anda.

> **Important**
>
> Tindakan menghapus filter tidak dapat diurungkan.

9. Pilih Hapus.

   Amazon S3 menghapus filter Anda.

Penggunaan API REST

Anda juga dapat menambahkan konfigurasi metrik secara terprogram dengan API REST Amazon S3. Untuk informasi selengkapnya tentang menambahkan dan bekerja dengan konfigurasi metrik, lihat topik berikut di Referensi API Amazon Simple Storage Service:

- Konfigurasi Metrik Bucket PUT
- Konfigurasi Metrik Bucket GET
- Daftar Konfigurasi Metrik Bucket
- Konfigurasi Metrik Bucket DELETE

Notifikasi Peristiwa Amazon S3

Anda dapat menggunakan fitur Notifikasi Peristiwa Amazon S3 untuk menerima pemberitahuan ketika peristiwa tertentu terjadi di bucket S3 Anda. Untuk mengaktifkan notifikasi, tambahkan konfigurasi notifikasi yang mengidentifikasi peristiwa yang ingin dipublikasikan Amazon S3. Pastikan itu juga mengidentifikasi tujuan tempat Anda ingin Amazon S3 mengirim notifikasi. Anda menyimpan konfigurasi ini di sub-sumber daya notifikasi yang terkait dengan bucket. Untuk informasi selengkapnya, lihat opsi konfigurasi bucket tujuan umum. Amazon S3 menyediakan API bagi Anda untuk dapat mengelola sub-sumber daya ini.

> **Important**
>
> Pemberitahuan peristiwa Amazon S3 dirancang untuk dikirimkan setidaknya satu kali. Biasanya, pemberitahuan peristiwa dikirimkan dalam hitungan detik, tetapi terkadang perlu waktu satu menit atau lebih.

Gambaran Umum Notifikasi Peristiwa Amazon S3

Saat ini, Amazon S3 dapat menerbitkan pemberitahuan untuk peristiwa berikut:

- Objek baru yang dibuat peristiwa
- Peristiwa penghapusan objek
- Kembalikan peristiwa objek
- Mengurangi objek Penyimpanan Redundansi (RRS) yang hilang
- Peristiwa replikasi
- Acara kedaluwarsa Siklus Hidup S3
- Peristiwa transisi Siklus Hidup S3
- S3 Intelligent-Tiering peristiwa arsip otomatis
- Peristiwa pemberian tag objek
- Peristiwa objek ACL PUT

Untuk deskripsi lengkap semua jenis peristiwa yang didukung, lihat Jenis event yang didukung untuk SQS, SNS, dan Lambda.

Amazon S3 dapat mengirimkan pesan pemberitahuan peristiwa ke tujuan berikut. Anda menentukan nilai Amazon Resource Name (ARN) dari tujuan ini dalam konfigurasi notifikasi.

- Topik Amazon Simple Notification Service (Amazon SNS)
- Antrean Amazon Simple Queue Service (Amazon SQS)
- AWS Lambda fungsi
- Amazon EventBridge

Untuk informasi selengkapnya, lihat Tujuan peristiwa yang didukung.

> **Note**
>
> Antrean Amazon Simple Queue Service FIFO (First-In-First-Out) tidak didukung sebagai tujuan pemberitahuan peristiwa Amazon S3. Untuk mengirim pemberitahuan untuk acara Amazon S3 ke antrian FIFO Amazon SQS, Anda dapat menggunakan Amazon. EventBridge Untuk informasi selengkapnya, lihat Mengaktifkan Amazon EventBridge.

> **Warning**
>
> Jika notifikasi Anda akhirnya menulis ke bucket yang sama yang memicu notifikasi, ini dapat menyebabkan loop eksekusi. Misalnya, jika bucket memicu fungsi Lambda setiap kali suatu objek diunggah, dan fungsinya mengunggah sebuah objek ke bucket, maka fungsi tersebut secara tidak langsung memicu fungsi itu sendiri. Untuk menghindari hal ini, gunakan dua bucket, atau konfigurasikan pemicu agar hanya berlaku pada prefiks yang digunakan untuk objek masuk.
>
> Untuk informasi selengkapnya dan contoh penggunaan notifikasi Amazon S3 dengan AWS Lambda, lihat Menggunakan AWS Lambda dengan Amazon S3 di AWS Lambda Panduan Pengembang.

Untuk informasi selengkapnya tentang jumlah konfigurasi notifikasi peristiwa yang dapat Anda buat per bucket, lihat Kuota layanan Amazon S3 dalam AWS Referensi Umum.

Untuk informasi selengkapnya tentang notifikasi Peristiwa, lihat bagian berikut.

Topik

- Jenis dan tujuan pemberitahuan peristiwa
- Menggunakan Amazon SQS, Amazon SNS, dan Lambda
- Menggunakan EventBridge

Jenis dan tujuan pemberitahuan peristiwa

Amazon S3 mendukung beberapa jenis tujuan notifikasi peristiwa tempat notifikasi dapat diterbitkan. Anda dapat menentukan jenis peristiwa dan tujuan ketika mengonfigurasi notifikasi peristiwa Anda. Hanya satu tujuan yang dapat ditentukan untuk setiap pemberitahuan peristiwa. Pemberitahuan peristiwa Amazon S3 mengirim satu entri peristiwa untuk setiap pesan notifikasi.

Topik

- Tujuan peristiwa yang didukung
- Jenis event yang didukung untuk SQS, SNS, dan Lambda
- Jenis acara yang didukung untuk Amazon EventBridge
- Pemesanan acara dan duplikat acara

Tujuan peristiwa yang didukung

Amazon S3 dapat mengirimkan pesan pemberitahuan peristiwa ke tujuan berikut.

- Topik Amazon Simple Notification Service (Amazon SNS)
- Antrean Amazon Simple Queue Service (Amazon SQS)
- AWS Lambda
- Amazon EventBridge

Namun, hanya satu jenis tujuan yang dapat ditentukan untuk setiap pemberitahuan peristiwa.

> **Note**
>
> Anda harus memberikan izin Amazon S3 untuk mem-posting pesan ke topik Amazon SNS atau antrean Amazon SQS. Anda juga harus memberikan izin Amazon S3 untuk menjalankan AWS Lambda fungsi atas nama Anda. Untuk petunjuk tentang cara memberikan izin ini, lihat memberikan izin untuk memublikasikan pesan pemberitahuan peristiwa ke tujuan.

Topik Amazon SNS

Amazon SNS adalah layanan perpesanan push yang fleksibel, dan dikelola penuh. Anda dapat menggunakan layanan ini untuk mendorong pesan ke perangkat seluler atau layanan terdistribusi. Dengan SNS, Anda dapat memublikasikan pesan satu kali, dan mengirimkannya satu atau beberapa kali. Saat ini SNS Standar hanya diizinkan sebagai tujuan notifikasi peristiwa S3, sedangkan SNS FIFO tidak diizinkan.

Amazon SNS mengoordinasikan dan mengelola pengiriman dan pengiriman pesan ke titik akhir atau klien yang berlangganan. Anda dapat menggunakan konsol Amazon SNS untuk membuat topik Amazon SNS yang dapat dikirimkan pemberitahuan Anda.

Topiknya harus Wilayah AWS sama dengan bucket Amazon S3 Anda. Untuk petunjuk tentang cara membuat topik Amazon SNS, lihat Memulai dengan Amazon SNS di Panduan Pengembang Amazon Simple Notification Service dan FAQ Amazon SNS.

Sebelum Anda dapat menggunakan topik Amazon SNS yang Anda buat sebagai tujuan pemberitahuan peristiwa, Anda memerlukan hal berikut:

- Amazon Resource Name (ARN) untuk topik Amazon SNS

- Berlangganan topik Amazon SNS yang valid. Dengan itu, pelanggan topik diberi tahu saat pesan dipublikasikan ke topik Amazon SNS Anda.

Antrean Amazon SQS

Amazon SQS menawarkan antrean hostingan yang andal dan dapat diskalakan untuk menyimpan pesan saat mereka bepergian antar komputer. Anda dapat menggunakan Amazon SQS untuk mengirimkan volume data apa pun tanpa memerlukan layanan lain untuk selalu tersedia. Anda dapat menggunakan konsol Amazon SQS untuk membuat antrean Amazon SQS yang dapat dikirimkan pemberitahuan Anda.

Antrian Amazon SQS harus sama Wilayah AWS dengan bucket Amazon S3 Anda. Untuk petunjuk tentang cara membuat antrean Amazon SQS, lihat Apa itu Amazon Simple Queue Service dan Memulai dengan Amazon SQS di Panduan Pengembang Amazon Simple Storage Service.

Sebelum Anda dapat menggunakan antrean Amazon SQS sebagai tujuan pemberitahuan peristiwa, Anda memerlukan berikut ini:

- Amazon Resource Name (ARN) untuk antrean Amazon SQS

> **Note**
>
> Antrean Amazon Simple Queue Service FIFO (First-In-First-Out) tidak didukung sebagai tujuan pemberitahuan peristiwa Amazon S3. Untuk mengirim pemberitahuan untuk acara Amazon S3 ke antrian FIFO Amazon SQS, Anda dapat menggunakan Amazon. EventBridge Untuk informasi selengkapnya, lihat Mengaktifkan Amazon EventBridge.

Fungsi Lambda

Anda dapat menggunakan AWS Lambda untuk memperluas AWS layanan lain dengan logika khusus, atau membuat backend Anda sendiri yang beroperasi pada AWS skala, kinerja, dan keamanan. Dengan Lambda, Anda dapat membuat aplikasi terpisah berbasis peristiwa yang berjalan hanya saat diperlukan. Anda juga dapat menggunakannya untuk menskalakan aplikasi ini secara otomatis dari beberapa permintaan sehari hingga ribuan detik.

Lambda dapat menjalankan kode kustom sebagai respons terhadap peristiwa bucket Amazon S3. Anda unggah kode kustom Anda ke Lambda dan membuat apa yang disebut fungsi Lambda. Saat

Amazon S3 mendeteksi peristiwa dari jenis tertentu, Amazon S3 dapat memublikasikan acara ke AWS Lambda dan memanggil fungsi Anda di Lambda. Sebagai respons, Lambda menjalankan fungsi Anda. Salah satu jenis peristiwa yang mungkin dideteksi, misalnya, adalah peristiwa yang dibuat objek.

Anda dapat menggunakan AWS Lambda konsol untuk membuat fungsi Lambda yang menggunakan AWS infrastruktur untuk menjalankan kode atas nama Anda. Fungsi Lambda harus berada di Wilayah yang sama dengan bucket S3. Anda juga harus memiliki nama atau ARN dari fungsi Lambda untuk mengatur fungsi Lambda sebagai tujuan notifikasi peristiwa.

> **Warning**
>
> Jika notifikasi Anda akhirnya menulis ke bucket yang sama yang memicu notifikasi, ini dapat menyebabkan loop eksekusi. Misalnya, jika bucket memicu fungsi Lambda setiap kali suatu objek diunggah, dan fungsinya mengunggah sebuah objek ke bucket, maka fungsi tersebut secara tidak langsung memicu fungsi itu sendiri. Untuk menghindari hal ini, gunakan dua bucket, atau konfigurasikan pemicu agar hanya berlaku pada prefiks yang digunakan untuk objek masuk.
>
> Untuk informasi selengkapnya dan contoh penggunaan notifikasi Amazon S3 dengan AWS Lambda, lihat Menggunakan AWS Lambda dengan Amazon S3 di AWS Lambda Panduan Pengembang.

Amazon EventBridge

Amazon EventBridge adalah bus acara tanpa server, yang menerima acara dari AWS layanan. Anda dapat mengatur aturan untuk mencocokkan peristiwa dan mengirimkannya ke target, seperti layanan AWS atau titik akhir HTTP. Untuk informasi selengkapnya, lihat Apa yang ada EventBridge di Panduan EventBridge Pengguna Amazon.

Tidak seperti tujuan lain, Anda dapat mengaktifkan atau menonaktifkan acara yang akan dikirimkan EventBridge untuk ember. Jika Anda mengaktifkan pengiriman, semua acara dikirim ke EventBridge. Selain itu, Anda dapat menggunakan EventBridge aturan untuk merutekan acara ke target tambahan.

Jenis event yang didukung untuk SQS, SNS, dan Lambda

Amazon S3 dapat menerbitkan peristiwa dari jenis berikut. Anda menentukan jenis peristiwa ini di konfigurasi pemberitahuan.

Jenis peristiwa	Deskripsi
`s3:TestEvent`	Ketika notifikasi diaktifkan, Amazon S3 menerbitkan notifikasi tes. Hal ini untuk memastikan bahwa topik ada dan bahwa pemilik bucket memiliki izin untuk mempublik asikan topik tertentu. Jika mengaktifkan notifikasi gagal, Anda tidak menerima notifikasi tes.

`s3:ObjectCreated:*`
 	
`s3:ObjectCreated:Put`
 	
`s3:ObjectCreated:Post`
 	
`s3:ObjectCreated:Copy`
 	
`s3:ObjectCreated:C ompleteMultipartUpload`
	Operasi API Amazon S3 sepertiPUT,POST, dan COPY dapat membuat objek. Dengan jenis peristiwa ini, Anda dapat mengaktifkan notifikasi saat objek dibuat menggunakan operasi API tertentu. Atau, Anda dapat menggunakan jenis peristiwa `s3:ObjectCreated:*` untuk meminta notifikasi terlepas dari API yang digunakan untuk membuat objek.
	
	`s3:ObjectCreated:CompleteMultipartUp load` termasuk objek yang dibuat menggunakan UploadPartCopyuntuk operasi Salin.

`s3:ObjectRemoved:*`
 	
`s3:ObjectRemoved:Delete`
 	
`s3:ObjectRemoved:D eleteMarkerCreated`
	Dengan menggunakan tipe peristiwa `ObjectRemoved` , Anda dapat mengaktifkan pemberitahuan saat objek atau batch objek dihapus dari bucket.
	
	Anda dapat meminta pemberitahuan saat objek dihapus atau objek versi dihapus secara permanen menggunakan jenis peristiwa `s3:ObjectRemoved:Delete` . Atau, Anda dapat meminta notifikasi saat penanda penghapus an dibuat untuk objek versi menggunakan `s3:Object Removed:DeleteMarkerCreated` . Untuk petunjuk cara menghapus objek berversi, lihat Menghapus versi objek dari bucket dengan dukungan Penentuan Versi. Anda juga dapat menggunakan wildcard `s3:Object Removed:*` untuk meminta pemberitahuan kapan pun sebuah objek dihapus.

Jenis peristiwa	Deskripsi
	Pemberitahuan peristiwa ini tidak memperingatkan Anda untuk penghapusan otomatis dari konfigurasi siklus aktif atau dari operasi yang gagal.

`s3:ObjectRestore:*`
 	
`s3:ObjectRestore:Post`
 	
`s3:ObjectRestore:Completed`
 	
`s3:ObjectRestore:Delete`
	Dengan menggunakan jenis `ObjectRestore` acara, Anda dapat menerima pemberitahuan untuk inisiasi dan penyelesaian acara saat memulihkan objek dari kelas penyimpanan Pengambilan Fleksibel Gletser S3, kelas penyimpanan S3 Glacier Deep Archive, tingkat Akses Arsip Tingkat Cerdas S3, dan tingkat Akses Arsip Tingkat Cerdas S3. Anda juga dapat menerima pemberitahuan ketika salinan objek yang dipulihkan kedaluwarsa.
	
	Jenis peristiwa `s3:ObjectRestore:Post` memberi tahu Anda tentang inisiasi restorasi objek. Jenis peristiwa `s3:ObjectRestore:Completed` memberi tahu Anda tentang penyelesaian restorasi. Jenis peristiwa `s3:ObjectRestore:Delete` memberi tahu Anda saat salinan sementara objek yang dipulihkan kedaluwarsa.

`s3:ReducedRedundancyLostObject`	Anda menerima peristiwa notifikasi ini saat Amazon S3 mendeteksi bahwa objek kelas penyimpanan RRS hilang.

Jenis peristiwa	Deskripsi

`s3:Replication:*`
 	
`s3:Replication:OperationFailedReplication`
 	
`s3:Replication:OperationMissedThreshold`
 	
`s3:Replication:OperationReplicatedAfterThreshold`
 	
`s3:Replication:OperationNotTracked`
	Dengan menggunakan jenis `Replication` acara, Anda dapat menerima pemberitahuan untuk konfigurasi replikasi yang mengaktifkan metrik Replikasi S3 atau Kontrol Waktu Replikasi S3 (S3 RTC). Anda dapat memantau minute-by-minute kemajuan peristiwa replikasi dengan melacak byte yang tertunda, operasi tertunda, dan latensi replikasi. Untuk informasi tentang metrik replikasi, lihat Memantau replikasi dengan metrik, pemberitahuan acara, dan status.
	
	• Jenis peristiwa `s3:Replication:OperationFailedReplication` memberi tahu Anda saat objek yang memenuhi syarat untuk replikasi gagal mereplikasi.
	• Jenis `s3:Replication:OperationMissedThreshold` acara memberi tahu Anda ketika objek yang memenuhi syarat untuk replikasi yang menggunakan S3 RTC melebihi ambang batas 15 menit untuk replikasi.
	• Jenis `s3:Replication:OperationReplicatedAfterThreshold` acara memberi tahu Anda saat objek yang memenuhi syarat untuk replikasi yang menggunakan S3 RTC bereplikasi setelah ambang batas 15 menit.
	• Jenis `s3:Replication:OperationNotTracked` acara memberi tahu Anda saat objek yang memenuhi syarat untuk replikasi langsung (baik Same-Region Replication [SRR] atau Cross-Region Replication [CRR]) tidak lagi dilacak oleh metrik replikasi.

Jenis peristiwa	Deskripsi

`s3:LifecycleExpiration:*`
 	
`s3:LifecycleExpiration:Delete`
 	
`s3:LifecycleExpiration:DeleteMarkerCreated`
	Dengan menggunakan jenis `LifecycleExpiration` acara, Anda dapat menerima pemberitahuan saat Amazon S3 menghapus objek berdasarkan konfigurasi Siklus Hidup S3 Anda.
	
	Jenis peristiwa `s3:LifecycleExpiration:Delete` akan memberi tahu Anda saat objek dalam bucket yang tidak berversi dihapus. Ini juga memberi tahu Anda ketika versi objek dihapus secara permanen oleh konfigurasi Siklus Hidup S3. Jenis peristiwa `s3:LifecycleExpiration:DeleteMarkerCreated` akan memberi tahu Anda saat Siklus Hidup S3 membuat penanda hapus saat versi objek saat ini dalam bucket berversi dihapus.

`s3:LifecycleTransition`	Anda menerima peristiwa notifikasi ini saat objek dialihkan ke kelas penyimpanan Amazon S3 lainnya dengan konfigurasi Siklus Hidup S3.
`s3:IntelligentTiering`	Anda menerima peristiwa notifikasi ini ketika objek dalam kelas penyimpanan S3 Intelligent-Tiering pindah ke tingkat Archive Access atau tingkat Deep Archive Access.

`s3:ObjectTagging:*`
 	
`s3:ObjectTagging:Put`
 	
`s3:ObjectTagging:Delete`
	Dengan menggunakan jenis `ObjectTagging` acara, Anda dapat mengaktifkan notifikasi saat tag objek ditambahkan atau dihapus dari objek.
	
	Jenis peristiwa `s3:ObjectTagging:Put` memberi tahu Anda saat tag PUT pada objek atau tag yang ada diperbarui. Jenis peristiwa `s3:ObjectTagging:Delete` memberi tahu Anda saat tag dihapus dari objek.

`s3:ObjectAcl:Put`	Anda menerima peristiwa pemberitahuan ini ketika ACL PUT pada objek atau ketika ACL yang ada diubah. Peristiwa tidak dihasilkan ketika permintaannya tidak menghasilkan perubahan pada ACL objek.

Jenis acara yang didukung untuk Amazon EventBridge

Untuk daftar jenis acara yang akan dikirimkan Amazon S3 ke Amazon EventBridge, lihat. Menggunakan EventBridge

Pemesanan acara dan duplikat acara

Pemberitahuan Acara Amazon S3 dirancang untuk mengirimkan pemberitahuan setidaknya sekali, tetapi pemberitahuan tersebut tidak dijamin tiba dalam urutan yang sama dengan peristiwa yang terjadi. Pada kesempatan yang jarang terjadi, mekanisme coba ulang Amazon S3 dapat menyebabkan duplikat Pemberitahuan Acara S3 untuk peristiwa objek yang sama. Untuk selengkapnya tentang penanganan kejadian duplikat atau di luar pesanan, lihat Mengelola pemesanan acara dan duplikat peristiwa dengan Pemberitahuan Acara Amazon S3 di AWS Blog Penyimpanan.

Menggunakan Amazon SQS, Amazon SNS, dan Lambda

Mengaktifkan notifikasi adalah operasi tingkat bucket. Anda menyimpan konfigurasi notifikasi di sub-sumber daya notifikasi yang berasosiasi dengan bucket. Setelah Anda membuat atau mengubah konfigurasi notifikasi bucket, biasanya diperlukan waktu sekitar lima menit sampai perubahan diterapkan. `s3:TestEvent` terjadi ketika notifikasi pertama kali diaktifkan. Anda dapat menggunakan salah satu metode berikut untuk mengelola konfigurasi pemberitahuan:

- Menggunakan konsol Amazon S3—Anda dapat menggunakan UI konsol untuk mengatur konfigurasi notifikasi pada bucket tanpa harus tulis kode apa pun. Untuk informasi selengkapnya, lihat Mengaktifkan dan mengonfigurasi notifikasi peristiwa menggunakan konsol Amazon S3.
- Secara terprogram menggunakan AWS SDKs — Secara internal, baik konsol maupun SDKs panggilan Amazon S3 REST API untuk mengelola subresource notifikasi yang terkait dengan bucket. Untuk contoh konfigurasi notifikasi yang menggunakan SDK AWS , lihat Panduan: Mengonfigurasi bucket untuk notifikasi (topik SNS atau antrean SQS).

> **Note**
>
> Anda juga dapat melakukan panggilan API REST Amazon S3 secara langsung dari kode Anda. Namun, ini dapat merepotkan karena untuk melakukannya Anda harus menulis kode untuk mengautentikasi permintaan Anda.

Terlepas dari metode yang Anda gunakan, Amazon S3 menyimpan konfigurasi notifikasi sebagai XL di sub-sumber daya notifikasi yang terkait dengan bucket. Untuk informasi tentang sub-sumber daya bucket, lihat opsi konfigurasi bucket tujuan umum.

> **Note**
>
> Jika Anda memiliki beberapa pemberitahuan peristiwa gagal karena tujuan yang dihapus, Anda mungkin menerima Tidak dapat memvalidasi konfigurasi tujuan berikut saat mencoba menghapusnya. Anda dapat mengatasi ini di konsol S3 dengan menghapus semua notifikasi yang gagal secara bersamaan.

Topik

- memberikan izin untuk memublikasikan pesan pemberitahuan peristiwa ke tujuan
- Mengaktifkan dan mengonfigurasi notifikasi peristiwa menggunakan konsol Amazon S3
- Mengonfigurasi notifikasi peristiwa secara terprogram
- Panduan: Mengonfigurasi bucket untuk notifikasi (topik SNS atau antrean SQS)
- Mengonfigurasi notifikasi peristiwa dengan filter nama kunci objek
- Struktur pesan peristiwa

memberikan izin untuk memublikasikan pesan pemberitahuan peristiwa ke tujuan

Anda harus memberikan izin yang diperlukan kepada pengguna utama Amazon S3 untuk memanggil API terkait guna menerbitkan pesan ke topik SNS, antrean SQS, atau fungsi Lambda. Hal ini agar Amazon S3 dapat memublikasikan pesan pemberitahuan peristiwa ke tujuan.

Untuk memecahkan masalah memublikasikan pesan pemberitahuan peristiwa ke tujuan, lihat Memecahkan masalah untuk memublikasikan pemberitahuan peristiwa Amazon S3 ke topik Layanan Pemberitahuan Sederhana Amazon .

Topik

- Memberikan izin untuk menjalankan fungsi AWS Lambda
- memberikan izin untuk memublikasikan pesan ke topik SNS atau antrean SQS

Memberikan izin untuk menjalankan fungsi AWS Lambda

Amazon S3 menerbitkan pesan peristiwa AWS Lambda dengan menjalankan fungsi Lambda dan menyediakan pesan acara sebagai argumen.

Saat Anda menggunakan konsol Amazon S3 untuk mengonfigurasi notifikasi peristiwa pada bucket Amazon S3 untuk fungsi Lambda, konsol menyiapkan izin yang diperlukan pada fungsi Lambda. Ini agar Amazon S3 memiliki izin untuk menginvokasi fungsi dari bucket. Untuk informasi selengkapnya, lihat Mengaktifkan dan mengonfigurasi notifikasi peristiwa menggunakan konsol Amazon S3.

Anda juga dapat memberikan izin Amazon S3 dari AWS Lambda untuk menjalankan fungsi Lambda Anda. Untuk informasi selengkapnya, lihat Tutorial: Menggunakan AWS Lambda Amazon S3 di Panduan AWS Lambda Pengembang.

memberikan izin untuk memublikasikan pesan ke topik SNS atau antrean SQS

Untuk memberikan izin Amazon S3 untuk memublikasikan pesan ke topik SNS atau antrean SQS, lampirkan kebijakan AWS Identity and Access Management (IAM) ke topik SNS tujuan atau antrean SQS.

Sebagai contoh cara melampirkan kebijakan ke topik SNS atau antrean SQS, lihat Panduan: Mengonfigurasi bucket untuk notifikasi (topik SNS atau antrean SQS). Untuk informasi lebih lanjut tentang izin, lihat topik berikut:

- Contoh kasus untuk kendali akses Amazon SNS dalam Panduan Amazon Simple Notification Service
- Manajemen Identitas dan Akses di Amazon SQS dalam Panduan Pengembang Amazon Simple Queue Service

Kebijakan IAM untuk topik SNS tujuan

Berikut ini adalah contoh kebijakan AWS Identity and Access Management (IAM) yang Anda lampirkan ke topik SNS tujuan. Untuk petunjuk tentang cara menggunakan kebijakan ini untuk menyiapkan topik Amazon SNS tujuan untuk pemberitahuan peristiwa, lihat Panduan: Mengonfigurasi bucket untuk notifikasi (topik SNS atau antrean SQS).

```
{
    "Version": "2012-10-17",
    "Id": "example-ID",
    "Statement": [
```

```
        {
            "Sid": "Example SNS topic policy",
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": [
                "SNS:Publish"
            ],
            "Resource": "SNS-topic-ARN",
            "Condition": {
                "ArnEquals": {
                    "aws:SourceArn": "arn:aws:s3:::amzn-s3-demo-bucket"
                },
                "StringEquals": {
                    "aws:SourceAccount": "bucket-owner-account-id"
                }
            }
        }
    ]
}
```

Kebijakan IAM untuk antrean SQS tujuan

Berikut ini adalah contoh kebijakan IAM yang Anda lampirkan ke antrean SQS tujuan. Untuk petunjuk tentang cara menggunakan kebijakan ini untuk menyiapkan antrean Amazon SQS tujuan untuk pemberitahuan peristiwa, lihat Panduan: Mengonfigurasi bucket untuk notifikasi (topik SNS atau antrean SQS).

Untuk menggunakan kebijakan ini, Anda harus memperbarui ARN antrian Amazon SQS, nama bucket, dan ID pemilik bucket. Akun AWS

```
{
    "Version": "2012-10-17",
    "Id": "example-ID",
    "Statement": [
        {
            "Sid": "example-statement-ID",
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": [
```

```
                "SQS:SendMessage"
            ],
            "Resource": "arn:aws:sqs:Region:account-id:queue-name",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:*:*:awsexamplebucket1"
                },
                "StringEquals": {
                    "aws:SourceAccount": "bucket-owner-account-id"
                }
            }
        }
    ]
}
```

Untuk kebijakan Amazon SNS dan Amazon SQS IAM, Anda dapat menentukan `StringLike` kondisi dalam kebijakan, bukan kondisi `ArnLike`.

Ketika `ArnLike` digunakan, partisi, layanan, account-id, resource-type, dan sebagian resource-id bagian dari ARN harus memiliki pencocokan yang tepat dengan ARN dalam konteks permintaan. Hanya wilayah dan jalur sumber daya yang memungkinkan pencocokan sebagian.

Ketika `StringLike` digunakan sebagai pengganti `ArnLike`, pencocokan mengabaikan struktur ARN dan memungkinkan pencocokan paral, terlepas dari bagian yang diberi wildcard. Untuk informasi selengkapnya, lihat Elemen Kebijakan IAM JSON dalam Panduan Pengguna IAM.

```
"Condition": {
  "StringLike": { "aws:SourceArn": "arn:aws:s3:*:*:amzn-s3-demo-bucket" }
  }
```

AWS KMS kebijakan kunci

Jika antrean SQS atau topik SNS dienkripsi dengan kunci yang dikelola pelanggan AWS Key Management Service (AWS KMS), Anda harus memberikan izin utama layanan Amazon S3 untuk bekerja dengan topik atau antrian terenkripsi. Untuk memberikan izin pengguna utama layanan Amazon S3, tambahkan pernyataan berikut ke kebijakan kunci untuk CMK.

```
{
    "Version": "2012-10-17",
    "Id": "example-ID",
    "Statement": [
```

```
        {
            "Sid": "example-statement-ID",
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*"
        }
    ]
}
```

Untuk informasi selengkapnya tentang kebijakan AWS KMS utama, lihat Menggunakan kebijakan utama AWS KMS di Panduan AWS Key Management Service Pengembang.

Untuk informasi selengkapnya tentang penggunaan enkripsi sisi server dengan Amazon AWS KMS SQS dan Amazon SNS, lihat berikut ini:

- Pengelolaan kunci di Panduan Pengembang Layanan Notifikasi Sederhana Amazon.
- Pengelolaan kunci di Panduan Pengembang Layanan Antrean Sederhana Amazon.
- Mengenkripsi pesan yang diterbitkan untuk Amazon SNS dengan AWS KMS dalam Blog Komputasi AWS .

Mengaktifkan dan mengonfigurasi notifikasi peristiwa menggunakan konsol Amazon S3

Anda dapat mengaktifkan peristiwa bucket tujuan umum Amazon S3 tertentu untuk mengirim pesan notifikasi ke tujuan kapan pun peristiwa tersebut terjadi. Bagian ini menjelaskan cara menggunakan konsol Amazon S3 untuk mengaktifkan pemberitahuan peristiwa. Untuk informasi tentang cara menggunakan notifikasi acara dengan Amazon S3 REST AWS SDKs dan Amazon S3 APIs, lihat. Mengonfigurasi notifikasi peristiwa secara terprogram

Prasyarat: Sebelum dapat mengaktifkan notifikasi peristiwa untuk bucket Anda, Anda harus menyiapkan salah satu jenis tujuan dan kemudian mengonfigurasi izin. Untuk informasi selengkapnya, lihat Tujuan peristiwa yang didukung dan memberikan izin untuk memublikasikan pesan pemberitahuan peristiwa ke tujuan.

> **Note**
>
> Antrean Amazon Simple Queue Service FIFO (First-In-First-Out) tidak didukung sebagai tujuan pemberitahuan peristiwa Amazon S3. Untuk mengirim pemberitahuan untuk acara Amazon S3 ke antrian FIFO Amazon SQS, Anda dapat menggunakan Amazon. EventBridge Untuk informasi selengkapnya, lihat Mengaktifkan Amazon EventBridge.

Topik

- Mengaktifkan notifikasi Amazon SNS, Amazon SQS, atau Lambda menggunakan konsol Amazon S3

Mengaktifkan notifikasi Amazon SNS, Amazon SQS, atau Lambda menggunakan konsol Amazon S3

Untuk mengaktifkan dan mengonfigurasi pemberitahuan peristiwa untuk bucket S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Dalam daftar bucket, pilih nama bucket tempat Anda ingin mengaktifkan event.
4. Pilih Properti.
5. Navigasikan ke Pemberitahuan Acara dan pilih Buat pemberitahuan peristiwa.
6. Di bagian Konfigurasi umum, sebutkan nama peristiwa deskriptif untuk pemberitahuan peristiwa Anda. Atau, Anda juga dapat menentukan prefiks dan sufiks untuk membatasi pemberitahuan ke objek dengan kunci yang berakhir dalam karakter yang ditentukan.

    a. Masukkan deskripsi untuk Nama peristiwa.

    Jika Anda tidak memasukkan nama, pengidentifikasi unik global (GUID) akan dibuat dan digunakan untuk nama.

    b. (Opsional) Untuk memfilter pemberitahuan peristiwa dengan prefiks, masukkan Prefiks.

    Misalnya, Anda dapat menyiapkan filter prefiks sehingga Anda menerima pemberitahuan hanya ketika file ditambahkan ke folder tertentu (misalnya, `images/`).

    c. (Opsional) Untuk memfilter pemberitahuan peristiwa dengan sufiks, masukkan Sufiks.

Untuk informasi selengkapnya, lihat Mengonfigurasi notifikasi peristiwa dengan filter nama kunci objek.

7. Di bagian Jenis peristiwa, pilih satu atau beberapa jenis peristiwa yang ingin Anda terima pemberitahuannya.

   Untuk daftar jenis peristiwa berbeda, lihat Jenis event yang didukung untuk SQS, SNS, dan Lambda.

8. Di bagian Tujuan, pilih tujuan pemberitahuan peristiwa.

   > **Note**
   >
   > Sebelum Anda dapat menerbitkan pemberitahuan peristiwa, Anda harus memberikan izin yang diperlukan kepada pengguna utama Amazon S3 untuk memanggil API terkait. Ini agar dapat menerbitkan pemberitahuan ke fungsi Lambda, topik SNS, atau antrean SQS.

   a. Pilih jenis tujuan: Fungsi Lambda, Topik SNS, atau Antrean SQS.
   b. Setelah memilih jenis tujuan Anda, pilih fungsi, topik, atau antrean dari daftar.
   c. Atau, jika Anda ingin menentukan Amazon Resource Name (ARN), pilih Maemasukkan ARN dan memasukkan ARN.

   Untuk informasi selengkapnya, lihat Tujuan peristiwa yang didukung.

9. Memilih Simpan perubahan, dan Amazon S3 mengirimkan pesan pengujian ke destinasi notifikasi peristiwa.

Mengonfigurasi notifikasi peristiwa secara terprogram

Secara default, notifikasi tidak diaktifkan untuk jenis peristiwa apa pun. Oleh karena itu, awalnya sub-sumber daya notifikasi menyimpan konfigurasi kosong.

```
<NotificationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
</NotificationConfiguration>
```

Untuk mengaktifkan pemberitahuan tentang peristiwa jenis tertentu, Anda mengganti XML dengan konfigurasi yang sesuai yang mengidentifikasi jenis peristiwa yang Anda ingin Amazon S3 untuk

terbitkan dan tujuan di mana Anda ingin peristiwa tersebut diterbitkan. Untuk setiap tujuan, Anda menambahkan konfigurasi XML terkait.

Untuk mempublikasikan pesan peristiwa ke antrean SQS

Untuk mengatur antrean SQS sebagai tujuan notifikasi untuk satu atau beberapa jenis peristiwa, Anda menambahkan `QueueConfiguration`.

```
<NotificationConfiguration>
  <QueueConfiguration>
    <Id>optional-id-string</Id>
    <Queue>sqs-queue-arn</Queue>
    <Event>event-type</Event>
    <Event>event-type</Event>
     ...
  </QueueConfiguration>
   ...
</NotificationConfiguration>
```

Untuk memublikasikan pesan uji ke topik SNS

Untuk menetapkan topik SNS sebagai tujuan notifikasi untuk jenis peristiwa tertentu, Anda menambahkan `TopicConfiguration`.

```
<NotificationConfiguration>
  <TopicConfiguration>
     <Id>optional-id-string</Id>
     <Topic>sns-topic-arn</Topic>
     <Event>event-type</Event>
     <Event>event-type</Event>
      ...
  </TopicConfiguration>
   ...
</NotificationConfiguration>
```

Untuk memanggil AWS Lambda fungsi dan memberikan pesan acara sebagai argumen

Untuk mengatur fungsi Lambda sebagai tujuan notifikasi untuk jenis peristiwa tertentu, Anda menambahkan `CloudFunctionConfiguration`.

```
<NotificationConfiguration>
  <CloudFunctionConfiguration>
```

```
      <Id>optional-id-string</Id>
      <CloudFunction>cloud-function-arn</CloudFunction>
      <Event>event-type</Event>
      <Event>event-type</Event>
       ...
   </CloudFunctionConfiguration>
    ...
</NotificationConfiguration>
```

Untuk menghapus semua notifikasi yang dikonfigurasi pada bucket

Untuk menghapus semua notifikasi yang dikonfigurasi di bucket, simpan elemen `<NotificationConfiguration/>` kosong dalam sub-sumber daya notifikasi.

Saat Amazon S3 mendeteksi peristiwa dengan jenis tertentu, itu menerbitkan pesan dengan informasi peristiwa. Untuk informasi selengkapnya, lihat Struktur pesan peristiwa.

Untuk informasi lebih lanjut tentang mengonfigurasi notifikasi Peristiwa, lihat topik berikut:

- Panduan: Mengonfigurasi bucket untuk notifikasi (topik SNS atau antrean SQS).
- Mengonfigurasi notifikasi peristiwa dengan filter nama kunci objek

Panduan: Mengonfigurasi bucket untuk notifikasi (topik SNS atau antrean SQS)

Anda dapat menerima notifikasi Amazon S3 menggunakan Amazon Simple Notification Service (Amazon SNS); atau Amazon Simple Queue Service (Amazon SQS). Dalam panduan selanjutnya, Anda akan menambahkan konfigurasi notifikasi ke bucket Anda menggunakan topik Amazon SNS dan antrean Amazon SQS.

> **Note**
>
> Antrean Amazon Simple Queue Service FIFO (First-In-First-Out) tidak didukung sebagai tujuan pemberitahuan peristiwa Amazon S3. Untuk mengirim pemberitahuan untuk acara Amazon S3 ke antrian FIFO Amazon SQS, Anda dapat menggunakan Amazon. EventBridge Untuk informasi selengkapnya, lihat Mengaktifkan Amazon EventBridge.

Topik

- Ringkasan panduan
- Langkah 1: Buat antrean Amazon SQS

- Langkah 2: Buat Topik Amazon SNS
- Langkah 3: Menambahkan konfigurasi notifikasi ke bucket Anda
- Langkah 4: Menguji pengaturan

Ringkasan panduan

Panduan ini membantu Anda melakukan hal berikut ini:

- Terbitkan peristiwa dari jenis `s3:ObjectCreated:*` ke antrean Amazon SQS.
- Terbitkan peristiwa dari jenis `s3:ReducedRedundancyLostObject` ke topik Amazon SNS.

Untuk informasi tentang konfigurasi pemberitahuan, lihat Menggunakan Amazon SQS, Amazon SNS, dan Lambda.

Anda dapat melakukan semua langkah ini menggunakan konsol, tanpa menulis kode apa pun. Selain itu, contoh kode yang digunakan AWS SDKs untuk Java dan .NET juga disediakan untuk membantu Anda menambahkan konfigurasi notifikasi secara terprogram.

Prosedur ini mencakup langkah-langkah berikut:

1. Membuat antrean Amazon SQS.

   Menggunakan konsol Amazon SQS, buat antrean SQS. Anda dapat mengakses pesan apa pun yang dikirimkan Amazon S3 ke antrean secara terprogram. Namun, untuk panduan ini, Anda memverifikasi pesan pemberitahuan di konsol.

   Anda melampirkan kebijakan akses ke antrean untuk memberikan izin kepada Amazon S3 untuk memposting pesan.
2. Buat topik Amazon SNS.

   Dengan konsol Amazon SNS, buat topik SNS dan berlangganan ke topik tersebut. Dengan begitu, setiap peristiwa yang diposting ke sana dikirimkan kepada Anda. Anda menentukan email sebagai protokol komunikasi. Setelah Anda membuat topik, Amazon SNS mengirim email. Anda menggunakan tautan dalam email untuk mengonfirmasi langganan topik.

   Anda melampirkan kebijakan akses ke topik untuk memberikan izin kepada Amazon S3 untuk memposting pesan.
3. Tambahkan konfigurasi pemberitahuan ke bucket.

Langkah 1: Buat antrean Amazon SQS

Ikuti langkah-langkah untuk membuat dan berlangganan antrean Amazon Simple Queue Service (Amazon SQS).

1. Menggunakan konsol Amazon SQS, buat antrean. Untuk instruksi, lihat Memulai dengan Amazon SQS dalam Panduan Pengembang Amazon Simple Queue Service.
2. Ganti kebijakan akses yang terlampir pada antrean dengan kebijakan berikut.

   a. Di konsol Amazon SQS, dalam daftar Antrian, pilih nama antrean.

   b. Pada tab Kebijakan akses, pilih Edit.

   c. Ganti kebijakan akses yang terlampir ke antrean. Di dalamnya, berikan ARN Amazon SQS, nama bucket sumber, dan ID akun pemilik bucket.

```
{
    "Version": "2012-10-17",
    "Id": "example-ID",
    "Statement": [
        {
            "Sid": "example-statement-ID",
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": [
                "SQS:SendMessage"
            ],
            "Resource": "SQS-queue-ARN",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:*:*:awsexamplebucket1"
                },
                "StringEquals": {
                    "aws:SourceAccount": "bucket-owner-account-id"
                }
            }
        }
    ]
}
```

   d. Pilih Simpan.

3. (Opsional) Jika antrean Amazon SQS atau topik Amazon SNS diaktifkan dengan enkripsi sisi server dengan AWS Key Management Service (AWS KMS), tambahkan kebijakan berikut ke kunci terkelola pelanggan enkripsi simetris terkait.

   Anda harus menambahkan kebijakan ke CMK karena Anda tidak dapat memodifikasi kunci AWS terkelola untuk Amazon SQS atau Amazon SNS.

   ```
   {
       "Version": "2012-10-17",
       "Id": "example-ID",
       "Statement": [
           {
               "Sid": "example-statement-ID",
               "Effect": "Allow",
               "Principal": {
                   "Service": "s3.amazonaws.com"
               },
               "Action": [
                   "kms:GenerateDataKey",
                   "kms:Decrypt"
               ],
               "Resource": "*"
           }
       ]
   }
   ```

   Untuk informasi selengkapnya tentang penggunaan SSE untuk Amazon SQS dan Amazon SNS AWS KMS with, lihat berikut ini:

   - Pengelolaan kunci di Panduan Pengembang Layanan Notifikasi Sederhana Amazon.
   - Pengelolaan kunci di Panduan Pengembang Layanan Antrean Sederhana Amazon.

4. Perhatikan antrean ARN.

   Antrean SQS yang Anda buat adalah sumber daya lain di Akun AWS Anda. Amazon Resource Name (ARN) yang unik. Anda membutuhkan ARN ini di langkah berikutnya. ARN memiliki format berikut:

   ```
   arn:aws:sqs:aws-region:account-id:queue-name
   ```

Langkah 2: Buat Topik Amazon SNS

Ikuti langkah-langkah untuk membuat dan berlangganan topik Amazon SNS.

1. Menggunakan konsol Amazon SNS membuat topik. Untuk instruksi, lihat Membuat topik Amazon SNS dalam Panduan Pengembang Amazon Simple Notification Service.
2. Berlangganan topik tersebut. Untuk latihan ini, gunakan email sebagai protokol komunikasi. Untuk instruksi, lihat Berlangganan topik Amazon SNS dalam Panduan Pengembang Amazon Simple Notification Service.

   Anda mendapatkan email yang meminta Anda untuk mengonfirmasi langganan Anda ke topik tersebut. Konfirmasi langganan.
3. Ganti kebijakan akses yang terlampir pada topik dengan kebijakan berikut. Di dalamnya, berikan ARN topik SNS, nama bucket, dan ID akun pemilik bucket.

```
{
    "Version": "2012-10-17",
    "Id": "example-ID",
    "Statement": [
        {
            "Sid": "Example SNS topic policy",
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": [
                "SNS:Publish"
            ],
            "Resource": "SNS-topic-ARN",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:*:*:amzn-s3-demo-bucket"
                },
                "StringEquals": {
                    "aws:SourceAccount": "bucket-owner-account-id"
                }
            }
        }
    ]
}
```

4. Perhatikan topik ARN.

Topik SNS yang Anda buat adalah sumber daya lain di Anda Akun AWS, dan memiliki ARN yang unik. Anda akan membutuhkan ARN ini di langkah berikutnya. ARN akan memiliki format berikut:

```
arn:aws:sns:aws-region:account-id:topic-name
```

Langkah 3: Menambahkan konfigurasi notifikasi ke bucket Anda

Anda dapat mengaktifkan notifikasi bucket baik dengan menggunakan konsol Amazon S3 atau secara terprogram dengan menggunakan. AWS SDKs Memilih salah satu opsi untuk mengonfigurasi notifikasi pada bucket Anda. Bagian ini memberikan contoh kode menggunakan AWS SDKs untuk Java dan .NET.

Opsi A: Mengaktifkan notifikasi pada bucket menggunakan konsol tersebut

Menggunakan konsol Amazon S3, menambahkan konfigurasi notifikasi yang meminta Amazon S3 untuk melakukan hal berikut ini:

- Terbitkan peristiwa Semua objek membuat peristiwa ke antrean Amazon SQS Anda.
- Terbitkan peristiwa Objek dalam RRS hilang ketik topik Amazon SNS Anda.

Setelah Anda menyimpan konfigurasi pemberitahuan, Amazon S3 memposting pesan uji, yang Anda dapatkan melalui email.

Untuk petunjuk, lihat Mengaktifkan dan mengonfigurasi notifikasi peristiwa menggunakan konsol Amazon S3.

Opsi B: Aktifkan notifikasi pada ember menggunakan AWS SDKs

.NET

Contoh kode C# berikut memberikan daftar kode lengkap yang menambahkan konfigurasi pemberitahuan ke bucket. Anda harus memperbarui kode dan memberikan nama bucket dan ARN topik SNS. Untuk informasi tentang menyiapkan dan menjalankan contoh kode, lihat Memulai SDK for .NET di AWSAWS SDK for .NET Developer Guide.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
```

```
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class EnableNotificationsTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string snsTopic = "*** SNS topic ARN ***";
        private const string sqsQueue = "*** SQS topic ARN ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            EnableNotificationAsync().Wait();
        }

        static async Task EnableNotificationAsync()
        {
            try
            {
               PutBucketNotificationRequest request = new
 PutBucketNotificationRequest
                {
                    BucketName = bucketName
                };

                TopicConfiguration c = new TopicConfiguration
                {
                    Events = new List<EventType> { EventType.ObjectCreatedCopy },
                    Topic = snsTopic
                };
                request.TopicConfigurations = new List<TopicConfiguration>();
                request.TopicConfigurations.Add(c);
                request.QueueConfigurations = new List<QueueConfiguration>();
                request.QueueConfigurations.Add(new QueueConfiguration()
                {
                    Events = new List<EventType> { EventType.ObjectCreatedPut },
                    Queue = sqsQueue
                });
```

```
                PutBucketNotificationResponse response = await
 client.PutBucketNotificationAsync(request);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' ",
 e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown error encountered on server.
 Message:'{0}' ", e.Message);
            }
        }
    }
}
```

Java

Contoh berikut menunjukkan cara menambahkan konfigurasi pemberitahuan ke bucket. Untuk petunjuk tentang cara membuat dan menguji sampel kerja, lihat Memulai di Panduan AWS SDK untuk Java Pengembang.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.io.IOException;
import java.util.EnumSet;

public class EnableNotificationOnABucket {

    public static void main(String[] args) throws IOException {
        String bucketName = "*** Bucket name ***";
        Regions clientRegion = Regions.DEFAULT_REGION;
        String snsTopicARN = "*** SNS Topic ARN ***";
        String sqsQueueARN = "*** SQS Queue ARN ***";
```

```
        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();
            BucketNotificationConfiguration notificationConfiguration = new
 BucketNotificationConfiguration();

            // Add an SNS topic notification.
            notificationConfiguration.addConfiguration("snsTopicConfig",
                    new TopicConfiguration(snsTopicARN,
 EnumSet.of(S3Event.ObjectCreated)));

            // Add an SQS queue notification.
            notificationConfiguration.addConfiguration("sqsQueueConfig",
                    new QueueConfiguration(sqsQueueARN,
 EnumSet.of(S3Event.ObjectCreated)));

            // Create the notification configuration request and set the bucket
 notification
            // configuration.
            SetBucketNotificationConfigurationRequest request = new
 SetBucketNotificationConfigurationRequest(
                    bucketName, notificationConfiguration);
            s3Client.setBucketNotificationConfiguration(request);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Langkah 4: Menguji pengaturan

Sekarang Anda dapat menguji pengaturan dengan mengunggah objek ke bucket Anda dan memverifikasi pemberitahuan peristiwa di konsol Amazon SQS. Untuk instruksi, lihat Menerima Pesan dalam Bagian "Memulai" Panduan Pengembang Amazon Simple Queue Service.

Mengonfigurasi notifikasi peristiwa dengan filter nama kunci objek

Ketika mengkonfigurasi notifikasi peristiwa Amazon S3, Anda harus menentukan mana jenis peristiwa Amazon S3 yang didukung yang menyebabkan Amazon S3 untuk mengirim notifikasi. Jika jenis peristiwa yang Anda tidak tentukan terjadi di bucket S3 Anda, Amazon S3 tidak mengirim notifikasi.

Anda dapat mengonfigurasi pemberitahuan untuk difilter dengan prefiks dan sufiks nama kunci objek. Misalnya, Anda dapat menyiapkan konfigurasi di mana Anda akan dikirimi pemberitahuan hanya ketika file gambar dengan ekstensi nama file "`.jpg`" file ditambahkan ke bucket. Atau, Anda dapat memiliki konfigurasi yang mengirimkan pemberitahuan ke topik Amazon SNS ketika objek dengan awalan `images/` "" ditambahkan ke ember, sambil memiliki pemberitahuan untuk objek dengan awalan `logs/` "" di ember yang sama dikirim ke suatu fungsi. AWS Lambda

> **Note**
>
> Karakter wildcard (“*”) tidak dapat digunakan dalam filter sebagai prefiks atau sufiks. Jika prefiks atau sufiks Anda mengandung spasi, Anda harus menggantinya dengan karakter "+". Jika Anda menggunakan karakter kustom apa pun dalam nilai prefiks atau sufiks, Anda harus memasukkan mereka ke dalam Format yang dikodekan URL (persen-dikodekan). Untuk daftar lengkap karakter kustom yang harus dikonversi ke format yang dikodekan URL saat digunakan dalam prefiks atau sufiks untuk pemberitahuan peristiwa, lihat. Karakter aman

Anda dapat mengatur konfigurasi notifikasi yang menggunakan filter nama kunci objek di konsol Amazon S3. Anda dapat melakukannya dengan menggunakan Amazon S3 APIs melalui AWS SDKs atau REST APIs secara langsung. Untuk informasi tentang penggunaan UI konsol tersebut untuk mengatur konfigurasi notifikasi pada bucket, lihat Mengaktifkan dan mengonfigurasi notifikasi peristiwa menggunakan konsol Amazon S3.

Amazon S3 menyimpan konfigurasi pemberitahuan sebagai XML di sub-sumber daya notifikasi yang terkait dengan bucket, seperti yang telah dijelaskan di Menggunakan Amazon SQS, Amazon SNS, dan Lambda. Anda menggunakan struktur XML `Filter` untuk menentukan aturan notifikasi yang

akan difilter dengan prefiks atau sufiks nama kunci objek. Untuk informasi tentang struktur XML `Filter`, lihat Notifikasi PUT Bucket di Referensi API Amazon Simple Storage Service.

Konfigurasi pemberitahuan yang menggunakan `Filter` tidak dapat menentukan aturan pemfilteran dengan prefiks yang tumpang tindih, sufiks yang tumpang tindih, atau prefiks dan sufiks yang tumpang tindih. Bagian berikut memiliki contoh konfigurasi pemberitahuan yang valid dengan pemfilteran nama kunci objek. Juga berisi contoh konfigurasi notifikasi yang tidak valid karena prefiks dan sufiks tumpang tindih.

Topik

- Contoh konfigurasi pemberitahuan yang valid dengan pemfilteran nama kunci objek
- Contoh konfigurasi notifikasi dengan prefiks dan sufiks yang tidak valid saling tumpang tindih

Contoh konfigurasi pemberitahuan yang valid dengan pemfilteran nama kunci objek

Konfigurasi notifikasi berikut berisi konfigurasi antrean yang mengidentifikasi antrean Amazon SQS untuk Amazon S3 untuk menerbitkan peristiwa ke jenis `s3:ObjectCreated:Put`. Peristiwa dipublikasikan kapan pun objek yang memiliki prefiks `images/` dan sufiks `jpg` di-PUT ke dalam bucket.

```
<NotificationConfiguration>
  <QueueConfiguration>
      <Id>1</Id>
      <Filter>
          <S3Key>
              <FilterRule>
                  <Name>prefix</Name>
                  <Value>images/</Value>
              </FilterRule>
              <FilterRule>
                  <Name>suffix</Name>
                  <Value>jpg</Value>
              </FilterRule>
          </S3Key>
     </Filter>
     <Queue>arn:aws:sqs:us-west-2:444455556666:s3notificationqueue</Queue>
     <Event>s3:ObjectCreated:Put</Event>
  </QueueConfiguration>
</NotificationConfiguration>
```

Konfigurasi pemberitahuan berikut memiliki beberapa prefiks yang tidak tumpang tindih. Konfigurasi tersebut menentukan bahwa notifikasi untuk permintaan PUT di folder `images/` masuk ke antrean A, sedangkan notifikasi untuk permintaan PUT di `logs/` masuk ke antrean B.

```
<NotificationConfiguration>
  <QueueConfiguration>
     <Id>1</Id>
     <Filter>
            <S3Key>
                <FilterRule>
                    <Name>prefix</Name>
                    <Value>images/</Value>
                </FilterRule>
            </S3Key>
     </Filter>
     <Queue>arn:aws:sqs:us-west-2:444455556666:sqs-queue-A</Queue>
     <Event>s3:ObjectCreated:Put</Event>
  </QueueConfiguration>
  <QueueConfiguration>
     <Id>2</Id>
     <Filter>
            <S3Key>
                <FilterRule>
                    <Name>prefix</Name>
                    <Value>logs/</Value>
                </FilterRule>
            </S3Key>
     </Filter>
     <Queue>arn:aws:sqs:us-west-2:444455556666:sqs-queue-B</Queue>
     <Event>s3:ObjectCreated:Put</Event>
  </QueueConfiguration>
</NotificationConfiguration>
```

Konfigurasi notifikasi berikut memiliki beberapa sufiks yang tidak tumpang tindih. Konfigurasi mendefinisikan bahwa semua `.jpg` gambar yang baru ditambahkan ke bucket diproses oleh cloud-function-A Lambda, dan semua gambar yang baru `.png` ditambahkan diproses oleh. cloud-function-B Sufiks `.png` dan `.jpg` sufiks tidak tumpang tindih meskipun mereka memiliki huruf terakhir yang sama. Jika string tertentu dapat diakhiri dengan kedua sufiks, kedua sufiks dianggap sebagai tumpang tindih. String tidak dapat diakhiri dengan kedua `.png` dan `.jpg`, sehingga sufiks dalam konfigurasi contoh tidak tumpang tindih dengan sufiks.

```
<NotificationConfiguration>
  <CloudFunctionConfiguration>
     <Id>1</Id>
     <Filter>
            <S3Key>
                <FilterRule>
                    <Name>suffix</Name>
                    <Value>.jpg</Value>
                </FilterRule>
            </S3Key>
     </Filter>
     <CloudFunction>arn:aws:lambda:us-west-2:444455556666:cloud-function-A</
CloudFunction>
     <Event>s3:ObjectCreated:Put</Event>
  </CloudFunctionConfiguration>
  <CloudFunctionConfiguration>
     <Id>2</Id>
     <Filter>
            <S3Key>
                <FilterRule>
                    <Name>suffix</Name>
                    <Value>.png</Value>
                </FilterRule>
            </S3Key>
     </Filter>
     <CloudFunction>arn:aws:lambda:us-west-2:444455556666:cloud-function-B</
CloudFunction>
     <Event>s3:ObjectCreated:Put</Event>
  </CloudFunctionConfiguration>
</NotificationConfiguration>
```

Konfigurasi notifikasi Anda yang menggunakan `Filter` tidak dapat menentukan aturan pemfilteran dengan prefiks yang tumpang tindih untuk jenis peristiwa yang sama. Mereka hanya bisa melakukannya, jika prefiks tumpang tindih yang digunakan dengan sufiks yang tidak tumpang tindih. Konfigurasi contoh berikut ini menunjukkan bagaimana objek yang dibuat dengan prefiks yang sama, tetapi sufiks yang tidak tumpang tindih dapat dikirim ke tujuan yang berbeda.

```
<NotificationConfiguration>
  <CloudFunctionConfiguration>
     <Id>1</Id>
     <Filter>
            <S3Key>
```

```
                <FilterRule>
                    <Name>prefix</Name>
                    <Value>images</Value>
                </FilterRule>
                <FilterRule>
                    <Name>suffix</Name>
                    <Value>.jpg</Value>
                </FilterRule>
            </S3Key>
     </Filter>
     <CloudFunction>arn:aws:lambda:us-west-2:444455556666:cloud-function-A</
CloudFunction>
     <Event>s3:ObjectCreated:Put</Event>
  </CloudFunctionConfiguration>
  <CloudFunctionConfiguration>
     <Id>2</Id>
     <Filter>
            <S3Key>
                <FilterRule>
                    <Name>prefix</Name>
                    <Value>images</Value>
                </FilterRule>
                <FilterRule>
                    <Name>suffix</Name>
                    <Value>.png</Value>
                </FilterRule>
            </S3Key>
     </Filter>
     <CloudFunction>arn:aws:lambda:us-west-2:444455556666:cloud-function-B</
CloudFunction>
     <Event>s3:ObjectCreated:Put</Event>
  </CloudFunctionConfiguration>
</NotificationConfiguration>
```

Contoh konfigurasi notifikasi dengan prefiks dan sufiks yang tidak valid saling tumpang tindih

Pada umumnya, konfigurasi pemberitahuan Anda yang menggunakan `Filter` tidak dapat menentukan aturan pemfilteran dengan prefiks yang tumpang tindih, sufiks yang tumpang tindih, atau kombinasi prefiks dan sufiks yang tumpang tindih untuk jenis peristiwa yang sama. Anda dapat memiliki prefiks yang tumpang tindih selama sufiks tidak tumpang tindih. Sebagai contoh, lihat Mengonfigurasi notifikasi peristiwa dengan filter nama kunci objek.

Anda dapat menggunakan filter nama kunci objek yang tumpang tindih dengan jenis peristiwa yang berbeda. Misalnya, Anda dapat membuat konfigurasi notifikasi yang menggunakan prefiks `image/` untuk jenis peristiwa `ObjectCreated:Put`, dan prefiks `image/` untuk jenis peristiwa `ObjectRemoved:*`.

Anda mendapatkan kesalahan jika mencoba menyimpan konfigurasi pemberitahuan yang memiliki filter nama yang tumpang tindih yang tidak valid untuk jenis peristiwa yang sama saat menggunakan konsol atau API Amazon S3. Bagian ini menunjukkan contoh konfigurasi notifikasi yang tidak valid karena filter nama yang tumpang tindih.

Aturan konfigurasi pemberitahuan apa pun yang ada diasumsikan memiliki prefiks dan sufiks default yang cocok dengan setiap prefiks dan sufiks lain. Konfigurasi pemberitahuan berikut tidak valid karena memiliki prefiks yang tumpang tindih. Secara kustom, prefiks root tumpang tindih dengan prefiks lainnya. Hal yang sama berlaku jika Anda menggunakan sufiks alih-alih prefiks dalam contoh ini. Akhiran root tumpang tindih dengan sufiks lainnya.

```
<NotificationConfiguration>
     <TopicConfiguration>
         <Topic>arn:aws:sns:us-west-2:444455556666:sns-notification-one</Topic>
         <Event>s3:ObjectCreated:*</Event>
    </TopicConfiguration>
    <TopicConfiguration>
         <Topic>arn:aws:sns:us-west-2:444455556666:sns-notification-two</Topic>
         <Event>s3:ObjectCreated:*</Event>
         <Filter>
             <S3Key>
                 <FilterRule>
                     <Name>prefix</Name>
                     <Value>images</Value>
                 </FilterRule>
            </S3Key>
        </Filter>
    </TopicConfiguration>
</NotificationConfiguration>
```

Konfigurasi pemberitahuan berikut tidak valid karena memiliki sufiks yang tumpang tindih. Jika string tertentu dapat diakhiri dengan kedua sufiks, kedua sufiks dianggap sebagai tumpang tindih. Sebuah string dapat diakhiri dengan `jpg` dan `pg`. Jadi, sufiksnya tumpang tindih. Hal yang sama berlaku untuk prefiks. Jika string tertentu dapat dimulai dengan kedua prefiks, kedua prefiks dianggap tumpang tindih.

```
 <NotificationConfiguration>
     <TopicConfiguration>
         <Topic>arn:aws:sns:us-west-2:444455556666:sns-topic-one</Topic>
         <Event>s3:ObjectCreated:*</Event>
         <Filter>
             <S3Key>
                 <FilterRule>
                     <Name>suffix</Name>
                     <Value>jpg</Value>
                 </FilterRule>
            </S3Key>
        </Filter>
    </TopicConfiguration>
    <TopicConfiguration>
         <Topic>arn:aws:sns:us-west-2:444455556666:sns-topic-two</Topic>
         <Event>s3:ObjectCreated:Put</Event>
         <Filter>
             <S3Key>
                 <FilterRule>
                     <Name>suffix</Name>
                     <Value>pg</Value>
                 </FilterRule>
            </S3Key>
        </Filter>
    </TopicConfiguration>
</NotificationConfiguration
```

Konfigurasi notifikasi berikut tidak valid karena memiliki prefiks dan sufiks yang tumpang tindih.

```
<NotificationConfiguration>
     <TopicConfiguration>
         <Topic>arn:aws:sns:us-west-2:444455556666:sns-topic-one</Topic>
         <Event>s3:ObjectCreated:*</Event>
         <Filter>
             <S3Key>
                 <FilterRule>
                     <Name>prefix</Name>
                     <Value>images</Value>
                 </FilterRule>
                 <FilterRule>
                     <Name>suffix</Name>
                     <Value>jpg</Value>
                 </FilterRule>
```

```
            </S3Key>
        </Filter>
    </TopicConfiguration>
    <TopicConfiguration>
         <Topic>arn:aws:sns:us-west-2:444455556666:sns-topic-two</Topic>
         <Event>s3:ObjectCreated:Put</Event>
         <Filter>
             <S3Key>
                 <FilterRule>
                     <Name>suffix</Name>
                     <Value>jpg</Value>
                 </FilterRule>
            </S3Key>
        </Filter>
    </TopicConfiguration>
</NotificationConfiguration>
```

Struktur pesan peristiwa

Pesan pemberitahuan yang dikirimkan Amazon S3 untuk menerbitkan kegiatan ada dalam format JSON.

Untuk ikhtisar umum dan petunjuk tentang mengonfigurasi pemberitahuan peristiwa, lihat Notifikasi Peristiwa Amazon S3.

Contoh ini menunjukkan versi 2.2 dari struktur JSON notifikasi peristiwa. Amazon S3 menggunakan versi 2.1, 2.2, dan 2.3 dari struktur peristiwa ini. Amazon S3 menggunakan versi 2.2 untuk pemberitahuan peristiwa replikasi lintas wilayah. Ini menggunakan versi 2.3 untuk Siklus Hidup S3, S3 Intelligent-Tiering, objek ACL, penandaan objek, dan peristiwa penghapusan restorasi objek. Versi ini berisi informasi tambahan kustom untuk operasi ini. Versi 2.2 dan 2.3 sebaliknya kompatibel dengan versi 2.1, yang saat ini digunakan Amazon S3 untuk semua jenis pemberitahuan peristiwa lainnya.

```
{
   "Records":[
      {
         "eventVersion":"2.2",
         "eventSource":"aws:s3",
         "awsRegion":"us-west-2",
         "eventTime":"The time, in ISO-8601 format, for example,
 1970-01-01T00:00:00.000Z, when Amazon S3 finished processing the request",
         "eventName":"event-type",
```

```
         "userIdentity":{
            "principalId":"Amazon-customer-ID-of-the-user-who-caused-the-event"
         },
         "requestParameters":{
            "sourceIPAddress":"ip-address-where-request-came-from"
         },
         "responseElements":{
            "x-amz-request-id":"Amazon S3 generated request ID",
            "x-amz-id-2":"Amazon S3 host that processed the request"
         },
         "s3":{
            "s3SchemaVersion":"1.0",
            "configurationId":"ID found in the bucket notification configuration",
            "bucket":{
               "name":"amzn-s3-demo-bucket",
               "ownerIdentity":{
                  "principalId":"Amazon-customer-ID-of-the-bucket-owner"
               },
               "arn":"bucket-ARN"
            },
            "object":{
               "key":"object-key",
               "size":"object-size in bytes",
               "eTag":"object eTag",
               "versionId":"object version if bucket is versioning-enabled, otherwise
 null",
               "sequencer": "a string representation of a hexadecimal value used to
 determine event sequence, only used with PUTs and DELETEs"
            }
         },
         "glacierEventData": {
            "restoreEventData": {
               "lifecycleRestorationExpiryTime": "The time, in ISO-8601 format, for
 example, 1970-01-01T00:00:00.000Z, of Restore Expiry",
               "lifecycleRestoreStorageClass": "Source storage class for restore"
            }
         }
      }
   ]
}
```

Perhatikan hal berikut tentang struktur pesan peristiwa:

- Nilai kunci `eventVersion` mengandung versi besar dan kecil, dalam bentuk `<major>.<minor>`.

  Versi utama bertambah jika Amazon S3 membuat perubahan pada struktur peristiwa yang tidak kompatibel dengan versi sebelumnya. Ini termasuk menghapus bidang JSON yang sudah ada atau mengubah bagaimana isi bidang direpresentasikan (misalnya, format tanggal).

  Versi minor ditingkatkan jika Amazon S3 menambahkan bidang baru ke struktur peristiwa. Hal ini dapat terjadi jika informasi baru diberikan untuk beberapa atau semua peristiwa yang ada. Ini mungkin juga terjadi jika informasi baru hanya diberikan pada jenis peristiwa yang baru diperkenalkan. Aplikasi sebaiknya mengabaikan bidang baru untuk tetap kompatibel dengan versi kecil struktur peristiwa baru.

  Jika jenis peristiwa baru diperkenalkan tetapi struktur peristiwa tidak dimodifikasi, versi peristiwa tidak akan berubah.

  Untuk memastikan aplikasi Anda dapat menguraikan struktur peristiwa dengan benar, kami menyarankan Anda melakukan perbandingan yang setara dengan nomor versi besar. Untuk memastikan bahwa bidang yang diharapkan oleh aplikasi Anda ada, kami juga merekomendasikan melakukan perbandingan greater-than-or-equal -to pada versi minor.
- Angka `eventName` mereferensikan daftar jenis notifikasiperistiwa tetapi tidak berisi prefiks `s3:`.
- Nilai `responseElements` kunci berguna jika Anda ingin melacak permintaan dengan menindaklanjuti AWS Dukungan. Baik `x-amz-request-id` maupun `x-amz-id-2` membantu Amazon S3 menelusuri permintaan individu. Nilai-nilai ini sama dengan nilai yang dikembalikan Amazon S3 sebagai respons terhadap permintaan yang memulai peristiwa. Ini agar mereka dapat digunakan untuk mencocokkan peristiwa dengan permintaan.
- Kunci `s3` memberikan informasi tentang bucket dan objek yang terlibat dalam peristiwa tersebut. Nilai nama kunci objek dienkode URL. Misalnya, "red flower.jpg" menjadi "red+flower.jpg" (Amazon S3 mengembalikan "`application/x-www-form-urlencoded`" sebagai jenis konten di dalam respons tersebut).
- Kunci `sequencer` memberikan cara untuk menentukan urutan peristiwa. Pemberitahuan peristiwa tidak dijamin akan tiba dalam urutan yang sama dengan peristiwa yang terjadi. Namun, pemberitahuan dari peristiwa yang membuat objek (PUT) dan menghapus objek berisi file `sequencer`. Ini dapat digunakan untuk menentukan urutan peristiwa untuk kunci objek tertentu.

  Jika Anda membandingkan string `sequencer` dari dua notifikasi peristiwa pada kunci objek yang sama, pemberitahuan peristiwa dengan nilai heksadesimal `sequencer` yang lebih besar adalah peristiwa yang terjadi kemudian. Jika Anda menggunakan notifikasi peristiwa untuk memelihara

basis data atau indeks terpisah dari objek Amazon S3, kami sarankan Anda membandingkan dan menyimpan nilai `sequencer` saat Anda memproses setiap pemberitahuan peristiwa.

Perhatikan hal berikut:

- Anda tidak dapat menggunakan `sequencer` untuk menentukan urutan peristiwa dengan kunci objek yang berbeda.
- Urutan dapat memiliki panjang berbeda. Jadi, untuk membandingkan nilai-nilai ini, pertama-tama masukkan nilai yang lebih pendek dengan nol, dan kemudian melakukan perbandingan leksikografis.

- Kunci `glacierEventData` hanya terlihat untuk peristiwa `s3:ObjectRestore:Completed`.
- Kunci `restoreEventData` berisi atribut yang terkait dengan permintaan pemulihan Anda.
- Kunci `replicationEventData` hanya terlihat untuk peristiwa replikasi.
- Kunci `intelligentTieringEventData` hanya terlihat untuk peristiwa S3 Intelligent-Tiering.
- Kunci `lifecycleEventData` hanya terlihat untuk peristiwa transisi Siklus Hidup S3.

Contoh pesan

Berikut ini adalah contoh dari pesan notifikasi peristiwa Amazon S3.

Olahpesan pengujian Amazon S3

Setelah mengonfigurasi notifikasi peristiwa di bucket, Amazon S3 mengirimkan pesan pengujian berikut.

```
{
   "Service":"Amazon S3",
   "Event":"s3:TestEvent",
   "Time":"2014-10-13T15:57:02.089Z",
   "Bucket":"amzn-s3-demo-bucket",
   "RequestId":"5582815E1AEA5ADF",
   "HostId":"8cLeGAmw098X5cv4Zkwcmo8vvZa3eH3eKxsPzbB9wrR+YstdA6Knx4Ip8EXAMPLE"
}
```

> **Note**
>
> `s3:TestEvent`Pesan menggunakan format yang berbeda dari notifikasi acara S3 biasa. Tidak seperti pemberitahuan acara lain yang menggunakan struktur `Records` array yang ditunjukkan di atas, acara pengujian menggunakan format yang disederhanakan

dengan bidang langsung. Saat menerapkan penanganan acara, pastikan kode Anda dapat membedakan antara dan menangani kedua format pesan dengan benar.

Contoh pesan ketika sebuah objek dibuat menggunakan permintaan PUT

Olahpesan berikut adalah contoh pesan yang dikirimkan Amazon S3 untuk menerbitkan peristiwa `s3:ObjectCreated:Put`.


```
{
   "Records":[
      {
         "eventVersion":"2.1",
         "eventSource":"aws:s3",
         "awsRegion":"us-west-2",
         "eventTime":"1970-01-01T00:00:00.000Z",
         "eventName":"ObjectCreated:Put",
         "userIdentity":{
            "principalId":"AIDAJDPLRKLG7UEXAMPLE"
         },
         "requestParameters":{
            "sourceIPAddress":"127.0.0.1"
         },
         "responseElements":{
            "x-amz-request-id":"C3D13FE58DE4C810",
            "x-amz-id-2":"FMyUVURIY8/IgAtTv8xRjskZQpcIZ9KG4V5Wp6S7S/
JRWeUWerMUE5JgHvANOjpD"
         },
         "s3":{
            "s3SchemaVersion":"1.0",
            "configurationId":"testConfigRule",
            "bucket":{
               "name":"amzn-s3-demo-bucket",
               "ownerIdentity":{
                  "principalId":"A3NL1KOZZKExample"
               },
               "arn":"arn:aws:s3:::amzn-s3-demo-bucket"
            },
            "object":{
               "key":"HappyFace.jpg",
               "size":1024,
               "eTag":"d41d8cd98f00b204e9800998ecf8427e",
               "versionId":"096fKKXTRTtl3on89fVO.nfljtsv6qko",
```

```
                "sequencer":"0055AED6DCD90281E5"
            }
         }
      }
   ]
}
```

Untuk definisi setiap prefiks identifikasi IAM (misalnya, AIDA, AROA, AGPA), lihat Pengidentifikasi IAM di Panduan Pengguna IAM.

Menggunakan EventBridge

Amazon S3 dapat mengirim acara ke Amazon EventBridge setiap kali peristiwa tertentu terjadi di bucket Anda. Tidak seperti tujuan lain, Anda tidak perlu memilih jenis peristiwa mana yang ingin Anda kirimkan. Setelah EventBridge diaktifkan, semua acara di bawah ini dikirim ke EventBridge. Anda dapat menggunakan EventBridge aturan untuk merutekan acara ke target tambahan. Berikut ini mencantumkan peristiwa yang dikirimkan Amazon S3. EventBridge

Jenis peristiwa	Deskripsi
Objek Dibuat	Sebuah objek telah dibuat. Bidang alasan dalam struktur pesan peristiwa menunjukkan S3 API mana yang digunakan untuk membuat objek: PutObject, POST Object, CopyObject, atau CompleteMultipartUpload.
Objek Dihapus (DeleteObject) Objek Dihapus (Kedaluwarsa siklus hidup)	Sebuah objek telah dihapus. Saat objek dihapus menggunakan panggilan API S3, bidang alasan disetel ke DeleteObject. Saat objek dihapus oleh aturan kedaluwarsa Siklus Hidup S3, bidang alasan disetel ke Kedaluwarsa Siklus Hidup. Untuk informasi selengkapnya, lihat Mengakhiri objek. Ketika objek yang tidak berversi dihapus, atau objek berversi dihapus secara permanen, bidang jenis penghapusan diatur ke Dihapus Secara Permanen. Saat penanda penghapusan dibuat untuk objek versi, bidang jenis penghapusan diatur ke Delete Marker Created. Untuk

Jenis peristiwa	Deskripsi
	informasi selengkapnya, lihat Menghapus versi objek dari bucket dengan dukungan Penentuan Versi.
Pemulihan Objek Dimulai	Pemulihan objek dimulai dari kelas penyimpanan S3 Glacier atau S3 Glacier Deep Archive storage class atau from S3 Intelligent-Tiering Archive Access atau Deep Archive Access tier. Untuk informasi selengkapnya, lihat Bekerja dengan objek yang diarsipkan.
Pemulihan Objek Selesai	Pemulihan objek selesai.
Pemulihan Objek Kedaluwarsa	Salinan sementara dari objek yang dipulihkan dari S3 Glacier atau S3 Glacier Deep Archive telah kedaluwarsa dan dihapus.
Kelas Penyimpanan Objek Berubah	Sebuah objek dialihkan ke kelas penyimpanan yang berbeda. Untuk informasi selengkapnya, lihat Transisi objek menggunakan Siklus Hidup Amazon S3.
Tingkat Akses Objek Berubah	Sebuah objek dialihkan ke tingkat S3 Intelligent-Tiering Archive Access atau tingkat Akses Arsip Dalam. Untuk informasi selengkapnya, lihat Mengelola biaya penyimpanan dengan Amazon S3 Intelligent-Tiering.
Objek ACL Diperbarui	Daftar kontrol akses objek (ACL) ditetapkan menggunakan PutObject ACL. Peristiwa tidak dihasilkan ketika permintaannya tidak menghasilkan perubahan pada ACL objek. Untuk informasi selengkapnya, lihat Gambaran umum daftar kontrol akses (ACL).
Tag Objek Ditambahkan	Satu set tag ditambahkan ke objek menggunakan PutObjectTagging. Untuk informasi selengkapnya, lihat Mengategorikan penyimpanan Anda menggunakan tag.
Tag Objek Dihapus	Semua tag telah dihapus dari objek menggunakan DeleteObjectTagging. Untuk informasi selengkapnya, lihat Mengategorikan penyimpanan Anda menggunakan tag.

> **Note**
>
> Untuk informasi selengkapnya tentang cara memetakan jenis peristiwa Amazon S3 ke jenis EventBridge acara, lihat. EventBridge Pemetaan dan pemecahan masalah Amazon

Anda dapat menggunakan Pemberitahuan Acara Amazon S3 EventBridge untuk menulis aturan yang mengambil tindakan saat peristiwa terjadi di bucket Anda. Misalnya, Anda dapat mengirimi Anda pemberitahuan. Untuk informasi selengkapnya, lihat Apa yang ada EventBridge di Panduan EventBridge Pengguna Amazon.

Untuk informasi selengkapnya tentang tindakan dan tipe data yang dapat berinteraksi dengan menggunakan EventBridge API, lihat Referensi Amazon EventBridge API di Referensi Amazon EventBridge API.

Untuk informasi tentang harga, lihat EventBridge harga Amazon.

Topik

- EventBridge Izin Amazon
- Mengaktifkan Amazon EventBridge
- EventBridge struktur pesan acara
- EventBridge Pemetaan dan pemecahan masalah Amazon

EventBridge Izin Amazon

Amazon S3 tidak memerlukan izin tambahan untuk mengirimkan acara ke Amazon. EventBridge

Mengaktifkan Amazon EventBridge

Anda dapat mengaktifkan Amazon EventBridge menggunakan konsol S3, AWS Command Line Interface (AWS CLI), atau Amazon S3 REST API.

> **Note**
>
> Setelah Anda mengaktifkan EventBridge, dibutuhkan sekitar lima menit agar perubahan diterapkan.

Menggunakan konsol S3

Untuk mengaktifkan pengiriman EventBridge acara di konsol S3.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Dalam daftar bucket, pilih nama bucket tempat Anda ingin mengaktifkan event.
4. Pilih Properti.
5. Arahkan ke bagian Pemberitahuan Acara dan temukan EventBridge subbagian Amazon. Pilih Edit.
6. Di bawah Kirim notifikasi ke Amazon EventBridge untuk semua acara di bucket ini pilih Aktif.

Menggunakan AWS CLI

Contoh berikut membuat konfigurasi notifikasi bucket untuk bucket `amzn-s3-demo-bucket1` dengan Amazon EventBridge diaktifkan.

```
aws s3api put-bucket-notification-configuration --bucket amzn-s3-demo-bucket1 --
notification-configuration='{ "EventBridgeConfiguration": {} }'
```

Penggunaan API REST

Anda dapat mengaktifkan Amazon secara terprogram EventBridge di bucket dengan memanggil Amazon S3 REST API. Untuk informasi selengkapnya, lihat PutBucketNotificationConfigurationdi Referensi API Amazon Simple Storage Service.

Contoh berikut menunjukkan XHTML yang digunakan untuk membuat konfigurasi notifikasi bucket dengan Amazon EventBridge diaktifkan.

```
<NotificationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <EventBridgeConfiguration>
  </EventBridgeConfiguration>
</NotificationConfiguration>
```

Membuat EventBridge aturan

Setelah diaktifkan, Anda dapat membuat EventBridge aturan Amazon untuk tugas-tugas tertentu. Misalnya, Anda dapat mengirim notifikasi email ketika objek dibuat. Untuk tutorial selengkapnya, lihat

Tutorial: Mengirim pemberitahuan saat objek Amazon S3 dibuat di EventBridge Panduan Pengguna Amazon.

EventBridge struktur pesan acara

Pesan pemberitahuan yang dikirimkan Amazon S3 untuk menerbitkan kegiatan ada dalam format JSON. Saat Amazon S3 mengirim acara ke Amazon EventBridge, bidang berikut hadir.

- version—Saat ini adalah 0 (nol) untuk semua peristiwa.
- id — UUID yang dihasilkan untuk setiap acara.
- detail-type—Jenis peristiwa yang sedang dikirim. Lihat Menggunakan EventBridge untuk daftar jenis peristiwa.
- source—Mengidentifikasi layanan yang menghasilkan peristiwa.
- account—ID 12 digit Akun AWS dari pemilik bucket.
- time—Waktu peristiwa terjadi.
- region—Mengidentifikasi Wilayah AWS dari bucket tersebut.
- resources—Array JSON yang berisi Amazon Resource Name (ARN) pada bucket.
- detail—Objek JSON yang berisi informasi tentang peristiwa. Untuk informasi selengkapnya tentang apa saja yang dapat disertakan dalam bidang ini, lihat Kolom detail pesan peristiwa.

Contoh struktur pesan peristiwa

Berikut ini adalah contoh dari beberapa pesan pemberitahuan acara Amazon S3 yang dapat dikirim ke Amazon. EventBridge

Objek dibuat

```
{
  "version": "0",
  "id": "17793124-05d4-b198-2fde-7ededc63b103",
  "detail-type": "Object Created",
  "source": "aws.s3",
  "account": "111122223333",
  "time": "2021-11-12T00:00:00Z",
  "region": "ca-central-1",
  "resources": [
    "arn:aws:s3:::amzn-s3-demo-bucket1"
  ],
```

```
  "detail": {
    "version": "0",
    "bucket": {
      "name": "amzn-s3-demo-bucket1"
    },
    "object": {
      "key": "example-key",
      "size": 5,
      "etag": "b1946ac92492d2347c6235b4d2611184",
      "version-id": "IYV3p45BT0ac8hjHg1houSdS1a.Mro8e",
      "sequencer": "617f08299329d189"
    },
    "request-id": "N4N7GDK58NMKJ12R",
    "requester": "123456789012",
    "source-ip-address": "1.2.3.4",
    "reason": "PutObject"
  }
}
```

Objek dihapus (menggunakan DeleteObject)

```
{
  "version": "0",
  "id": "2ee9cc15-d022-99ea-1fb8-1b1bac4850f9",
  "detail-type": "Object Deleted",
  "source": "aws.s3",
  "account": "111122223333",
  "time": "2021-11-12T00:00:00Z",
  "region": "ca-central-1",
  "resources": [
    "arn:aws:s3:::amzn-s3-demo-bucket1"
  ],
  "detail": {
    "version": "0",
    "bucket": {
      "name": "amzn-s3-demo-bucket1"
    },
    "object": {
      "key": "example-key",
      "etag": "d41d8cd98f00b204e9800998ecf8427e",
      "version-id": "1QW9g1Z99LUNbvaaYVpW9xDlOLU.qxgF",
      "sequencer": "617f0837b476e463"
```

```
    },
    "request-id": "0BH729840619AG5K",
    "requester": "123456789012",
    "source-ip-address": "1.2.3.4",
    "reason": "DeleteObject",
    "deletion-type": "Delete Marker Created"
  }
}
```

Objek dihapus (menggunakan kedaluwarsa siklus hidup)

```
{
  "version": "0",
  "id": "ad1de317-e409-eba2-9552-30113f8d88e3",
  "detail-type": "Object Deleted",
  "source": "aws.s3",
  "account": "111122223333",
  "time": "2021-11-12T00:00:00Z",
  "region": "ca-central-1",
  "resources": [
    "arn:aws:s3:::amzn-s3-demo-bucket1"
  ],
  "detail": {
    "version": "0",
    "bucket": {
      "name": "amzn-s3-demo-bucket1"
    },
    "object": {
      "key": "example-key",
      "etag": "d41d8cd98f00b204e9800998ecf8427e",
      "version-id": "mtB0cV.jejK63XkRNceanNMC.qXPWLeK",
      "sequencer": "617b398000000000"
    },
    "request-id": "20EB74C14654DC47",
    "requester": "s3.amazonaws.com",
    "reason": "Lifecycle Expiration",
    "deletion-type": "Delete Marker Created"
  }
}
```

Pemulihan objek selesai

```
{
  "version": "0",
  "id": "6924de0d-13e2-6bbf-c0c1-b903b753565e",
  "detail-type": "Object Restore Completed",
  "source": "aws.s3",
  "account": "111122223333",
  "time": "2021-11-12T00:00:00Z",
  "region": "ca-central-1",
  "resources": [
    "arn:aws:s3:::amzn-s3-demo-bucket1"
  ],
  "detail": {
    "version": "0",
    "bucket": {
      "name": "amzn-s3-demo-bucket1"
    },
    "object": {
      "key": "example-key",
      "size": 5,
      "etag": "b1946ac92492d2347c6235b4d2611184",
      "version-id": "KKsjUC1.6gIjqtvhfg5AdMI0eCePIiT3"
    },
    "request-id": "189F19CB7FB1B6A4",
    "requester": "s3.amazonaws.com",
    "restore-expiry-time": "2021-11-13T00:00:00Z",
    "source-storage-class": "GLACIER"
  }
}
```

Kolom detail pesan peristiwa

Bidang detail berisi objek JSON dengan informasi tentang peristiwa tersebut. Bidang berikut mungkin ada di bidang detail.

- version—Saat ini adalah 0 (nol) untuk semua peristiwa.
- bucket—Informasi tentang bucket Amazon S3 yang terlibat dalam peristiwa tersebut.
- object—Informasi tentang objek Amazon S3 yang terlibat dalam peristiwa tersebut.
- request-id—ID permintaan dalam respons S3.

- pemohon — Akun AWS ID atau prinsip AWS layanan pemohon.
- source-ip-address— Alamat IP sumber permintaan S3. Hanya terjadi untuk peristiwa yang dipicu oleh permintaan S3.
- alasan - Untuk peristiwa Object Created, API S3 digunakan untuk membuat objek: PutObject, POST Object, CopyObject, atau CompleteMultipartUpload. Untuk peristiwa Object Deleted, ini diatur ke DeleteObjectsaat objek dihapus oleh panggilan API S3, atau Kedaluwarsa Siklus Hidup saat objek dihapus oleh aturan kedaluwarsa Siklus Hidup S3. Untuk informasi selengkapnya, lihat Mengakhiri objek.
- deletion-type—Untuk peristiwa Object Deleted, ketika objek yang tidak berversi dihapus, atau objek berversi dihapus secara permanen, ini diatur ke Permanently Deleted. Saat penanda penghapusan dibuat untuk objek versi, ini diatur ke Hapus Penanda yang Dibuat. Untuk informasi selengkapnya, lihat Menghapus versi objek dari bucket dengan dukungan Penentuan Versi.

  > **Note**
  >
  > Beberapa atribut objek (seperti `etag` dan`size`) hadir hanya ketika penanda hapus dibuat.

- restore-expiry-time— Untuk acara Object Restore Selesai, waktu ketika salinan sementara objek akan dihapus dari S3. Untuk informasi selengkapnya, lihat Bekerja dengan objek yang diarsipkan.
- source-storage-class— Untuk peristiwa Object Restore Inisiated dan Object Restore Completed, kelas penyimpanan objek dipulihkan. Untuk informasi selengkapnya, lihat Bekerja dengan objek yang diarsipkan.
- destination-storage-class— Untuk Object Storage Class Changed event, kelas penyimpanan baru dari objek. Untuk informasi selengkapnya, lihat Transisi objek menggunakan Siklus Hidup Amazon S3.
- destination-access-tier— Untuk acara Object Access Tier Changed, tingkat akses baru objek. Untuk informasi selengkapnya, lihat Mengelola biaya penyimpanan dengan Amazon S3 Intelligent-Tiering.

EventBridge Pemetaan dan pemecahan masalah Amazon

Tabel berikut menjelaskan bagaimana jenis peristiwa Amazon S3 dipetakan ke jenis peristiwa Amazon EventBridge .

Jenis peristiwa S3	Jenis EventBridge detail Amazon
ObjectCreated:Masukan ObjectCreated:Posting ObjectCreated:Salin ObjectCreated:CompleteMultipartUpload	Objek Dibuat
ObjectRemoved:Hapus ObjectRemoved:DeleteMarkerCreated LifecycleExpiration:Hapus LifecycleExpiration:DeleteMarkerCreated	Object Deleted
ObjectRestore:Posting	Pemulihan Objek Dimulai
ObjectRestore:Selesai	Pemulihan Objek Selesai
ObjectRestore:Hapus	Pemulihan Objek Kedaluwarsa
LifecycleTransition	Kelas Penyimpanan Objek Berubah
IntelligentTiering	Tingkat Akses Objek Berubah
ObjectTagging:Masukan	Tag Objek Ditambahkan
ObjectTagging:Hapus	Tag Objek Dihapus
ObjectAcl:Masukan	Objek ACL Diperbarui

EventBridge Pemecahan masalah Amazon

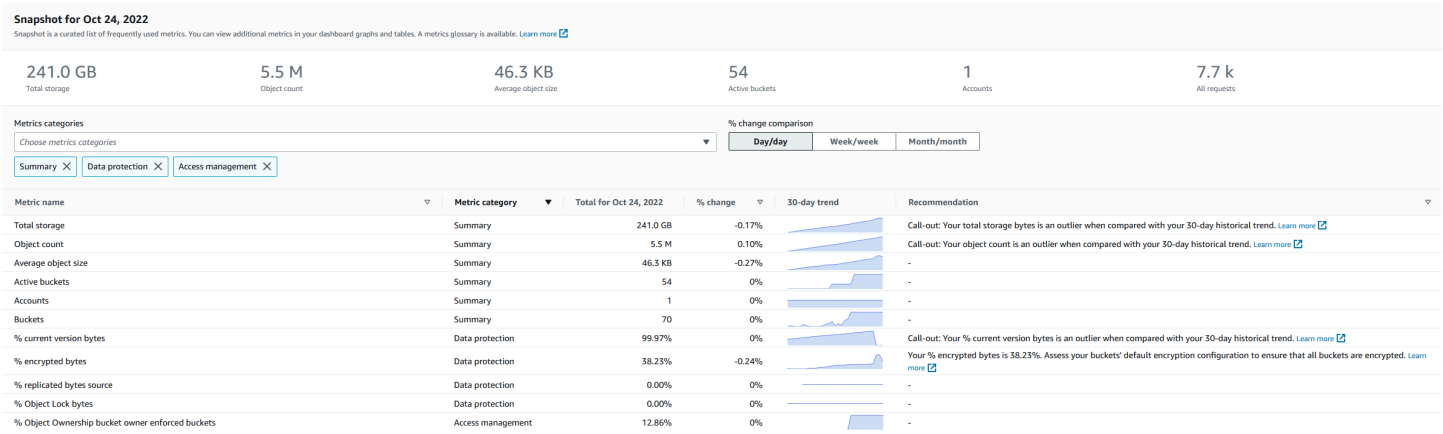
Untuk informasi tentang cara memecahkan masalah EventBridge, lihat Memecahkan Masalah Amazon EventBridge di Panduan Pengguna Amazon EventBridge .

Menilai aktivitas penyimpanan dan penggunaan dengan Lensa Penyimpanan Amazon S3

Lensa Penyimpanan Amazon S3 adalah fitur analisis penyimpanan cloud yang dapat Anda gunakan untuk mendapatkan visibilitas seluruh organisasi ke dalam penggunaan dan aktivitas penyimpanan objek. Lensa Penyimpanan S3 menganalisis metrik untuk memberikan rekomendasi kontekstual yang dapat Anda gunakan untuk mengoptimalkan biaya penyimpanan dan menerapkan praktik terbaik untuk melindungi data Anda.

Anda dapat menggunakan metrik Lensa Penyimpanan S3 untuk menghasilkan wawasan ringkasan. Misalnya, Anda dapat mengetahui berapa banyak penyimpanan yang dimiliki di seluruh organisasi atau bucket dan prefiks mana yang paling cepat berkembang. Anda juga dapat menggunakan metrik Lensa Penyimpanan S3 untuk mengidentifikasi peluang optimasi biaya, menerapkan praktik terbaik untuk perlindungan data dan keamanan, serta meningkatkan kinerja beban kerja aplikasi. Misalnya, Anda dapat mengidentifikasi bucket yang tidak memiliki aturan Siklus Hidup S3 yang disiapkan untuk mengakhiri unggahan multibagian yang tidak lengkap yang berumur lebih dari 7 hari. Anda juga dapat mengidentifikasi bucket yang tidak mengikuti praktik terbaik perlindungan data, seperti menggunakan Replikasi S3 atau Penentuan Versi S3.

Lensa Penyimpanan S3 menggabungkan metrik Anda dan menampilkan informasi di bagian Snapshot akun di halaman Bucket konsol Amazon S3. Lensa Penyimpanan S3 juga menyediakan dasbor interaktif yang dapat Anda gunakan untuk memvisualisasikan wawasan dan tren, menandai outlier, serta menerima rekomendasi untuk mengoptimalkan biaya penyimpanan dan menerapkan praktik terbaik perlindungan data. Dasbor Anda memiliki opsi perincian untuk menghasilkan dan memvisualisasikan wawasan di tingkat organisasi, akun, Wilayah AWS, kelas penyimpanan, bucket, prefiks, atau grup Lensa Penyimpanan. Anda juga dapat mengirimkan ekspor metrik harian dalam format CSV atau Parquet ke bucket S3.



Metrik dan fitur Lensa Penyimpanan S3

Lensa Penyimpanan S3 menyediakan dasbor default interaktif yang diperbarui setiap hari. Lensa Penyimpanan S3 mengonfigurasi dasbor ini sebelumnya untuk memvisualisasikan ringkasan wawasan dan tren untuk seluruh akun dan memperbaruinya setiap hari di konsol S3. Metrik dari dasbor ini juga diringkas di dalam snapshot akun Anda di halaman Bucket. Untuk informasi selengkapnya, lihat Dasbor default.

Untuk membuat dasbor lain dan mencakupnya berdasarkan Wilayah AWS, bucket S3, atau akun (untuk AWS Organizations), Anda membuat konfigurasi dasbor Lensa Penyimpanan S3. Anda dapat membuat dan mengelola konfigurasi dasbor S3 Storage Lens dengan menggunakan konsol Amazon S3 AWS Command Line Interface ,AWS CLI() AWS SDKs, atau Amazon S3 REST API. Saat membuat atau mengedit dasbor Lensa Penyimpanan S3, tentukan cakupan dasbor dan pemilihan metrik.

Lensa Penyimpanan S3 menawarkan metrik gratis dan metrik dan rekomendasi lanjutan, yang dapat ditingkatkan dengan biaya tambahan. Dengan metrik dan rekomendasi lanjutan, Anda dapat mengakses metrik dan fitur tambahan untuk mendapatkan wawasan tentang penyimpanan Anda. Fitur-fitur ini mencakup kategori metrik lanjutan, agregasi awalan, rekomendasi kontekstual, dan penerbitan Amazon. CloudWatch Agregasi prefiks dan rekomendasi kontekstual hanya tersedia di konsol Amazon S3. Untuk informasi tentang harga Lensa Penyimpanan S3, lihat harga Amazon S3.

Kategori metrik

Di tingkatan gratis dan lanjutan, metrik diatur ke dalam kategori yang selaras dengan kasus penggunaan utama, seperti optimisasi biaya dan perlindungan data. Metrik gratis mencakup ringkasan, optimisasi biaya, perlindungan data, manajemen akses, kinerja, dan metrik acara. Saat meningkatkan ke metrik dan rekomendasi lanjutan, Anda dapat mengaktifkan metrik optimisasi biaya dan perlindungan data lanjutan. Anda dapat menggunakan metrik canggih ini untuk lebih mengurangi biaya penyimpanan S3 dan meningkatkan sikap perlindungan data Anda. Anda juga dapat mengaktifkan metrik aktivitas dan metrik kode status terperinci untuk meningkatkan kinerja beban kerja aplikasi yang mengakses bucket S3. Untuk informasi selengkapnya tentang kategori metrik gratis dan tingkat lanjut, lihat Pilihan metrik.

Anda dapat menilai penyimpanan berdasarkan praktik terbaik S3, seperti menganalisis persentase bucket yang mengaktifkan enkripsi atau Kunci Objek S3 atau Penentuan Versi S3. Anda juga dapat mengidentifikasi peluang penghematan biaya potensial. Misalnya, Anda dapat menggunakan metrik jumlah aturan Siklus Hidup S3 untuk mengidentifikasi bucket yang tidak memiliki aturan kedaluwarsa

atau transisi siklus hidup. Anda juga dapat menganalisis aktivitas permintaan per bucket untuk menemukan bucket tempat objek dapat dialihkan ke kelas penyimpanan dengan biaya yang lebih rendah. Untuk informasi selengkapnya, lihat Kasus penggunaan metrik Lensa Penyimpanan Amazon S3.

Ekspor metrik

Selain melihat dasbor di konsol S3, Anda dapat mengekspor metrik dalam format CSV atau format Parquet ke bucket S3 untuk analisis lebih lanjut dengan alat analisis pilihan Anda. Untuk informasi selengkapnya, lihat Melihat metrik Lensa Penyimpanan Amazon S3 menggunakan ekspor data.

CloudWatch Penerbitan Amazon

Anda dapat memublikasikan metrik penggunaan dan aktivitas Lensa Penyimpanan S3 CloudWatch ke Amazon untuk membuat tampilan terpadu kesehatan operasional Anda di dasbor. CloudWatch Anda juga dapat menggunakan CloudWatch fitur, seperti alarm dan tindakan yang dipicu, matematika metrik, dan deteksi anomali, untuk memantau dan mengambil tindakan pada metrik Lensa Penyimpanan S3. Selain itu, operasi CloudWatch API memungkinkan aplikasi, termasuk penyedia pihak ketiga, untuk mengakses metrik Lensa Penyimpanan S3 Anda. Opsi CloudWatch penerbitan tersedia untuk dasbor yang ditingkatkan ke metrik dan rekomendasi lanjutan S3 Storage Lens. Untuk informasi selengkapnya tentang dukungan untuk metrik Lensa Penyimpanan S3 di CloudWatch, lihat. Pantau metrik Lensa Penyimpanan S3 di CloudWatch

Untuk informasi selengkapnya tentang penggunaan Lensa Penyimpanan S3, lihat topik berikut.

Topik

- Memahami Lensa Penyimpanan Amazon S3
- Glosarium metrik Lensa Penyimpanan Amazon S3
- Menyetel izin Lensa Penyimpanan Amazon S3
- Bekerja dengan Lensa Penyimpanan Amazon S3 menggunakan konsol dan API
- Melihat metrik dengan Lensa Penyimpanan Amazon S3
- Menggunakan Lensa Penyimpanan Amazon S3 dengan AWS Organizations
- Bekerja dengan grup Lensa Penyimpanan S3 untuk memfilter dan mengumpulkan metrik

Memahami Lensa Penyimpanan Amazon S3

> **Important**
>
> Amazon S3 sudah menerapkan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3) sebagai tingkat dasar enkripsi bagi setiap bucket di Amazon S3. Mulai 5 Januari 2023, semua unggahan objek baru ke Amazon S3 secara otomatis akan dienkripsi tanpa biaya tambahan dan tidak akan berdampak pada kinerja. Status enkripsi otomatis untuk konfigurasi enkripsi default bucket S3 dan untuk unggahan objek baru tersedia di AWS CloudTrail log, S3 Inventory, S3 Storage Lens, konsol Amazon S3, dan sebagai header respons API Amazon S3 tambahan di dan. AWS Command Line Interface AWS SDKs Untuk informasi selengkapnya, lihat FAQ enkripsi default.

Lensa Penyimpanan Amazon S3 adalah fitur analisis penyimpanan cloud yang dapat Anda gunakan untuk mendapatkan visibilitas seluruh organisasi ke dalam penggunaan dan aktivitas penyimpanan objek. Anda dapat menggunakan metrik Lensa Penyimpanan S3 untuk menghasilkan wawasan ringkasan, seperti mencari tahu berapa banyak penyimpanan yang Anda miliki di seluruh organisasi atau bucket dan prefiks yang paling cepat berkembang. Anda juga dapat menggunakan metrik Lensa Penyimpanan S3 untuk mengidentifikasi peluang optimasi biaya, menerapkan praktik terbaik untuk perlindungan data dan keamanan, serta meningkatkan kinerja beban kerja aplikasi. Misalnya, Anda dapat mengidentifikasi bucket yang tidak memiliki aturan Siklus Hidup S3 untuk mengakhiri unggahan multibagian yang tidak lengkap dan berusia lebih dari 7 hari. Anda juga dapat mengidentifikasi bucket yang tidak mengikuti praktik terbaik perlindungan data, seperti menggunakan Replikasi S3 atau Penentuan Versi S3. Lensa Penyimpanan S3 menganalisis metrik untuk memberikan rekomendasi kontekstual yang dapat Anda gunakan untuk mengoptimalkan biaya penyimpanan dan menerapkan praktik terbaik untuk melindungi data Anda.

Lensa Penyimpanan S3 menggabungkan metrik Anda dan menampilkan informasi di bagian Snapshot akun di halaman Bucket konsol Amazon S3. Lensa Penyimpanan S3 juga menyediakan dasbor interaktif yang dapat Anda gunakan untuk memvisualisasikan wawasan dan tren, menandai outlier, serta menerima rekomendasi untuk mengoptimalkan biaya penyimpanan dan menerapkan praktik terbaik perlindungan data. Dasbor Anda memiliki opsi perincian untuk menghasilkan dan memvisualisasikan wawasan di tingkat organisasi, akun, Wilayah AWS, kelas penyimpanan, bucket, prefiks, atau grup Lensa Penyimpanan. Anda juga dapat mengirimkan ekspor metrik harian dalam format CSV atau Parquet ke bucket S3. Anda dapat membuat dan mengelola dasbor Lensa

Penyimpanan S3 menggunakan konsol Amazon S3 AWS Command Line Interface ,AWS CLI() AWS SDKs, atau Amazon S3 REST API.

Konsep dan terminologi Lensa Penyimpanan S3

Bagian ini berisi terminologi dan konsep yang penting agar berhasil memahami dan menggunakan Lensa Penyimpanan Amazon S3.

Topik

- Konfigurasi dasbor
- Dasbor default
- Dasbor
- Snapshot akun
- Ekspor metrik
- Wilayah Asal
- Periode penahanan
- Kategori metrik
- Rekomendasi
- Pilihan metrik
- Lensa Penyimpanan S3 dan AWS Organizations

Konfigurasi dasbor

Lensa Penyimpanan S3 membutuhkan konfigurasi dasbor yang berisi properti yang diperlukan untuk menggabungkan metrik atas nama Anda untuk satu dasbor atau ekspor. Saat Anda membuat konfigurasi, pilih nama dasbor dan Wilayah asal yang tidak dapat diubah setelah membuat dasbor. Anda dapat menambahkan tag secara opsional dan mengonfigurasi ekspor metrik dalam format CSV atau Parquet.

Di konfigurasi dasbor, tentukan cakupan dasbor dan pemilihan metrik. Cakupan dapat menyertakan semua penyimpanan untuk akun organisasi atau bagian yang difilter berdasarkan Wilayah, bucket, dan akun. Saat mengonfigurasi pemilihan metrik, pilih antara metrik gratis dan metrik dan rekomendasi lanjutan, yang dapat ditingkatkan dengan biaya tambahan. Dengan metrik dan rekomendasi lanjutan, Anda dapat mengakses metrik dan fitur tambahan. Fitur-fitur ini mencakup

kategori metrik lanjutan, agregasi tingkat awalan, rekomendasi kontekstual, dan penerbitan Amazon. CloudWatch Untuk informasi tentang harga Lensa Penyimpanan S3, lihat harga Amazon S3.

Dasbor default

Dasbor default S3 Storage Lens pada konsol diberi nama default-account-dashboard. S3 mengonfigurasi dasbor ini sebelumnya untuk memvisualisasikan ringkasan wawasan dan tren untuk seluruh akun dan memperbaruinya setiap hari di konsol S3. Anda tidak dapat mengubah cakupan konfigurasi dari dasbor default, namun Anda dapat meningkatkan pilihan metrik dari metrik gratis ke metrik dan rekomendasi lanjutan. Anda dapat mengonfigurasi ekspor metrik opsional, atau bahkan menonaktifkan dasbor. Namun, Anda tidak dapat menghapus dasbor default.

> **Note**
>
> Jika Anda menonaktifkan dasbor default, dasbor itu tidak akan diperbarui lagi. Anda tidak akan lagi menerima metrik harian baru di dasbor Lensa Penyimpanan S3, ekspor metrik, atau snapshot akun di laman Bucket S3. Jika dasbor Anda menggunakan metrik dan rekomendasi lanjutan, Anda tidak akan dikenakan biaya lagi. Anda masih dapat melihat data historis di dasbor hingga periode 14 hari untuk kueri data kedaluwarsa. Periodenya adalah 15 bulan jika Anda telah mengaktifkan metrik dan rekomendasi lanjutan. Untuk mengakses data historis, Anda dapat mengaktifkan kembali dasbor di periode kedaluwarsa.

Dasbor

Anda dapat membuat dasbor Lensa Penyimpanan S3 tambahan dan mencakupnya berdasarkan Wilayah AWS, bucket S3, atau akun (untuk). AWS Organizations Saat membuat atau mengedit dasbor Lensa Penyimpanan S3, tentukan cakupan dasbor dan pemilihan metrik. Lensa Penyimpanan S3 menawarkan metrik gratis dan metrik dan rekomendasi lanjutan, yang dapat ditingkatkan dengan biaya tambahan. Dengan metrik dan rekomendasi lanjutan, Anda dapat mengakses metrik dan fitur tambahan untuk mendapatkan wawasan tentang penyimpanan Anda. Ini termasuk kategori metrik lanjutan, agregasi tingkat awalan, rekomendasi kontekstual, dan penerbitan Amazon. CloudWatch Untuk informasi tentang harga Lensa Penyimpanan S3, lihat harga Amazon S3.

Anda juga dapat menonaktifkan atau menghapus dasbor. Jika dasbor dinonaktifkan, dasbor tidak lagi diperbarui, dan Anda tidak akan lagi menerima metrik harian baru. Anda masih dapat melihat data historis hingga masa kedaluwarsa 14 hari. Jika metrik dan rekomendasi lanjutan diaktifkan untuk dasbor tersebut, periodenya adalah 15 bulan. Untuk mengakses data historis, Anda dapat mengaktifkan kembali dasbor di periode kedaluwarsa.

Jika Anda menghapus dasbor, Anda akan kehilangan semua pengaturan konfigurasi dasbor. Anda tidak akan lagi menerima metrik harian baru, dan Anda juga akan kehilangan akses ke data historis yang terkait dengan dasbor tersebut. Jika Anda ingin mengakses data historis untuk dasbor yang telah dihapus, Anda harus membuat dasbor lain dengan nama yang sama di Wilayah asal yang sama.

> **Note**
>
> - Anda dapat menggunakan Lensa Penyimpanan S3 untuk membuat hingga 50 dasbor per Wilayah asal.
> - Dasbor tingkat organisasi hanya dapat dibatasi untuk lingkup Regional saja.

Snapshot akun

Snapshot Akun Lensa Penyimpanan S3 meringkas metrik dari dasbor default dan menampilkan total penyimpanan, jumlah objek, dan ukuran objek rata-rata di laman Bucket konsol S3. Snapshot akun ini memberikan akses cepat ke wawasan tentang penyimpanan tanpa harus meninggalkan laman Bucket. Snapshot akun juga menyediakan akses sekali klik ke dasbor Lensa Penyimpanan S3 interaktif.

Anda dapat menggunakan dasbor untuk memvisualisasikan wawasan dan tren, menandai outlier, dan menerima rekomendasi untuk mengoptimalkan biaya penyimpanan dan menerapkan praktik terbaik untuk perlindungan data. Dasbor memiliki opsi penelusuran untuk menghasilkan wawasan di tingkat organisasi, akun, bucket, objek, atau awalan. Anda juga dapat mengirim ekspor metrik sekali sehari ke bucket S3 dalam format CSV atau Parquet.

Anda tidak dapat mengubah cakupan dasbor default-account karena dasbor tersebut tertaut ke snapshot Akun. Namun, Anda dapat meningkatkan pilihan metrik default-account-dashboarddari metrik gratis ke metrik dan rekomendasi lanjutan berbayar. Setelah ditingkatkan, Anda kemudian dapat menampilkan semua permintaan, byte yang diunggah, dan byte yang diunduh di snapshot Akun Lensa Penyimpanan S3.

> **Note**
>
> Jika Anda menonaktifkan dasbor default, snapshot Akun Anda tidak akan lagi diperbarui. Untuk terus menampilkan metrik di snapshot Akun, Anda dapat mengaktifkan kembali metrik. default-account-dashboard

Ekspor metrik

Ekspor metrik Lensa Penyimpanan S3 adalah file yang berisi semua metrik yang diidentifikasi di dalam konfigurasi Lensa Penyimpanan S3. Informasi ini dihasilkan setiap hari dalam format CSV atau Parquet dan dikirim ke bucket S3. Anda dapat menggunakan ekspor metrik untuk analisis lebih lanjut dengan menggunakan alat metrik pilihan Anda. Bucket S3 untuk ekspor metrik harus berada di Wilayah yang sama dengan konfigurasi Lensa Penyimpanan S3 Anda. Anda dapat menghasilkan ekspor metrik Lensa Penyimpanan S3 dari konsol S3 dengan mengedit konfigurasi dasbor. Anda juga dapat mengonfigurasi ekspor metrik dengan menggunakan AWS CLI dan AWS SDKs.

Wilayah Asal

Wilayah rumah adalah Wilayah AWS tempat semua metrik Lensa Penyimpanan S3 untuk konfigurasi dasbor tertentu disimpan. Anda harus memilih Wilayah asal ketika membuat konfigurasi dasbor Lensa Penyimpanan S3. Setelah memilih Wilayah asal, Anda tidak dapat mengubahnya. Selain itu, jika Anda membuat grup Lensa Penyimpanan, sebaiknya pilih Wilayah asal yang sama dengan dasbor Lensa Penyimpanan Anda.

> **Note**
>
> Anda dapat memilih salah satu Wilayah berikut sebagai Wilayah asal:
>
> - AS Timur (Virginia Utara)–`us-east-1`
> - AS Timur (Ohio)–`us-east-2`
> - AS Barat (California Utara)–`us-west-1`
> - AS Barat (Oregon)–`us-west-2`
> - Asia Pasifik (Mumbai)–`ap-south-1`
> - Asia Pasifik (Seoul)–`ap-northeast-2`
> - Asia Pasifik (Singapura)–`ap-southeast-1`
> - Asia Pasifik (Sydney)–`ap-southeast-2`
> - Asia Pasifik (Tokyo)–`ap-northeast-1`
> - Kanada (Pusat)–`ca-central-1`
> - Tiongkok (Beijing)–`cn-north-1`
> - Tiongkok (Ningxia)–`cn-northwest-1`
> - Eropa (Frankfurt)–`eu-central-1`
> - Eropa (Irlandia)–`eu-west-1`

- Eropa (London)–`eu-west-2`
- Eropa (Paris)–`eu-west-3`
- Eropa (Stockholm)–`eu-north-1`
- Amerika Selatan (Sao Paulo)–`sa-east-1`

Periode penahanan

Metrik Lensa Penyimpanan S3 dipertahankan sehingga Anda dapat melihat tren historis dan membandingkan perbedaan di penyimpanan dan aktivitas dari waktu ke waktu. Anda dapat menggunakan metrik Lensa Penyimpanan Amazon S3 untuk kueri sehingga Anda dapat melihat tren historis dan membandingkan perbedaan dalam penggunaan dan aktivitas penyimpanan dari waktu ke waktu.

Semua metrik Lensa Penyimpanan S3 dipertahankan untuk jangka waktu 15 bulan. Namun, metrik hanya tersedia untuk kueri selama durasi tertentu, yang bergantung pada pilihan metrik. Durasi ini tidak dapat diubah. Metrik gratis tersedia untuk kueri selama periode 14 hari, dan metrik lanjutan tersedia untuk kueri selama periode 15 bulan.

Kategori metrik

Di tingkatan gratis dan lanjutan, metrik Lensa Penyimpanan S3 dikelola ke dalam kategori yang selaras dengan kasus penggunaan utama, seperti pengoptimalan biaya dan perlindungan data. Metrik gratis mencakup ringkasan, optimisasi biaya, perlindungan data, manajemen akses, kinerja, dan metrik acara. Saat meningkatkan ke metrik dan rekomendasi lanjutan, Anda dapat mengaktifkan metrik pengoptimalan biaya dan perlindungan data tambahan yang dapat digunakan untuk lebih mengurangi biaya penyimpanan S3 dan memastikan data terlindungi. Anda juga dapat mengaktifkan metrik aktivitas dan metrik kode status terperinci yang dapat digunakan untuk meningkatkan kinerja alur kerja aplikasi.

Daftar berikut menunjukkan semua kategori metrik gratis dan metrik lanjutan. Untuk daftar lengkap metrik individual yang disertakan di dalam setiap kategori, lihat Glosarium metrik.

Metrik ringkasan

Metrik ringkasan memberikan wawasan umum tentang penyimpanan S3, termasuk total byte penyimpanan dan jumlah objek.

Metrik optimisasi biaya

Metrik optimisasi biaya memberikan wawasan yang dapat digunakan untuk mengelola dan mengoptimalkan biaya penyimpanan. Misalnya, Anda dapat mengidentifikasi bucket yang memiliki unggahan multibagian yang tidak lengkap dan berusia lebih dari 7 hari.

Dengan metrik dan rekomendasi lanjutan, Anda dapat mengaktifkan metrik optimisasi biaya lanjutan. Metrik ini mencakup metrik jumlah aturan Siklus Hidup S3 yang dapat digunakan untuk mendapatkan kedaluwarsa per bucket dan jumlah aturan Siklus Hidup S3 transisi.

Metrik perlindungan data

Metrik perlindungan data memberikan wawasan untuk fitur perlindungan data, seperti enkripsi dan Penentuan Versi S3. Anda dapat menggunakan metrik ini untuk mengidentifikasi bucket yang tidak mengikuti praktik terbaik untuk perlindungan data. Misalnya, Anda dapat mengidentifikasi bucket yang tidak menggunakan enkripsi default dengan AWS Key Management Service kunci (SSE-KMS) atau S3 Versioning.

Dengan metrik dan rekomendasi lanjutan, Anda dapat mengaktifkan metrik perlindungan data lanjutan. Metrik ini mencakup metrik jumlah aturan replikasi per bucket.

Metrik manajemen akses

Metrik manajemen akses memberikan wawasan untuk Objek Kepemilikan S3. Anda dapat menggunakan metrik ini untuk melihat pengaturan Objek Kepemilikan yang digunakan oleh bucket.

Metrik peristiwa

Metrik peristiwa memberikan wawasan untuk Pemberitahuan Peristiwa S3. Dengan metrik peristiwa, Anda dapat melihat bucket mana yang memiliki Pemberitahuan Peristiwa S3 yang dikonfigurasi.

Metrik kinerja

Metrik kinerja memberikan wawasan untuk Percepatan Transfer S3. Dengan metrik kinerja, Anda dapat melihat bucket mana yang mengaktifkan Percepatan Transfer.

Metrik aktivitas (lanjutan)

Jika Anda meningkatkan dasbor ke Metrik dan rekomendasi lanjutan, Anda dapat mengaktifkan metrik aktivitas. Metrik aktivitas memberikan detail tentang cara penyimpanan diminta (misalnya, Semua permintaan, Dapatkan permintaan, Masukkan permintaan), Byte yang diunggah atau diunduh, dan kesalahan.

Metrik aktivitas tingkat awalan dapat digunakan untuk membantu menentukan awalan mana yang jarang digunakan, sehingga Anda dapat beralih ke kelas penyimpanan yang lebih optimal dengan menggunakan Siklus Hidup S3.

Metrik kode status terperinci (lanjutan)

Jika Anda meningkatkan dasbor ke Metrik dan rekomendasi lanjutan, Anda dapat mengaktifkan metrik kode status terperinci. Metrik kode status terperinci memberikan wawasan untuk kode status HTTP, seperti 403 Forbidden dan 503 Service Unavailable, yang dapat digunakan untuk memecahkan masalah akses atau kinerja. Misalnya, Anda dapat melihat metrik jumlah kesalahan 403 Forbidden untuk mengidentifikasi beban kerja yang mengakses bucket dengan izin yang tidak sesuai.

Metrik kode status terperinci tingkat awalan dapat digunakan untuk mendapatkan pemahaman yang lebih baik tentang kemunculan kode status HTTP berdasarkan awalan. Misalnya, 503 metrik jumlah kesalahan memungkinkan Anda mengidentifikasi awalan yang menerima permintaan throttling selama penyerapan data.

Rekomendasi

Lensa Penyimpanan S3 memberikan rekomendasi otomatis untuk membantu Anda mengoptimalkan penyimpanan. Rekomendasi ditempatkan secara kontekstual bersama metrik yang relevan di dasbor Lensa Penyimpanan S3. Data historis tidak memenuhi syarat untuk rekomendasi karena rekomendasi akan relevan dengan apa yang terjadi di periode terbaru. Rekomendasi hanya muncul jika relevan.

Rekomendasi Lensa Penyimpanan S3 tersedia dalam bentuk berikut:

- Saran

  Saran memperingatkan tentang tren dalam penyimpanan dan aktivitas yang mungkin menunjukkan peluang optimisasi biaya penyimpanan atau praktik terbaik untuk perlindungan data. Anda dapat menggunakan topik yang disarankan di Panduan Pengguna Amazon S3 dan dasbor Lensa Penyimpanan S3 untuk menelusuri detail selengkapnya tentang Wilayah, bucket, atau awalan tertentu.
- Panggilan

  Panggilan adalah rekomendasi yang memperingatkan tentang anomali yang menarik dalam penyimpanan dan aktivitas selama periode yang mungkin memerlukan perhatian atau pemantauan lebih lanjut.

- Panggilan outlier

  Lensa Penyimpanan S3 menyediakan panggilan untuk metrik outlier, berdasarkan tren 30 hari terakhir. Outlier dihitung dengan menggunakan skor standar, yang juga dikenal sebagai skor z. Di skor ini, metrik terkini akan dikurangi dari rata-rata 30 hari terakhir untuk metrik tersebut. Metrik terkini kemudian dibagi dengan standar deviasi untuk metrik tersebut selama 30 hari terakhir. Skor yang dihasilkan biasanya antara -3 dan +3. Angka ini mewakili jumlah standar deviasi metrik terkini dari nilai rata-rata.

  Lensa Penyimpanan S3 menganggap metrik dengan skor >2 atau <-2 sebagai outlier karena nilai tersebut lebih tinggi atau lebih rendah dari 95 persen data terdistribusi normal.

- Panggilan perubahan signifikan

  Panggilan perubahan signifikan berlaku untuk metrik yang diperkirakan akan lebih jarang berubah. Oleh karena itu, panggilan ini diatur ke sensitivitas yang lebih tinggi daripada perhitungan outlier, yang biasanya berada pada kisaran +/- 20 persen dibandingkan dengan hari, minggu, atau bulan sebelumnya.

  Mengatasi panggilan di penyimpanan dan aktivitas–Jika Anda menerima panggilan perubahan signifikan, hal tersebut belum tentu menjadi masalah. Panggilan tersebut bisa menjadi hasil dari perubahan yang diantisipasi di dalam penyimpanan. Misalnya, Anda mungkin baru saja menambahkan objek baru dalam jumlah besar, menghapus objek dalam jumlah besar, atau membuat perubahan terencana yang serupa.

  Jika Anda melihat panggilan perubahan signifikan di dasbor, perhatikan dan tentukan apakah panggilan tersebut dapat dijelaskan oleh keadaan terkini. Jika tidak, gunakan dasbor Lensa Penyimpanan S3 untuk menelusuri detail selengkapnya guna memahami Wilayah, bucket, atau awalan tertentu yang mendorong fluktuasi.

- Pengingat

  Pengingat memberikan wawasan tentang cara kerja Amazon S3. Pengingat dapat membantu mempelajari lebih lanjut tentang cara menggunakan fitur S3 untuk mengurangi biaya penyimpanan atau menerapkan praktik terbaik untuk perlindungan data.

Pilihan metrik

Lensa Penyimpanan S3 menawarkan dua pilihan metrik yang dapat dipilih untuk dasbor dan ekspor Anda: metrik gratis dan metrik dan rekomendasi lanjutan.

- Metrik gratis

  Lensa Penyimpanan S3 menawarkan metrik gratis untuk semua dasbor dan konfigurasi. Metrik gratis berisi metrik yang relevan dengan penyimpanan, seperti jumlah bucket dan objek di akun Anda. Metrik gratis juga menyertakan metrik berbasis kasus penggunaan (misalnya, optimalisasi biaya dan metrik perlindungan data) yang dapat digunakan untuk menyelidiki apakah penyimpanan dikonfigurasi sesuai dengan praktik terbaik S3. Semua metrik gratis dikumpulkan setiap hari. Data akan tersedia untuk kueri selama 14 hari. Untuk informasi selengkapnya tentang metrik mana yang tersedia dengan metrik gratis, lihat Glosarium metrik Lensa Penyimpanan Amazon S3.

- Metrik dan rekomendasi lanjutan

  Lensa Penyimpanan S3 menawarkan metrik gratis untuk semua dasbor dan konfigurasi dengan opsi untuk meningkatkan ke metrik dan rekomendasi lanjutan. Biaya tambahan berlaku. Untuk informasi selengkapnya, lihat Harga Amazon S3.

  Metrik dan rekomendasi lanjutan mencakup semua metrik di metrik gratis bersama dengan metrik tambahan, seperti metrik perlindungan data dan optimalisasi biaya lanjutan, metrik aktivitas, dan metrik kode status terperinci. Metrik dan rekomendasi lanjutan juga memberikan rekomendasi untuk membantu mengoptimalkan penyimpanan. Rekomendasi ditempatkan secara kontekstual di samping metrik yang relevan di dasbor.

  Metrik dan rekomendasi lanjutan mencakup fitur-fitur berikut:

  - Metrik lanjutan–Menghasilkan metrik tambahan. Untuk daftar lengkap kategori metrik lanjutan, lihat Kategori metrik. Untuk daftar metrik lengkap, lihat Glosarium metrik Lensa Penyimpanan Amazon S3.
  - Amazon CloudWatch publishing — Menerbitkan metrik Lensa Penyimpanan S3 CloudWatch untuk membuat tampilan terpadu kesehatan operasional Anda di dasbor. CloudWatch Anda juga dapat menggunakan operasi dan fitur CloudWatch API, seperti alarm dan tindakan yang dipicu, matematika metrik, dan deteksi anomali, untuk memantau dan mengambil tindakan pada metrik Lensa Penyimpanan S3. Untuk informasi selengkapnya, lihat Pantau metrik Lensa Penyimpanan S3 di CloudWatch.
  - Agregasi awalan–Mengumpulkan metrik di tingkat awalan. Mengaktifkan agregasi awalan dapat memperluas semua metrik yang disertakan di dalam konfigurasi dasbor di tingkat awalan. Metrik hanya dapat dibuat untuk awalan yang memenuhi ambang batas yang dikonfigurasi. Perhatikan bahwa metrik yang berlaku pada tingkat awalan tersedia dengan Agregasi awalan, kecuali untuk pengaturan tingkat bucket dan metrik jumlah aturan. Metrik tingkat awalan tidak dipublikasikan ke. CloudWatch

- Agregasi grup Lensa Penyimpanan–Mengumpulkan metrik di tingkat grup Lensa Penyimpanan. Setelah mengaktifkan Metrik dan rekomendasi lanjutan serta Agregasi grup Lensa Penyimpanan, Anda dapat menentukan grup Lensa Penyimpanan yang akan disertakan atau dikecualikan dari dasbor Lensa Penyimpanan. Setidaknya satu grup Lensa Penyimpanan harus ditentukan. Grup Lensa Penyimpanan yang ditentukan juga harus berada di Wilayah asal yang ditunjuk di akun dasbor. Metrik tingkat grup Lensa Penyimpanan tidak dipublikasikan ke. CloudWatch

Semua metrik lanjutan akan dikumpulkan setiap hari. Data tersedia untuk kueri hingga 15 bulan. Untuk informasi selengkapnya tentang metrik penyimpanan yang dikumpulkan oleh Lensa Penyimpanan S3, lihat Glosarium metrik Lensa Penyimpanan Amazon S3.

> **Note**
>
> Rekomendasi hanya akan tersedia bila Anda menggunakan dasbor Lensa Penyimpanan S3 di konsol Amazon S3.

Lensa Penyimpanan S3 dan AWS Organizations

AWS Organizations adalah Layanan AWS yang membantu Anda menggabungkan semua hierarki Akun AWS di bawah satu organisasi Anda. Amazon S3 Storage Lens berfungsi AWS Organizations untuk menyediakan satu tampilan penyimpanan objek dan aktivitas di seluruh penyimpanan Amazon S3 Anda.

Untuk informasi selengkapnya, lihat Menggunakan Lensa Penyimpanan Amazon S3 dengan AWS Organizations.

- Akses tepercaya

  Dengan menggunakan akun manajemen organisasi, Anda harus mengaktifkan akses tepercaya untuk Lensa Penyimpanan S3 guna menggabungkan metrik penyimpanan dan data penggunaan untuk semua akun anggota di organisasi. Kemudian, Anda dapat membuat dasbor atau ekspor untuk organisasi dengan menggunakan akun manajemen atau dengan memberikan akses administrator yang didelegasikan ke akun lain di organisasi.

  Anda dapat menonaktifkan akses tepercaya untuk Lensa Penyimpanan S3 kapan saja, yang menghentikan Lensa Penyimpanan S3 dari tindakan menggabungkan metrik untuk organisasi.

- Administrator yang didelegasikan

Anda dapat membuat dasbor dan metrik untuk S3 Storage Lens untuk organisasi Anda dengan menggunakan akun AWS Organizations manajemen Anda, atau dengan memberikan akses administrator yang didelegasikan ke akun lain di organisasi Anda. Anda dapat membatalkan pendaftaran administrator yang didelegasikan kapan saja. Membatalkan pendaftaran administrator yang didelegasikan juga secara otomatis akan menghentikan semua dasbor tingkat organisasi yang dibuat oleh administrator yang didelegasikan tersebut untuk menggabungkan metrik penyimpanan baru.

Untuk informasi lebih lanjut, lihat Lensa Penyimpanan Amazon S3 dan AWS Organizations di Panduan Pengguna AWS Organizations .

Peran terkait layanan Lensa Penyimpanan Amazon S3

Seiring dengan akses AWS Organizations tepercaya, Amazon S3 Storage Lens menggunakan peran terkait AWS Identity and Access Management layanan (IAM). Peran terkait layanan adalah tipe peran IAM unik yang ditautkan langsung ke Lensa Penyimpanan S3. Peran terkait layanan telah ditentukan sebelumnya oleh Lensa Penyimpanan S3 dan menyertakan semua izin yang diperlukan untuk mengumpulkan metrik penggunaan dan aktivitas penyimpanan harian dari akun anggota di organisasi.

Untuk informasi selengkapnya, lihat Menggunakan peran terkait layanan untuk Lensa Penyimpanan Amazon S3.

Glosarium metrik Lensa Penyimpanan Amazon S3

Glosarium metrik Lensa Penyimpanan Amazon S3 menyediakan daftar lengkap metrik yang gratis dan canggih untuk Lensa Penyimpanan S3.

Lensa Penyimpanan S3 menawarkan metrik gratis untuk semua dasbor dan konfigurasi, dengan opsi untuk meningkatkan ke metrik lanjutan.

- Metrik gratis berisi metrik yang relevan dengan penggunaan penyimpanan, seperti jumlah bucket dan objek di akun Anda. Metrik gratis juga mencakup metrik berbasis kasus penggunaan, seperti optimisasi biaya dan metrik perlindungan data. Semua metrik gratis dikumpulkan setiap hari, dan data tersedia untuk kueri hingga 14 hari.
- Metrik dan rekomendasi lanjutan mencakup semua metrik di metrik gratis bersama dengan metrik tambahan, seperti perlindungan data tingkat lanjut dan metrik optimisasi biaya. Metrik lanjutan juga

mencakup kategori metrik tambahan, seperti metrik aktivitas dan metrik kode status terperinci. Data metrik lanjutan tersedia untuk kueri selama 15 bulan.

Ada biaya tambahan ketika Anda menggunakan Lensa Penyimpanan S3 dengan metrik dan rekomendasi lanjutan. Untuk informasi selengkapnya, lihat Harga Amazon S3. Untuk informasi selengkapnya tentang fitur metrik dan rekomendasi lanjutan, lihat Pilihan metrik.

> **Note**
>
> Untuk grup Lensa Penyimpanan, hanya metrik penyimpanan tingkat gratis yang tersedia. Metrik lanjutan tidak tersedia di tingkat grup Lensa Penyimpanan.

Nama metrik

Kolom nama Metrik di tabel berikut berisikan nama masing-masing Lensa Penyimpanan S3 di konsol S3. Kolom CloudWatch dan ekspor memberikan nama setiap metrik di Amazon CloudWatch dan file ekspor metrik yang dapat Anda konfigurasikan di dasbor Lensa Penyimpanan S3 Anda.

Rumus metrik turunan

Metrik turunan tidak tersedia untuk ekspor metrik dan opsi CloudWatch penerbitan. Namun, Anda dapat menggunakan rumus metrik yang ditampilkan di kolom Rumus metrik turunan untuk menghitungnya.

Menafsirkan simbol awalan Lensa Penyimpanan Amazon S3 untuk kelipatan unit metrik (K, M, G, dan sebagainya)

Kelipatan unit metrik Lensa Penyimpanan S3 ditulis dengan simbol awalan. Simbol awalan ini cocok dengan simbol Sistem Internasional (SI) yang distandarisasi oleh Biro Internasional untuk Ukuran dan Timbangan (BIPM). Simbol-simbol ini juga digunakan di Kode Terpadu untuk Satuan Ukuran (UCUM). Untuk informasi selengkapnya, lihat Daftar simbol awalan SI.

> **Note**
>
> - Unit pengukuran untuk byte penyimpanan S3 adalah dalam gigabyte biner (GB), di mana 1 GB adalah 2^{30} byte, 1 TB adalah 2^{40} byte, dan 1 PB adalah 2^{50} byte. Unit pengukuran ini juga dikenal sebagai gibibyte (GiB), sebagaimana didefinisikan oleh Komisi Elektroteknik Internasional (IEC).

- Saat objek mencapai akhir masa pakai berdasarkan konfigurasi siklus hidupnya, Amazon S3 akan mengantre objek untuk dihapus dan menghapusnya tidak secara bersamaan. Oleh karena itu, penundaan antara tanggal kedaluwarsa dan tanggal saat Amazon S3 menghapus objek mungkin ditemukan. Lensa Penyimpanan S3 tidak menyertakan metrik untuk objek yang telah kedaluwarsa tetapi belum dihapus. Untuk informasi selengkapnya tentang tindakan kedaluwarsa di Siklus Hidup S3, lihat Mengakhiri objek.

Tabel berikut menunjukkan glosarium metrik Lensa Penyimpanan S3.

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
Penyimpanan total	StorageBytes	Penyimpanan total, termasuk unggahan multibagian yang tidak lengkap, metadata objek, dan penanda hapus	Grati	Ring	T	-
Perhitungan Objek	ObjectCount	Perhitungan objek total	Grati	Ring	T	-
Ukuran objek rata-rata	-	Ukuran objek rata-rata	Grati	Ring	Y	jumlah (StorageBytes) / jumlah () ObjectCount
Bucket aktif	-	Jumlah total ember dengan	Kosc	Ring	T	-

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
		penyimpanan> 0 byte				
Bucket	-	Jumlah total bucket	Grati	Ring	T	-
Akun	-	Jumlah akun yang penyimpanannya di dalam cakupan	Grati	Ring	T	-
Byte versi terkini	CurrentVersionStorageBytes	Jumlah byte yang merupakan versi terkini dari suatu objek	Grati	Opti Biay	T	-
% byte versi terkini	-	Persentase byte di dalam cakupan yang merupakan versi objek terkini	Grati	Opti Biay	Y	jumlah (CurrentVersionStorageBytes) / jumlah () StorageBytes
Jumlah objek versi saat ini	CurrentVersionObjectCount	Perhitungan objek versi terkini	Kosc	Perli gan data	T	-

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
% objek versi terkini	-	Persentase objek di dalam cakupan yang merupakan versi terkini	Grati	Opti Biay	Y	jumlah (CurrentVersionObjectCount) / jumlah () ObjectCount
Byte bukan versi terkini	NonCurrentVersionStorageBytes	Jumlah byte bukan versi terkini	Grati	Opti Biay	T	-
% byte bukan versi terkini	-	Persentase byte di dalam cakupan yang bukan versi terkini	Grati	Opti Biay	Y	jumlah (NonCurrentVersionStorageBytes) / jumlah () StorageBytes
Jumlah objek versi tidak terkini	NonCurrentVersionObjectCount	Jumlah objek bukan versi terkini	Grati	Opti Biay	T	-

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
% objek bukan versi terkini	-	Persentase objek di dalam cakupan yang bukan merupakan versi terkini	Grati	Opti Biay	Y	jumlah (NonCurrentVersionObjectCount) / jumlah () ObjectCount
Byte penanda hapus	DeleteMarkerStorageBytes	Jumlah byte di dalam cakupan yang merupakan penanda hapus	Grati	Opti Biay	N	-
% byte penanda hapus	-	Persentase byte di dalam cakupan yang merupakan penanda hapus	Grati	Opti Biay	Y	jumlah (DeleteMarkerStorageBytes) / jumlah () StorageBytes
Hapus Jumlah Objek Penanda	DeleteMarkerObjectCount	Jumlah total objek dengan penanda hapus	Grati	Opti Biay	T	-

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
% objek penanda hapus	-	Persentase objek di dalam cakupan dengan penanda hapus	Grati	Opti Biay	Y	jumlah (DeleteMarkerObjectCount) / jumlah () ObjectCount
Byte unggahan multibagian yang tidak lengkap	IncompleteMultipartUploadStorageBytes	Total byte di dalam cakupan untuk unggahan multibagian yang tidak lengkap	Grati	Opti Biay	T	-
% byte unggahan multibagian yang tidak lengkap	-	Persentase byte di dalam cakupan yang merupakan hasil dari unggahan multibagian yang tidak lengkap	Grati	Opti Biay	Y	jumlah (IncompleteMultipartUploadStorageBytes) / jumlah () StorageBytes
Jumlah objek unggahan multibagian yang tidak lengkap	IncompleteMultipartUploadObjectCount	Jumlah objek di dalam cakupan yang merupakan unggahan multibagian yang tidak lengkap	Grati	Opti Biay	T	-

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
% objek unggahan multibagian yang tidak lengkap	-	Persentase objek di dalam cakupan yang merupakan unggahan multibagian yang tidak lengkap	Grati	Opti Biay	Y	jumlah (IncompleteMultipartUploadObjectCount) / jumlah () ObjectCount
Byte penyimpanan unggahan multibagian yang tidak lengkap dan lebih dari 7 hari	7Days Tidak Lengkap MPUStorage BytesOlderThan	Total byte di dalam cakupan untuk unggahan multibagian yang tidak lengkap dan berusia lebih dari 7 hari	Grati	Opti Biay	T	-
% byte penyimpanan unggahan multibagian yang tidak lengkap dan lebih dari 7 hari	-	Persentase byte untuk unggahan multibagian yang tidak lengkap dan berusia lebih dari 7 hari	Grati	Opti Biay	Y	sum (MPUStorageBytesOlderThan7Hari Tidak Lengkap) / sum () StorageBytes

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
Jumlah objek unggahan multibagian yang tidak lengkap dan lebih dari 7 hari	7Days Tidak Lengkap MPUObject CountOlde rThan	Jumlah objek yang merupakan unggahan multibagian yang tidak lengkap dan lebih dari 7 hari	Grati	Opti Biay	T	-
% Jumlah objek unggahan multibagian yang tidak lengkap dan lebih dari 7 hari	-	Persentase objek yang merupakan unggahan multibagian yang tidak lengkap dan lebih dari 7 hari	Grati	Opti Biay	Y	sum (MPUObjec tCountOld erThan7Ha ri Tidak Lengkap) / sum () ObjectCou nt
Jumlah aturan siklus hidup transisi	Transitio nLifecycl eRuleCount	Perhitungan aturan siklus hidup untuk mengalihk an objek ke kelas penyimpanan lain	Lanju	Opti Biay	T	-

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
Aturan siklus hidup transisi rata-rata per bucket	-	Jumlah rata-rata aturan siklus hidup untuk mengalihkan objek ke kelas penyimpanan lain	Lanju	Optir Biay	Y	jumlah (TransitionLifecycleRuleCount) / jumlah () DistinctNumberOfBuckets
Jumlah aturan siklus hidup kedaluwarsa	ExpirationLifecycleRuleCount	Perhitungan aturan siklus hidup untuk objek kedaluwarsa	Lanju	Optir Biay	T	-
Aturan siklus hidup kedaluwarsa rata-rata per bucket	-	Jumlah rata-rata aturan siklus hidup untuk mengakhiri objek	Lanju	Optir Biay	Y	jumlah (ExpirationLifecycleRuleCount) / jumlah () DistinctNumberOfBuckets

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
Jumlah aturan siklus hidup transisi versi tidak saat ini	NoncurrentVersionTransitionLifecycleRuleCount	Perhitungan aturan siklus hidup untuk mengalihkan objek bukan versi terkini ke kelas penyimpanan lain	Lanju	Optir Biay	T	
Rata-rata aturan siklus hidup transisi bukan versi terkini per bucket	-	Jumlah rata-rata aturan siklus hidup untuk mengalihkan objek bukan versi terkini ke kelas penyimpanan lain	Lanju	Optir Biay	Y	jumlah (NoncurrentVersionTransitionLifecycleRuleCount) / jumlah () DistinctNumberOfBuckets
Jumlah aturan siklus hidup kedaluwarsa bukan versi terkini	NoncurrentVersionExpirationLifecycleRuleCount	Perhitungan aturan siklus hidup untuk mengakhiri objek bukan versi terkini	Lanju	Optir Biay	T	-

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
Rata-rata aturan siklus hidup kedaluwarsa bukan versi terkini per bucket	-	Jumlah rata-rata aturan siklus hidup yang mengakhiri objek bukan versi terkini	Lanju	Opti Biay	Y	jumlah (NoncurrentVersionExpirationLifecycleRuleCount) / jumlah () DistinctNumberOfBuckets
Batalkan jumlah aturan siklus hidup unggahan multibagian yang tidak lengkap	AbortIncompleteMPULifecycleRuleCount	Perhitungan aturan siklus hidup untuk menghapus unggahan multibagian yang tidak lengkap	Lanju	Opti Biay	T	-
Rata-rata pembatalan aturan siklus hidup unggahan multibagian yang tidak lengkap per bucket	-	Jumlah rata-rata aturan siklus hidup untuk menghapus unggahan multibagian yang tidak lengkap	Lanju	Opti Biay	Y	jumlah (AbortIncompleteMPULifecycleRuleCount) / jumlah () DistinctNumberOfBuckets

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
Perhitungan aturan siklus hidup penanda hapus objek yang kedaluwarsa	ExpiredObjectDeleteMarkerLifecycleRuleCount	Perhitungan aturan siklus hidup untuk menghapus penanda hapus objek yang kedaluwarsa	Lanju	Optir Biay	T	-
Rata-rata aturan siklus hidup penanda hapus objek yang kedaluwarsa per bucket	-	Rata-rata jumlah aturan siklus hidup untuk menghapus penanda hapus objek yang kedaluwarsa	Lanju	Optir Biay	Y	jumlah (ExpiredObjectDeleteMarkerLifecycleRuleCount) / jumlah () DistinctNumberOfBuckets
Jumlah aturan siklus hidup total	TotalLifecycleRuleCount	Perhitungan total aturan siklus hidup	Lanju	Optir Biay	T	-
Rata-rata perhitungan aturan siklus hidup per bucket	-	Jumlah rata-rata aturan siklus hidup	Lanju	Optir Biay	Y	jumlah (TotalLifecycleRuleCount) / jumlah () DistinctNumberOfBuckets

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
Byte yang dienkripsi	EncryptedStorageBytes	Jumlah total byte yang dienkripsi	Grati	Perli gan data	T	-
% byte yang dienkripsi	-	Persentase total byte yang dienkripsi	Grati	Perli gan data	Y	jumlah (EncrypteObjectCount) / jumlah () StorageBytes
Perhitungan objek yang dienkripsi	EncryptedObjectCount	Penghitungan total objek yang dienkripsi	Grati	Perli gan data	T	-
% objek yang terenkripsi	-	Persentase objek yang dienkripsi	Grati	Perli gan data	Y	jumlah (EncrypteStorageBytes) / jumlah () ObjectCount

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
Byte yang tidak terenkripsi	UnencryptedStorageBytes	Jumlah byte yang tidak terenkripsi	Grati	Perligan data	Y	jumlah (StorageBytes) - jumlah (EncryptedStorageBytes)
% byte yang tidak terenkripsi	-	Persentase byte yang tidak terenkripsi	Grati	Perligan data	Y	jumlah (UnencryptedStorageBytes) / jumlah () StorageBytes
Perhitungan objek yang tidak terenkripsi	UnencryptedObjectCount	Perhitungan total objek yang tidak terenkripsi	Grati	Perligan data	Y	jumlah (ObjectCount) - jumlah (EncryptedObjectCount)

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
% objek tidak terenkripsi	-	Persentase objek yang tidak terenkripsi	Grati	Perli gan data	Y	jumlah (UnencryptedStorageBytes) / jumlah () ObjectCount
Sumber byte penyimpanan yang direplikasi	ReplicatedStorageBytesSource	Total jumlah byte yang direplikasi dari bucket sumber	Grati	Perli gan data	T	-
% sumber byte yang direplikasi	-	Persentase total byte yang direplikasi dari bucket sumber	Grati	Perli gan data	Y	jumlah (ReplicatedStorageBytesSource) / jumlah () StorageBytes
Sumber perhitungan objek yang direplikasi	ReplicatedObjectCountSource	Perhitungan objek yang direplikasi dari bucket sumber	Grati	Perli gan data	T	-

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
% sumber objek yang direplikasi	-	Persentase total objek yang direplikasi dari bucket sumber	Grati	Perli gan data	Y	jumlah (ReplicatedStorageObjectCount) / jumlah () ObjectCount
Tujuan byte penyimpanan yang direplikasi	ReplicatedStorageBytes	Total jumlah byte yang direplikasi ke bucket tujuan	Grati	Perli gan data	T	-
% tujuan byte yang direplikasi	-	Persentase total byte yang direplikasi ke bucket tujuan	Grati	Perli gan data	Y	jumlah (ReplicatedStorageBytes) / jumlah () StorageBytes
Tujuan perhitungan objek yang direplikasi	ReplicatedObjectCount	Perhitungan objek yang direplikasi ke bucket tujuan	Grati	Perli gan data	T	-

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
% tujuan objek yang direplikasi	-	Persentase total objek yang direplikasi ke bucket tujuan	Grati	Perli gan data	Y	jumlah (ReplicatedObjectCount) / jumlah () ObjectCount
Byte Kunci Objek	ObjectLockEnabledStorageBytes	Perhitungan total byte penyimpanan yang mengaktifkan Kunci Objek	Grati	Perli gan data	T	jumlah (UnencryptedStorageBytes) / sum (ObjectLockEnabledStorageCount) - sum () ObjectLockEnabledStorageBytes

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
% byte Kunci Objek	-	Persentase byte penyimpanan yang mengaktifkan Kunci Objek	Grati	Perli gan data	Y	jumlah (ObjectLockEnabledStorageBytes) / jumlah () StorageBytes
Perhitungan objek Kunci Objek	ObjectLockEnabledObjectCount	Perhitungan total objek Kunci Objek	Grati	Perli gan data	N	-
% objek Kunci Objek	-	Persentase total objek yang mengaktifkan Kunci Objek	Grati	Perli gan data	Y	jumlah (ObjectLockEnabledObjectCount) / jumlah () ObjectCount
Perhitungan bucket yang mengaktifkan Penentuan Versi	VersioningEnabledBucketCount	Perhitungan bucket yang mengaktifkan Penentuan Versi S3	Grati	Perli gan data	T	-

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
Bucket yang mengaktifkan Penentuan Versi	-	Persentase bucket yang mengaktifkan Penentuan Versi S3	Grati	Perli gan data	Y	jumlah (VersioningEnabledBucketCount) / jumlah () DistinctNumberOfBuckets
Perhitungan bucket yang mengaktifkan penghapusan MFA	MFADeleteEnabledBucketCount	Perhitungan bucket yang mengaktifkan penghapusan MFA (autentikasi multi-faktor)	Grati	Perli gan data	T	-
% bucket yang mengaktifkan penghapusan MFA	-	Persentase bucket yang mengaktifkan penghapusan MFA (autentikasi multi-faktor)	Grati	Perli gan data	Y	jumlah (MFADeleteEnabledBucketCount) / jumlah () DistinctNumberOfBuckets

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
Jumlah bucket yang mengaktifkan SSE-KMS	SSEKMSEnabledBucketCount	Jumlah bucket yang menggunakan enkripsi sisi server dengan AWS Key Management Service kunci (SSE-KMS) untuk enkripsi bucket default	Kosc	Perli gan data	T	-
% bucket yang mengaktifkan SSE-KMS	-	Persentase bucket yang mengaktifkan SSE-KMS untuk enkripsi bucket default	Grati	Perli gan data	Y	jumlah (SSEKMSEnabledBucketCount) / jumlah () DistinctNumberOfBuckets
Semua permintaan tanda tangan yang tidak didukung	AllUnsupportedSignatureRequests	Jumlah total permintaan yang menggunakan versi AWS tanda tangan yang tidak didukung	Lanju	Perli gan data	T	-

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
% semua permintaan tanda tangan yang tidak didukung	-	Persentase permintaan yang menggunakan versi AWS tanda tangan yang tidak didukung	Lanju	Perli gan data	Y	jumlah (AllUnsup portedSig natureReq uests) / jumlah () AllReques ts
Semua permintaan TLS yang tidak didukung	AllUnsupp ortedTLSR equests	Jumlah permintaan yang menggunakan versi Keamanan Lapisan Pengangkutan (TLS) yang tidak didukung	Lanju	Perli gan data	T	-
% semua permintaan TLS yang tidak didukung	-	Persentase permintaan yang menggunakan versi TLS yang tidak didukung	Lanju	Perli gan data	Y	jumlah (AllUnsup portedTLS Requests) / jumlah () AllReques ts
Semua permintaan SSE-KMS	Semua SSEKMSReq uests	Jumlah total permintaan yang menentukan SSE-KMS	Lanju	Perli gan data	T	-

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
% semua permintaan SSE-KMS	-	Persentase permintaan yang menentukan SSE-KMS	Lanju	Perli gan data	Y	sum (SemuaSSE KMSReques ts) / sum () AllReques ts
Jumlah aturan Replikasi Wilayah yang Sama	SameRegio nReplicat ionRuleCount	Perhitungan aturan replikasi untuk Replikasi Wilayah yang Sama (SRR)	Lanju	Perli gan data	T	-
Rata-rata aturan Replikasi Lintas-Wilayah per bucket	-	Jumlah rata-rata aturan replikasi untuk SRR	Lanju	Perli gan data	Y	jumlah (SameRegi onReplica tionRuleC ount) / jumlah () DistinctN umberOfBu ckets
Jumlah aturan Replikasi Lintas Wilayah	CrossRegi onReplica tionRuleCount	Jumlah aturan replikasi untuk Replikasi Lintas Wilayah (CRR)	Lanju	Perli gan data	T	-

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
Rata-Rata Aturan Replikasi Lintas Wilayah per bucket	-	Jumlah rata-rata aturan replikasi untuk CRR	Lanju	Perli gan data	Y	jumlah (CrossReg ionReplic ationRule Count) / jumlah () DistinctN umberOfBu ckets
Jumlah aturan replikasi akun yang sama	SameAccou ntReplica tionRuleCount	Perhitungan aturan replikasi untuk replikasi di dalam akun yang sama	Lanju	Perli gan data	T	-
Rata-rata aturan replikasi akun yang sama per bucket	-	Jumlah rata-rata aturan replikasi untuk replikasi di dalam akun yang sama	Lanju	Perli gan data	Y	jumlah (SameAcco untReplic ationRule Count) / jumlah () DistinctN umberOfBu ckets
Jumlah aturan replikasi lintas akun	CrossAcco untReplic ationRuleCount	Perhitungan aturan replikasi untuk replikasi lintas akun	Lanju	Perli gan data	T	-

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
Aturan replikasi lintas akun rata-rata per bucket	-	Jumlah rata-rata aturan replikasi untuk replikasi lintas akun	Lanju	Perli gan data	Y	jumlah (CrossAccountReplicationRuleCount) / jumlah () DistinctNumberOfBuckets
Perhitungan aturan replikasi tujuan yang tidak valid	InvalidDestination ReplicationRuleCount	Perhitungan aturan replikasi dengan tujuan replikasi yang tidak valid	Lanju	Perli gan data	T	-
Aturan replikasi tujuan rata-rata tidak valid per bucket	-	Jumlah rata-rata aturan replikasi dengan tujuan replikasi yang tidak valid	Lanju	Perli gan data	Y	jumlah (InvalidReplicationRuleCount) / jumlah () DistinctNumberOfBuckets
Jumlah total aturan replikasi	-	Jumlah total aturan replikasi	Lanju	Perli gan data	Y	-

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
Perhitungan aturan replikasi rata-rata per bucket	-	Perhitungan aturan replikasi total rata-rata	Lanju	Perli gan data	Y	sum (semua metrik hitungan aturan replikasi) / sum () DistinctNumberOfBuckets
Perhitungan bucket yang diberlakukan pemilik bucket Kepemilikan Objek	ObjectOwnershipBucketOwnerEnforcedBucketCount	Jumlah total bucket yang memiliki daftar kontrol akses (ACLs) dinonaktifkan dengan menggunakan setelan yang diberlakukan pemilik bucket untuk Kepemilikan Objek	Kosc	Man akse	T	-

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
% Bucke yang diberlaku kan pemilik bucket Objek Kepemilikan	-	Persentase bucket yang telah ACLs dinonakti fkan dengan menggunakan pengaturan yang diberlakukan pemilik bucket untuk Kepemilikan Objek	Kosc	Man akse	Y	jumlah (ObjectOw nershipBu cketOwner EnforcedB ucketCoun t) / jumlah () DistinctN umberOfBu ckets
Penghitun gan bucket pilihan pemilik bucket Objek Kepemilikan	ObjectOwn ershipBuc ketOwnerP referredB ucketCount	Penghitungan total bucket yang menggunakan pengaturan bucket pilihan pemilik untuk Kepemilikan Objek	Grati	Man akse	T	-
% Bucket pilihan pemilik bucket Kepemilikan Objek	-	Persentase bucket yang menggunak an pengaturan bucket pilihan pemilik untuk Kepemilikan Objek	Grati	Man akse	Y	jumlah (ObjectOw nershipBu cketOwner Preferred BucketCou nt) / jumlah () DistinctN umberOfBu ckets

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
Perhitungan bucket penulis objek Kepemilikan Objek	ObjectOwnershipObjectWriterBucketCount	Perhitungan total bucket yang menggunakan pengaturan penulis objek untuk Kepemilikan Objek	Grati	Man akse	T	-
% Bucket penulis objek Kepemilikan Objek	-	Persentase bucket yang menggunakan pengaturan penulis objek untuk Kepemilikan Objek	Grati	Man akse	Y	jumlah (ObjectOwnershipObjectWriterBucketCount) / jumlah () DistinctNumberOfBuckets
Perhitungan bucket yang mengaktifkan Akselerasi Transfer	TransferAccelerationEnabledBucketCount	Jumlah total bucket yang mengaktifkan Akselerasi Transfer	Grati	Kine	T	-

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
% Bucket yang mengaktifkan Akselerasi Transfer	-	Persentase bucket yang mengaktifkan Akselerasi Transfer	Grati	Kine	Y	jumlah (TransferAccelerationEnabledBucketCount) / jumlah () DistinctNumberOfBuckets
Perhitungan bucket yang mengaktifkan Pemberitahuan Peristiwa	EventNotificationEnabledBucketCount	Jumlah total bucket yang mengaktifkan Pemberitahuan Peristiwa	Grati	Peris	T	
% Bucket yang mengaktifkan Pemberitahuan Peristiwa	-	Persentase bucket yang mengaktifkan Pemberitahuan Peristiwa	Grati	Peris	Y	jumlah (EventNotificationEnabledBucketCount) / jumlah () DistinctNumberOfBuckets
Semua permintaan	AllRequests	Jumlah total permintaan yang dibuat	Lanju	Aktif	T	-

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
Permintaan pengambilan	GetRequests	Jumlah total permintaan `GET` yang dibuat	Lanju	Aktif	T	-
Permintaan penempatan	PutRequests	Jumlah total permintaan `PUT` yang dibuat	Lanju	Aktif	T	-
Permintaan head	HeadRequests	Jumlah total permintaan `HEAD` yang dibuat	Lanju	Aktif	T	-
Permintaan penghapusan	DeleteRequests	Jumlah total permintaan `DELETE` yang dibuat	Lanju	Aktif	T	-
Permintaan pencantuman	ListRequests	Jumlah total permintaan `LIST` yang dibuat	Lanju	Aktif	T	-
Permintaan posting	PostRequests	Jumlah total permintaan `POST` yang dibuat	Lanju	Aktif	T	-
Permintaan Select	SelectRequests	Jumlah total permintaan S3 Select	Lanju	Aktif	T	-
Byte Select yang dipindai	SelectScannedBytes	Jumlah byte S3 Select yang dipindai	Lanju	Aktif	T	-

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
Byte Select yang ditampilkan	SelectReturnedBytes	Jumlah byte S3 Select yang ditampilkan	Lanju	Aktif	T	-
Byte yang diunduh	BytesDownloaded	Jumlah byte yang diunduh	Lanju	Aktif	T	-
% tingkat pengambilan	-	Persentase byte yang diunduh	Lanju	Aktif	Y	jumlah (BytesDownloaded) / jumlah () StorageBytes
Byte yang diunggah	BytesUploaded	Jumlah byte yang diunggah	Lanju	Aktif	T	-
% rasio konsumsi	-	Persentase byte yang diunggah	Lanju	Aktif	Y	jumlah (BytesUploaded) / jumlah () StorageBytes
Kesalahan 4xx	4xxErrors	Jumlah total kode status HTTP 4xx	Lanju	Aktif	T	-
Kesalahan 5xx	5xxErrors	Jumlah total kode status HTTP 5xx	Lanju	Aktif	T	-

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
Kesalahan total	-	Jumlah semua kesalahan 4xx dan 5xx	Lanju	Aktif	Y	sum(4xxErrors) + sum(5xxErrors)
% tingkat kesalahan	-	Jumlah total kesalahan 4xx dan 5xx sebagai persentase dari total permintaan	Lanju	Aktif	Y	jumlah (TotalErrors) / jumlah () TotalRequests
Perhitungan status 200 OK	200 OKStatus Hitungan	Perhitungan total kode status 200 OK	Lanju	Kode statu terpe i	T	-
% status 200 OK	-	Jumlah total kode status 200 OK sebagai persentase dari total permintaan	Lanju	Kode statu terpe i	T	jumlah (200 OKStatus Hitungan) / jumlah () AllRequests
Perhitungan status 206 Partial Content	206 PartialContentStatusCount	Perhitungan total kode status 206 Partial Content	Lanju	Kode statu terpe i	T	-

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
% status 206 Partial Content	-	Jumlah total kode status 206 Partial Content sebagai persentase dari total permintaan	Lanju	Kode statu terpe i	T	jumlah (206PartialContentStatusCount) / jumlah () AllRequests
Perhitungan kesalahan 400 Bad Request	400 BadRequestErrorCount	Jumlah total 400 kode status 400 Bad Request	Lanju	Kode statu terpe i	T	-
% kesalahan 400 Bad Request	-	Jumlah total kode status 400 Bad Request sebagai persentase dari total permintaan	Lanju	Kode statu terpe i	T	jumlah (400BadRequestErrorCount) / jumlah () AllRequests
Perhitungan kesalahan 403 Forbidden	403 Forbidden ErrorCount	Jumlah total 403 kode status 403 Forbidden	Lanju	Kode statu terpe i	T	-

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
% kesalahan 403 Forbidden	-	Total jumlah kode status 403 Forbidden sebagai persentase dari total permintaan	Lanju	Kode statu terpe i	T	jumlah (403ForbiddenError Count) / jumlah () AllRequests
Perhitungan kesalahan 404 Not Found	404 NotFoundErrorCount	Perhitungan total kode status 404 Not Found	Lanju	Kode statu terpe i	T	-
% kesalahan 404 Not Found	-	Jumlah total kode status 404 Not Found sebagai persentase dari total permintaan	Lanju	Kode statu terpe i	T	jumlah (404NotFoundErrorCount) / jumlah () AllRequests
Perhitungan 500 Internal Server Error	500 InternalServerErrorCount	Perhitungan total kode status 500 Internal Server Error	Lanju	Kode statu terpe i	T	-

Nama metrik	CloudWatch dan ekspor	Deskripsi	Tingk[1]	Kate[2]	Tu	Form metri turur
% 500 Internal Server Errors	-	Jumlah total kode status 500 Internal Server Error sebagai persentase dari total permintaan	Lanju	Kode statu terpe i	T	jumlah (500InternalServerErrorCount) / jumlah () AllRequests
Perhitungan kesalahan 503 Service Unavailable	503 ServiceUnavailableErrorCount	Jumlah total kode status 503 Service Unavailable	Lanju	Kode statu terpe i	T	-
% kesalahan 503 Service Unavailable	-	Jumlah total kode status 503 Service Unavailable sebagai persentase dari total permintaan	Lanju	Kode statu terpe i	T	jumlah (503ServiceUnavailableErrorCount) / jumlah () AllRequests

[1] Semua metrik penyimpanan tingkat gratis tersedia di tingkat grup Lensa Penyimpanan. Metrik lanjutan tidak tersedia di tingkat grup Lensa Penyimpanan.

[2] Metrik perhitungan aturan dan metrik pengaturan bucket tidak tersedia di tingkat awalan.

Menyetel izin Lensa Penyimpanan Amazon S3

Amazon S3 Storage Lens memerlukan izin baru di AWS Identity and Access Management (IAM) untuk mengotorisasi akses ke tindakan Lensa Penyimpanan S3. Untuk memberikan izin tersebut, Anda dapat menggunakan kebijakan IAM berbasis identitas. Anda dapat melampirkan kebijakan ini ke pengguna, grup, atau peran IAM untuk diberikan izin. Izin tersebut dapat mencakup kemampuan untuk mengaktifkan atau menonaktifkan Lensa Penyimpanan S3, atau untuk mengakses dasbor atau konfigurasi Lensa Penyimpanan S3 apa pun.

Pengguna atau peran IAM harus dimiliki oleh akun yang membuat atau memiliki dasbor atau konfigurasi, kecuali kedua kondisi berikut ini benar:

- Akun Anda adalah anggota AWS Organizations.
- Anda diberi akses untuk membuat dasbor tingkat organisasi oleh akun manajemen sebagai administrator yang didelegasikan.

> **Note**
>
> - Anda tidak dapat menggunakan akun kredensial pengguna root untuk melihat dasbor Lensa Penyimpanan Amazon S3. Untuk mengakses dasbor Lensa Penyimpanan S3, Anda harus memberikan izin IAM yang diperlukan untuk pengguna IAM baru atau yang sudah ada. Lalu, masuk dengan kredensial pengguna tersebut untuk mengakses dasbor Lensa Penyimpanan S3. Untuk informasi selengkapnya tentang administrator, lihat Praktik terbaik keamanan di IAM di dalam Panduan Pengguna IAM.
> - Menggunakan Lensa Penyimpanan S3 pada konsol Amazon S3 dapat memerlukan beberapa izin. Misalnya, untuk mengedit dasbor di konsol, Anda memerlukan izin berikut:
>   - `s3:ListStorageLensConfigurations`
>   - `s3:GetStorageLensConfiguration`
>   - `s3:PutStorageLensConfiguration`

Topik

- Mengatur izin akun untuk menggunakan Lensa Penyimpanan S3
- Mengatur izin akun untuk menggunakan grup Lensa Penyimpanan S3

- Mengatur izin untuk menggunakan S3 Storage Lens dengan AWS Organizations

Mengatur izin akun untuk menggunakan Lensa Penyimpanan S3

Untuk membuat dan mengelola dasbor Lensa Penyimpanan S3 dan konfigurasi dasbor Lensa Penyimpanan, Anda harus memiliki izin berikut, tergantung pada tindakan yang ingin dilakukan:

Tabel berikut menunjukkan izin IAM terkait Amazon S3 Storage Lens.

Tindakan	Izin IAM

Membuat atau memperbarui dasbor Lensa Penyimpanan S3 di konsol Amazon S3.	`s3:ListStorageLensConfigurations`
	`s3:GetStorageLensConfiguration`
	`s3:GetStorageLensConfigurationTagging`
	`s3:PutStorageLensConfiguration`
	`s3:PutStorageLensConfigurationTagging`

Mendapatkan tag dasbor Lensa Penyimpanan S3 di konsol Amazon S3.	`s3:ListStorageLensConfigurations` `s3:GetStorageLensConfigurationTagging`

Melihat dasbor Lensa Penyimpanan S3 di konsol Amazon S3.	`s3:ListStorageLensConfigurations`
	`s3:GetStorageLensConfiguration`
	`s3:GetStorageLensDashboard`

Menghapus dasbor Lensa Penyimpanan S3 di konsol Amazon S3.	`s3:ListStorageLensConfigurations`
	`s3:GetStorageLensConfiguration`
	`s3:DeleteStorageLensConfiguration`

Tindakan	Izin IAM
Membuat atau memperbarui konfigurasi S3 Storage Lens dengan menggunakan AWS CLI atau AWS SDK.	`s3:PutStorageLensConfiguration` `s3:PutStorageLensConfigurationTagging`
Dapatkan tag konfigurasi Lensa Penyimpanan S3 dengan menggunakan AWS CLI atau AWS SDK.	`s3:GetStorageLensConfigurationTagging`
Melihat konfigurasi Lensa Penyimpanan S3 menggunakan AWS CLI atau AWS SDK.	`s3:GetStorageLensConfiguration`
Hapus konfigurasi Lensa Penyimpanan S3 dengan menggunakan AWS CLI atau AWS SDK.	`s3:DeleteStorageLensConfiguration`

Note

- Anda dapat menggunakan tag sumber daya dalam kebijakan IAM untuk mengelola izin.
- Pengguna atau peran IAM dengan izin ini dapat melihat metrik dari bucket dan awalan yang mungkin tidak memiliki izin langsung untuk membaca atau mencantumkan objek.
- Untuk dasbor Lensa Penyimpanan S3 dengan metrik tingkat awalan yang diaktifkan, jika jalur awalan yang dipilih cocok dengan kunci objek, dasbor tersebut mungkin akan menampilkan kunci objek sebagai awalan lain.
- Untuk ekspor metrik, yang disimpan di dalam bucket di akun, izin akan diberikan dengan menggunakan izin `s3:GetObject` yang sudah ada di dalam kebijakan IAM. Demikian pula, untuk AWS Organizations entitas, akun manajemen organisasi atau akun administrator yang didelegasikan dapat menggunakan kebijakan IAM untuk mengelola izin akses untuk dasbor dan konfigurasi tingkat organisasi.

Mengatur izin akun untuk menggunakan grup Lensa Penyimpanan S3

Anda dapat menggunakan grup Lensa Penyimpanan S3 untuk memahami distribusi penyimpanan di dalam bucket berdasarkan awalan, akhiran, tag objek, ukuran objek, atau usia objek. Anda dapat melampirkan grup Lensa Penyimpanan ke dasbor untuk melihat metrik gabungannya.

Untuk dapat bekerja dengan grup Lensa Penyimpanan, Anda memerlukan izin tertentu. Untuk informasi selengkapnya, lihat the section called “Izin grup Lensa Penyimpanan”.

Mengatur izin untuk menggunakan S3 Storage Lens dengan AWS Organizations

Anda dapat menggunakan Amazon S3 Storage Lens untuk mengumpulkan metrik penyimpanan dan data penggunaan untuk semua akun yang merupakan bagian dari hierarki Anda. AWS Organizations Tabel berikut menunjukkan tindakan dan izin yang terkait dengan penggunaan S3 Storage Lens with Organizations.

Tindakan	Izin IAM
Aktifkan akses tepercaya untuk Lensa Penyimpanan S3 untuk organisasi Anda.	`organizations:EnableAWSServiceAccess`
Nonaktifkan akses tepercaya untuk S3 Lensa Penyimpanan untuk organisasi Anda.	`organizations:DisableAWSServiceAccess`
Daftarkan administrator yang didelegasikan untuk membuat dasbor atau konfigurasi Lensa Penyimpanan S3 untuk organisasi Anda.	`organizations:RegisterDelegatedAdministrator`
Batalkan pendaftaran administrator yang didelegasikan sehingga administrator tidak lagi dapat membuat dasbor atau konfigurasi Lensa Penyimpanan S3 untuk organisasi Anda.	`organizations:DeregisterDelegatedAdministrator`

Izin tambahan untuk membuat konfigurasi Lensa Penyimpanan S3 di seluruh organisasi.	`organizations:DescribeOrganization`
	
	`organizations:ListAccounts`

Tindakan	Izin IAM

	`organizations:ListAWSServiceAccessForOrganization`
	
	`organizations:ListDelegatedAdministrators`
	
	`iam:CreateServiceLinkedRole`

Bekerja dengan Lensa Penyimpanan Amazon S3 menggunakan konsol dan API

Lensa Penyimpanan Amazon S3 adalah fitur analisis penyimpanan cloud yang dapat Anda gunakan untuk mendapatkan visibilitas seluruh organisasi ke dalam penggunaan dan aktivitas penyimpanan objek. Anda dapat menggunakan metrik Lensa Penyimpanan S3 untuk menghasilkan wawasan ringkasan, seperti mencari tahu berapa banyak penyimpanan yang Anda miliki di seluruh organisasi atau bucket dan prefiks yang paling cepat berkembang. Anda juga dapat menggunakan metrik Lensa Penyimpanan S3 untuk mengidentifikasi peluang optimasi biaya, menerapkan praktik terbaik untuk perlindungan data dan keamanan, serta meningkatkan kinerja beban kerja aplikasi. Misalnya, Anda dapat mengidentifikasi bucket yang tidak memiliki aturan Siklus Hidup S3 untuk mengakhiri unggahan multibagian yang tidak lengkap dan berusia lebih dari 7 hari. Anda juga dapat mengidentifikasi bucket yang tidak mengikuti praktik terbaik perlindungan data, seperti menggunakan Replikasi S3 atau Penentuan Versi S3. Lensa Penyimpanan S3 menganalisis metrik untuk memberikan rekomendasi kontekstual yang dapat Anda gunakan untuk mengoptimalkan biaya penyimpanan dan menerapkan praktik terbaik untuk melindungi data Anda.

Lensa Penyimpanan S3 menggabungkan metrik Anda dan menampilkan informasi di bagian Snapshot akun di halaman Bucket konsol Amazon S3. Lensa Penyimpanan S3 juga menyediakan dasbor interaktif yang dapat Anda gunakan untuk memvisualisasikan wawasan dan tren, menandai outlier, serta menerima rekomendasi untuk mengoptimalkan biaya penyimpanan dan menerapkan praktik terbaik perlindungan data. Dasbor Anda memiliki opsi perincian untuk menghasilkan dan memvisualisasikan wawasan di tingkat organisasi, akun, Wilayah AWS, kelas penyimpanan, bucket, prefiks, atau grup Lensa Penyimpanan. Anda juga dapat mengirimkan ekspor metrik harian dalam format CSV atau Parquet ke bucket S3.

Bagian berikut berisi contoh membuat, memperbarui, dan melihat konfigurasi Lensa Penyimpanan S3 serta melakukan operasi yang terkait dengan fitur. Jika Anda menggunakan S3 Storage Lens dengan AWS Organizations, contoh-contoh ini juga mencakup kasus penggunaan tersebut. Dalam contoh, ganti nilai variabel dengan nilai spesifik untuk Anda.

Topik

- Buat dasbor Lensa Penyimpanan Amazon S3
- Perbarui dasbor Lensa Penyimpanan Amazon S3
- Nonaktifkan dasbor Lensa Penyimpanan Amazon S3
- Hapus dasbor Lensa Penyimpanan Amazon S3
- Daftar dasbor Lensa Penyimpanan Amazon S3
- Melihat detail konfigurasi dasbor Amazon S3 Storage Lens
- Mengelola tag AWS sumber daya dengan Lensa Penyimpanan S3
- File pembantu untuk menggunakan Lensa Penyimpanan Amazon S3

Buat dasbor Lensa Penyimpanan Amazon S3

Anda dapat membuat dasbor kustom S3 Storage Lens tambahan yang dapat dicakup ke organisasi Anda di dalam AWS Organizations atau ke spesifik Wilayah AWS atau bucket dalam akun.

> **Note**
>
> Setiap pembaruan konfigurasi dasbor dapat memakan waktu hingga 48 jam untuk ditampilkan atau divisualisasikan secara akurat.

Menggunakan konsol S3

Gunakan langkah-langkah berikut untuk membuat dasbor Lensa Penyimpanan Amazon S3 di konsol Amazon S3.

Langkah 1: Tentukan cakupan dasbor

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi di bagian atas halaman, pilih nama AWS Wilayah yang saat ini ditampilkan. Selanjutnya, pilih Wilayah yang ingin Anda alihkan.

3. Di panel navigasi kiri, di bawah Lensa Penyimpanan S3, pilih Dasbor.
4. Pilih Buat dasbor.
5. Di laman Dasbor, di bagian Umum, lakukan hal-hal berikut:

   a. Lihat Wilayah Beranda untuk dasbor Anda. Wilayah beranda adalah Wilayah AWS tempat konfigurasi dan metrik untuk dasbor Lensa Penyimpanan ini disimpan.

   b. Masukkan nama dasbor.

      Nama dasbor harus kurang dari 65 karakter dan tidak boleh berisi karakter khusus atau spasi.

      > **Note**
      >
      > Nama dasbor ini tidak dapat diubah setelah dasbor dibuat.

   c. Secara opsional, Anda dapat memilih untuk menambahkan Tag ke dasbor. Anda dapat menggunakan tag untuk mengelola izin dasbor dan melacak biaya untuk Lensa Penyimpanan S3.

      Untuk informasi lebih lanjut, lihat Mengontrol akses menggunakan tag sumber daya di Panduan Pengguna IAM dan AWS Tag Alokasi Biaya yang Dihasilkan di AWS Billing Panduan Pengguna.

      > **Note**
      >
      > Anda dapat menambahkan hingga 50 tag ke konfigurasi dasbor.

6. Di bagian Cakupan dasbor, lakukan hal-hal berikut:

   a. Pilih Wilayah dan bucket yang Anda inginkan untuk disertakan atau dikecualikan oleh Lensa Penyimpanan S3 di dasbor.

   b. Pilih bucket di Wilayah terpilih yang Anda inginkan untuk disertakan atau dikecualikan oleh Lensa Penyimpanan S3. Anda dapat menyertakan atau mengecualikan bucket, tetapi tidak keduanya. Opsi ini tidak tersedia jika Anda membuat dasbor tingkat organisasi.

> **Note**
>
> - Anda dapat menyertakan atau mengecualikan Wilayah dan bucket. Pilihan ini terbatas pada Wilayah saja saat membuat dasbor tingkat organisasi di seluruh akun anggota di organisasi Anda.
> - Anda dapat memilih hingga 50 bucket untuk disertakan atau dikecualikan.

Langkah 2: Konfigurasikan pilihan metrik

1. Di bagian Pilihan metrik, pilih jenis metrik yang ingin digabungkan untuk dasbor ini.

   - Untuk menyertakan metrik gratis yang digabungkan di tingkat bucket dan tersedia untuk kueri selama 14 hari, pilih Metrik gratis.
   - Untuk dapat mengaktifkan metrik lanjutan dan opsi lanjutan lainnya, pilih Metrik dan rekomendasi lanjutan. Opsi ini mencakup agregasi awalan lanjutan, CloudWatch penerbitan Amazon, dan rekomendasi kontekstual. Data tersedia untuk kueri selama 15 bulan. Metrik dan rekomendasi lanjutan memiliki biaya tambahan. Untuk informasi selengkapnya, lihat Harga Amazon S3.

   Untuk informasi selengkapnya tentang metrik lanjutan dan metrik gratis, lihat Pilihan metrik.
2. Di bagian bawah Fitur metrik dan rekomendasi lanjutan, pilih opsi yang ingin diaktifkan:

   - Metrik lanjutan
   - CloudWatch penerbitan
   - Agregasi awalan

> **Important**
>
> Jika Anda mengaktifkan agregasi awalan untuk konfigurasi Lensa Penyimpanan S3 Anda, metrik tingkat awalan tidak akan dipublikasikan. CloudWatch Hanya metrik Lensa Penyimpanan S3 tingkat bucket, akun, dan organisasi yang dipublikasikan. CloudWatch

3. Jika Anda mengaktifkan Metrik lanjutan, pilih Kategori metrik lanjutan yang ingin ditampilkan di dasbor Lensa Penyimpanan S3:

- Metrik aktivitas
- Metrik kode status terperinci
- Metrik pengoptimalan biaya lanjutan
- Metrik perlindungan data lanjutan

Untuk informasi selengkapnya tentang kategori metrik, lihat Kategori metrik. Untuk daftar lengkap metrik, lihat Glosarium metrik Lensa Penyimpanan Amazon S3.

4. Jika Anda memilih untuk mengaktifkan agregasi awalan, konfigurasikan hal-hal berikut ini:

   a. Pilih ukuran ambang batas awalan minimum untuk dasbor ini.

      Misalnya, ambang batas awalan sebesar 5 persen menunjukkan bahwa awalan yang membentuk 5 persen atau lebih dari total ukuran penyimpanan bucket akan digabungkan.

   b. Pilih kedalaman awalan.

      Pengaturan ini menunjukkan jumlah maksimum level hingga awalan dapat dievaluasi. Kedalaman awalan harus kurang dari 10.

   c. Masukkan karakter pembatas awalan.

      Nilai ini digunakan untuk mengidentifikasi setiap tingkat awalan. Nilai default di Amazon S3 adalah karakter `/`, tapi struktur penyimpanan Anda mungkin akan menggunakan karakter pembatas lain.

(Opsional) Langkah 3: Ekspor metrik untuk dasbor

1. Di bagian Ekspor metrik, untuk membuat ekspor metrik yang akan ditempatkan setiap hari di bucket tujuan pilihan Anda, pilih Aktifkan.

   Ekspor metrik berlaku dalam format CSV atau Apache Parquet. Ketentuan ini mewakili cakupan data yang sama dengan data dasbor Lensa Penyimpanan S3 Anda tanpa rekomendasi.

2. Jika ekspor metrik diaktifkan, pilih format output ekspor metrik harian Anda: CSV atau Apache Parquet.

   Parquet adalah format file sumber terbuka untuk Hadoop yang menyimpan data bertumpuk dalam format kolom datar.

3. Pilih bucket S3 tujuan untuk ekspor metrik Anda.

Anda dapat memilih bucket di akun saat ini di dasbor Lensa Penyimpanan S3. Atau Anda dapat memilih yang lain Akun AWS jika Anda memiliki izin bucket tujuan dan ID akun pemilik bucket tujuan.

4. Pilih bucket S3 tujuan (format: `s3://bucket-name/prefix`).

   Bucket harus berada di Wilayah asal dasbor Lensa Penyimpanan S3 Anda. Konsol S3 menampilkan Izin bucket tujuan yang akan ditambahkan oleh Amazon S3 ke kebijakan bucket tujuan. Amazon S3 memperbarui kebijakan bucket di bucket tujuan agar S3 dapat menempatkan data di bucket tersebut.

5. (Opsional) Untuk mengaktifkan enkripsi di sisi server untuk ekspor metrik Anda, pilih Tentukan kunci enkripsi. Kemudian, pilih Jenis enkripsi: Kunci terkelola Amazon S3 (SSE-S3) atau AWS Key Management Service kunci (SSE-KMS).

   Anda dapat memilih antara kunci terkelola Amazon S3 (SSE-S3) dan kunci AWS Key Management Service (AWS KMS) (SSE-KMS).

6. (Opsional) Untuk menentukan AWS KMS kunci, Anda harus memilih kunci KMS atau memasukkan kunci Nama Sumber Daya Amazon (ARN).

   Jika Anda memilih CMK, Anda harus memberikan izin Lensa Penyimpanan S3 untuk dapat mengenkripsi di dalam kebijakan kunci AWS KMS . Untuk informasi selengkapnya, lihat Menggunakan AWS KMS key untuk mengenkripsi ekspor metrik Anda.

7. Pilih Buat dasbor.

Menggunakan AWS CLI

Example

Perintah contoh berikut membuat konfigurasi Lensa Penyimpanan Amazon S3 dengan tag. Untuk menggunakan contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3control put-storage-lens-configuration --account-id=111122223333 --
config-id=example-dashboard-configuration-id --region=us-east-1 --storage-lens-
configuration=file://./config.json --tags=file://./tags.json
```

Example

Perintah contoh berikut membuat konfigurasi Lensa Penyimpanan Amazon S3 tanpa tag. Untuk menggunakan contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3control put-storage-lens-configuration --account-id=222222222222 --config-
id=your-configuration-id --region=us-east-1 --storage-lens-configuration=file://./
config.json
```

Menggunakan AWS SDK for Java

Example — Buat dan perbarui konfigurasi Lensa Penyimpanan Amazon S3

Contoh berikut membuat dan memperbarui konfigurasi Lensa Penyimpanan Amazon S3 di SDK for Java:

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.AccountLevel;
import com.amazonaws.services.s3control.model.ActivityMetrics;
import com.amazonaws.services.s3control.model.BucketLevel;
import com.amazonaws.services.s3control.model.CloudWatchMetrics;
import com.amazonaws.services.s3control.model.Format;
import com.amazonaws.services.s3control.model.Include;
import com.amazonaws.services.s3control.model.OutputSchemaVersion;
import com.amazonaws.services.s3control.model.PrefixLevel;
import com.amazonaws.services.s3control.model.PrefixLevelStorageMetrics;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.S3BucketDestination;
import com.amazonaws.services.s3control.model.SSES3;
import com.amazonaws.services.s3control.model.SelectionCriteria;
import com.amazonaws.services.s3control.model.StorageLensAwsOrg;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensDataExport;
import com.amazonaws.services.s3control.model.StorageLensDataExportEncryption;
import com.amazonaws.services.s3control.model.StorageLensTag;

import java.util.Arrays;
import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class CreateAndUpdateDashboard {
```

```
    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "111122223333";
        String exportAccountId = "Destination Account ID";
        String exportBucketArn = "arn:aws:s3:::destBucketName"; // The destination
 bucket for your metrics export must be in the same Region as your S3 Storage Lens
 configuration.
        String awsOrgARN = "arn:aws:organizations::123456789012:organization/o-
abcdefgh";
        Format exportFormat = Format.CSV;

        try {
            SelectionCriteria selectionCriteria = new SelectionCriteria()
                    .withDelimiter("/")
                    .withMaxDepth(5)
                    .withMinStorageBytesPercentage(10.0);
            PrefixLevelStorageMetrics prefixStorageMetrics = new
 PrefixLevelStorageMetrics()
                    .withIsEnabled(true)
                    .withSelectionCriteria(selectionCriteria);
            BucketLevel bucketLevel = new BucketLevel()
                    .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
                    .withAdvancedCostOptimizationMetrics(new
 AdvancedCostOptimizationMetrics().withIsEnabled(true))
                    .withAdvancedDataProtectionMetrics(new
 AdvancedDataProtectionMetrics().withIsEnabled(true))
                    .withDetailedStatusCodesMetrics(new
 DetailedStatusCodesMetrics().withIsEnabled(true))
                    .withPrefixLevel(new
 PrefixLevel().withStorageMetrics(prefixStorageMetrics));
            AccountLevel accountLevel = new AccountLevel()
                    .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
                    .withAdvancedCostOptimizationMetrics(new
 AdvancedCostOptimizationMetrics().withIsEnabled(true))
                    .withAdvancedDataProtectionMetrics(new
 AdvancedDataProtectionMetrics().withIsEnabled(true))
                    .withDetailedStatusCodesMetrics(new
 DetailedStatusCodesMetrics().withIsEnabled(true))
                    .withBucketLevel(bucketLevel);

            Include include = new Include()
                    .withBuckets(Arrays.asList("arn:aws:s3:::bucketName"))
                    .withRegions(Arrays.asList("us-west-2"));
```

```
            StorageLensDataExportEncryption exportEncryption = new
StorageLensDataExportEncryption()
                    .withSSES3(new SSES3());
            S3BucketDestination s3BucketDestination = new S3BucketDestination()
                    .withAccountId(exportAccountId)
                    .withArn(exportBucketArn)
                    .withEncryption(exportEncryption)
                    .withFormat(exportFormat)
                    .withOutputSchemaVersion(OutputSchemaVersion.V_1)
                    .withPrefix("Prefix");
            CloudWatchMetrics cloudWatchMetrics = new CloudWatchMetrics()
                    .withIsEnabled(true);
            StorageLensDataExport dataExport = new StorageLensDataExport()
                    .withCloudWatchMetrics(cloudWatchMetrics)
                    .withS3BucketDestination(s3BucketDestination);

            StorageLensAwsOrg awsOrg = new StorageLensAwsOrg()
                    .withArn(awsOrgARN);

            StorageLensConfiguration configuration = new StorageLensConfiguration()
                    .withId(configurationId)
                    .withAccountLevel(accountLevel)
                    .withInclude(include)
                    .withDataExport(dataExport)
                    .withAwsOrg(awsOrg)
                    .withIsEnabled(true);

            List<StorageLensTag> tags = Arrays.asList(
                    new StorageLensTag().withKey("key-1").withValue("value-1"),
                    new StorageLensTag().withKey("key-2").withValue("value-2")
            );

            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.putStorageLensConfiguration(new
PutStorageLensConfigurationRequest()
                    .withAccountId(sourceAccountId)
                    .withConfigId(configurationId)
                    .withStorageLensConfiguration(configuration)
                    .withTags(tags)
```

```
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Untuk mendapatkan visibilitas lebih lanjut ke penyimpanan, Anda dapat membuat satu atau beberapa grup Lensa Penyimpanan S3 dan melampirkannya ke dasbor. Grup Lensa Penyimpanan S3 adalah filter yang ditentukan khusus untuk objek berdasarkan awalan, akhiran, tag objek, ukuran objek, usia objek, atau kombinasi dari semua filter tersebut.

Anda dapat menggunakan grup Lensa Penyimpanan S3 untuk mendapatkan visibilitas terperinci ke dalam bucket bersama yang bersar, seperti danau data, untuk membuat keputusan bisnis yang lebih terinformasi. Misalnya, Anda dapat menyederhanakan alokasi penyimpanan dan mengoptimalkan pelaporan biaya dengan mengelompokkan penggunaan penyimpanan ke grup objek tertentu untuk masing-masing proyek dan pusat biaya dalam satu bucket atau beberapa bucket.

Untuk menggunakan grup Lensa Penyimpanan S3, Anda harus meningkatkan dasbor agar metrik dan rekomendasi tingkat lanjut dapat digunakan. Untuk informasi selengkapnya tentang grup Lensa Penyimpanan S3, lihat the section called “Bekerja dengan grup Storage Lens”.

Perbarui dasbor Lensa Penyimpanan Amazon S3

Dasbor default Amazon S3 Storage Lens adalah. `default-account-dashboard` Dasbor ini telah dikonfigurasi sebelumnya oleh Amazon S3 untuk membantu memvisualisasikan ringkasan wawasan dan tren untuk gabungan metrik gratis dan lanjutan dari seluruh akun di konsol. Anda tidak dapat mengubah cakupan konfigurasi dasbor default, tapi Anda dapat meningkatkan pilihan metrik dari metrik gratis ke metrik dan rekomendasi tingkat lanjut berbayar, mengonfigurasi ekspor metrik opsional, atau bahkan menonaktifkan dasbor default. Dasbor default tidak dapat dihapus, dan hanya dapat dinonaktifkan. Untuk informasi selengkapnya, lihat Menggunakan konsol S3.

Menggunakan konsol S3

Gunakan langkah-langkah berikut untuk memperbarui dasbor Lensa Penyimpanan Amazon S3 di konsol Amazon S3.

Langkah 1: Perbarui cakupan dasbor

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, pilih Lensa Penyimpanan, Dasbor.
3. Pilih dasbor yang ingin diedit, lalu pilih Edit.

   Laman Edit dasbor akan terbuka.

   > **Note**
   >
   > Anda tidak dapat mengubah hal-hal berikut:
   >
   > - Nama dasbor
   > - Wilayah asal
   > - Cakupan dasbor dari dasbor default, yang mencakup seluruh penyimpanan akun Anda

4. (Opsional) Di laman konfigurasi dasbor, di bagian Umum, perbarui dan tambahkan tag ke dasbor Anda.

   Anda dapat menggunakan tag untuk mengelola izin untuk dasbor Anda dan melacak biaya untuk Lensa Penyimpanan S3. Untuk informasi lebih lanjut, lihat Mengontrol akses menggunakan tag sumber daya di Panduan Pengguna IAM dan AWS Tag Alokasi Biaya yang Dihasilkan di AWS Billing Panduan Pengguna.

   > **Note**
   >
   > Anda dapat menambahkan hingga 50 tag ke konfigurasi dasbor.

5. Di bagian Cakupan dasbor, lakukan hal-hal berikut:

   a. Perbarui Wilayah dan bucket yang Anda inginkan untuk disertakan atau dikecualikan oleh Lensa Penyimpanan S3 di dasbor.

> **Note**
>
> - Anda dapat menyertakan atau mengecualikan Wilayah dan bucket. Pilihan ini terbatas pada Wilayah saja saat membuat dasbor tingkat organisasi di seluruh akun anggota di organisasi Anda.
> - Anda dapat memilih hingga 50 bucket untuk disertakan atau dikecualikan.

b. Perbarui bucket di Wilayah terpilih yang Anda inginkan untuk disertakan atau dikecualikan oleh Lensa Penyimpanan S3. Anda dapat menyertakan atau mengecualikan bucket, tetapi tidak keduanya. Opsi ini tidak tersedia saat dasbor tingkat organisasi dibuat.

Langkah 2: Perbarui pilihan metrik

1. Di bagian Pilihan metrik, pilih jenis metrik yang ingin digabungkan untuk dasbor ini.

   - Untuk menyertakan metrik gratis yang digabungkan di tingkat bucket dan tersedia untuk kueri selama 14 hari, pilih Metrik gratis.
   - Untuk dapat mengaktifkan metrik lanjutan dan opsi lanjutan lainnya, pilih Metrik dan rekomendasi lanjutan. Opsi ini mencakup agregasi awalan lanjutan, CloudWatch penerbitan Amazon, dan rekomendasi kontekstual. Data tersedia untuk kueri selama 15 bulan. Metrik dan rekomendasi lanjutan memiliki biaya tambahan. Untuk informasi selengkapnya, lihat Harga Amazon S3.

     Untuk informasi selengkapnya tentang metrik lanjutan dan metrik gratis, lihat Pilihan metrik.

2. Di bagian bawah Fitur metrik dan rekomendasi lanjutan, pilih opsi yang ingin diaktifkan:

   - Metrik lanjutan
   - CloudWatch penerbitan
   - Agregasi awalan

> **Important**
>
> Jika Anda mengaktifkan agregasi awalan untuk konfigurasi Lensa Penyimpanan S3 Anda, metrik tingkat awalan tidak akan dipublikasikan. CloudWatch Hanya metrik Lensa Penyimpanan S3 tingkat bucket, akun, dan organisasi yang dipublikasikan. CloudWatch

3. Jika Anda mengaktifkan Metrik lanjutan, pilih Kategori metrik lanjutan yang ingin ditampilkan di dasbor Lensa Penyimpanan S3:

   - Metrik aktivitas
   - Metrik kode status terperinci
   - Metrik pengoptimalan biaya lanjutan
   - Metrik perlindungan data lanjutan

   Untuk informasi selengkapnya tentang kategori metrik, lihat Kategori metrik. Untuk daftar lengkap metrik, lihat Glosarium metrik Lensa Penyimpanan Amazon S3.

4. Jika Anda memilih untuk mengaktifkan agregasi awalan, konfigurasikan hal-hal berikut ini:

   a. Pilih ukuran ambang batas awalan minimum untuk dasbor ini.

      Misalnya, ambang batas awalan sebesar 5 persen menunjukkan bahwa awalan yang membentuk 5 persen atau lebih dari total ukuran penyimpanan bucket akan digabungkan.

   b. Pilih kedalaman awalan.

      Pengaturan ini menunjukkan jumlah maksimum level hingga awalan dapat dievaluasi. Kedalaman awalan harus kurang dari 10.

   c. Masukkan karakter pembatas awalan.

      Ini adalah nilai yang digunakan untuk mengidentifikasi setiap tingkat awalan. Nilai default di Amazon S3 adalah karakter `/`, tapi struktur penyimpanan Anda mungkin akan menggunakan karakter pembatas lain.

(Opsional) Langkah 3: Ekspor metrik untuk dasbor

1. Di bagian Ekspor metrik, untuk membuat ekspor metrik yang akan ditempatkan setiap hari di bucket tujuan pilihan Anda, pilih Aktifkan. Untuk menonaktifkan ekspor metrik, pilih Nonaktifkan.

   Ekspor metrik berlaku dalam format CSV atau Apache Parquet. Ketentuan ini mewakili cakupan data yang sama dengan data dasbor Lensa Penyimpanan S3 Anda tanpa rekomendasi.

2. Jika sudah diaktifkan, pilih format output untuk ekspor metrik harian Anda: CSV atau Apache Parquet.

   Parquet adalah format file sumber terbuka untuk Hadoop yang menyimpan data bertumpuk dalam format kolom datar.

3. Pilih bucket S3 tujuan untuk ekspor metrik Anda.

   Anda dapat memilih bucket di akun saat ini di dasbor Lensa Penyimpanan S3. Atau Anda dapat memilih yang lain Akun AWS jika Anda memiliki izin bucket tujuan dan ID akun pemilik bucket tujuan.

4. Pilih bucket S3 tujuan (format: `s3://bucket-name/prefix`).

   Bucket harus berada di Wilayah asal dasbor Lensa Penyimpanan S3 Anda. Konsol S3 menampilkan Izin bucket tujuan yang akan ditambahkan oleh Amazon S3 ke kebijakan bucket tujuan. Amazon S3 memperbarui kebijakan bucket di bucket tujuan agar S3 dapat menempatkan data di bucket tersebut.

5. (Opsional) Untuk mengaktifkan enkripsi di sisi server untuk ekspor metrik Anda, pilih Tentukan kunci enkripsi. Kemudian, pilih Jenis enkripsi: Kunci terkelola Amazon S3 (SSE-S3) atau AWS Key Management Service kunci (SSE-KMS).

   Anda dapat memilih antara kunci terkelola Amazon S3 (SSE-S3) dan kunci AWS Key Management Service (AWS KMS) (SSE-KMS).

6. (Opsional) Untuk menentukan AWS KMS kunci, Anda harus memilih kunci KMS atau memasukkan kunci Nama Sumber Daya Amazon (ARN). Di bawah AWS KMS kunci, tentukan kunci KMS Anda dengan salah satu cara berikut ini:

   - Untuk memilih dari daftar kunci KMS yang tersedia, pilih Pilih dari AWS KMS keys, dan pilih Kunci KMS Anda dari daftar kunci yang tersedia.

Kunci Kunci yang dikelola AWS (`aws/s3`) dan kunci terkelola pelanggan Anda muncul dalam daftar ini. Untuk informasi selengkapnya tentang CMK, lihat Kunci pelanggan dan AWS kunci di AWS Key Management Service Panduan Pengembang.

> **Note**
>
> The Kunci yang dikelola AWS (`aws/S3`) tidak didukung untuk enkripsi SSE-KMS dengan S3 Storage Lens.

- Untuk memasukkan ARN kunci KMS, pilih Masukkan ARN, dan masukkan AWS KMS key ARN kunci KMS Anda di bidang yang muncul.
- Untuk membuat kunci terkelola pelanggan baru di AWS KMS konsol, pilih Buat kunci KMS.

  Jika Anda memilih CMK, Anda harus memberikan izin Lensa Penyimpanan S3 untuk dapat mengenkripsi di dalam kebijakan kunci AWS KMS . Untuk informasi selengkapnya, lihat Menggunakan AWS KMS key untuk mengenkripsi ekspor metrik Anda.

  Untuk informasi selengkapnya tentang membuat AWS KMS key, lihat Membuat Kunci di Panduan AWS Key Management Service Pengembang.

7. Pilih Simpan perubahan.

Untuk mendapatkan visibilitas lebih lanjut ke penyimpanan, Anda dapat membuat satu atau beberapa grup Lensa Penyimpanan S3 dan melampirkannya ke dasbor. Grup Lensa Penyimpanan S3 adalah filter yang ditentukan khusus untuk objek berdasarkan awalan, akhiran, tag objek, ukuran objek, usia objek, atau kombinasi dari semua filter tersebut.

Anda dapat menggunakan grup Lensa Penyimpanan S3 untuk mendapatkan visibilitas terperinci ke dalam bucket bersama yang bersar, seperti danau data, untuk membuat keputusan bisnis yang lebih terinformasi. Misalnya, Anda dapat menyederhanakan alokasi penyimpanan dan mengoptimalkan pelaporan biaya dengan mengelompokkan penggunaan penyimpanan ke grup objek tertentu untuk masing-masing proyek dan pusat biaya dalam satu bucket atau beberapa bucket.

Untuk menggunakan grup Lensa Penyimpanan S3, Anda harus meningkatkan dasbor agar metrik dan rekomendasi tingkat lanjut dapat digunakan. Untuk informasi selengkapnya tentang grup Lensa Penyimpanan S3, lihat the section called “Bekerja dengan grup Storage Lens”.

Menggunakan AWS CLI

Example

Perintah contoh berikut memperbarui konfigurasi dasbor Lensa Penyimpanan Amazon S3. Untuk menggunakan contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3control put-storage-lens-configuration --account-id=111122223333 --
config-id=example-dashboard-configuration-id --region=us-east-1 --storage-lens-
configuration=file://./config.json --tags=file://./tags.json
```

Menggunakan AWS SDK for Java

Example — Perbarui konfigurasi Lensa Penyimpanan Amazon S3 dengan metrik dan rekomendasi lanjutan

Contoh berikut menunjukkan cara memperbarui konfigurasi Lensa Penyimpanan S3 default dengan metrik dan rekomendasi lanjutan di SDK for Java:

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.AccountLevel;
import com.amazonaws.services.s3control.model.ActivityMetrics;
import com.amazonaws.services.s3control.model.BucketLevel;
import com.amazonaws.services.s3control.model.Format;
import com.amazonaws.services.s3control.model.Include;
import com.amazonaws.services.s3control.model.OutputSchemaVersion;
import com.amazonaws.services.s3control.model.PrefixLevel;
import com.amazonaws.services.s3control.model.PrefixLevelStorageMetrics;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.S3BucketDestination;
import com.amazonaws.services.s3control.model.SSES3;
import com.amazonaws.services.s3control.model.SelectionCriteria;
import com.amazonaws.services.s3control.model.StorageLensAwsOrg;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensDataExport;
import com.amazonaws.services.s3control.model.StorageLensDataExportEncryption;
import com.amazonaws.services.s3control.model.StorageLensTag;
```

```
import java.util.Arrays;
import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class UpdateDefaultConfigWithPaidFeatures {

    public static void main(String[] args) {
        String configurationId = "default-account-dashboard"; // This configuration ID
 cannot be modified.
        String sourceAccountId = "111122223333";

        try {
            SelectionCriteria selectionCriteria = new SelectionCriteria()
                    .withDelimiter("/")
                    .withMaxDepth(5)
                    .withMinStorageBytesPercentage(10.0);
            PrefixLevelStorageMetrics prefixStorageMetrics = new
 PrefixLevelStorageMetrics()
                    .withIsEnabled(true)
                    .withSelectionCriteria(selectionCriteria);
            BucketLevel bucketLevel = new BucketLevel()
                    .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
                    .withPrefixLevel(new
 PrefixLevel().withStorageMetrics(prefixStorageMetrics));
            AccountLevel accountLevel = new AccountLevel()
                    .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
                    .withBucketLevel(bucketLevel);

            StorageLensConfiguration configuration = new StorageLensConfiguration()
                    .withId(configurationId)
                    .withAccountLevel(accountLevel)
                    .withIsEnabled(true);

            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.putStorageLensConfiguration(new
 PutStorageLensConfigurationRequest()
                    .withAccountId(sourceAccountId)
                    .withConfigId(configurationId)
```

```
                    .withStorageLensConfiguration(configuration)
            );

        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

> **Note**
>
> Biaya tambahan berlaku untuk metrik dan rekomendasi tingkat lanjut. Untuk informasi selengkapnya, lihat metrik dan rekomendasi tingkat lanjut.

Nonaktifkan dasbor Lensa Penyimpanan Amazon S3

Anda dapat menonaktifkan dasbor Lensa Penyimpanan Amazon S3 dari konsol Amazon S3. Menonaktifkan dasbor mencegahnya menghasilkan metrik di masa mendatang. Dasbor yang dinonaktifkan masih dapat mempertahankan informasi konfigurasinya, sehingga dasbor dapat dilanjutkan dengan mudah ketika diaktifkan kembali. Dasbor yang dinonaktifkan akan mempertahankan data historisnya hingga tidak lagi tersedia untuk kueri.

Menggunakan konsol S3

Gunakan langkah-langkah berikut untuk menonaktifkan dasbor Lensa Penyimpanan Amazon S3 di konsol Amazon S3.

Untuk menonaktifkan dasbor Lensa Penyimpanan Amazon S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, pilih Lensa Penyimpanan, Dasbor.

3. Di daftar Dasbor, pilih dasbor yang ingin dinonaktifkan, lalu pilih Nonaktifkan di bagian atas daftar.
4. Di laman konfirmasi, lakukan konfirmasi bahwa Anda ingin menonaktifkan dasbor dengan memasukkan nama dasbor ke dalam kolom teks, lalu pilih Konfirmasi.

Hapus dasbor Lensa Penyimpanan Amazon S3

Anda tidak dapat menghapus dasbor default. Namun, Anda dapat menonaktifkannya. Sebelum menghapus dasbor yang sudah dibuat, pertimbangkan hal-hal berikut:

- Sebagai alternatif dari menghapus dasbor, Anda dapat menonaktifkan dasbor sehingga dasbor tersebut dapat tersedia untuk diaktifkan kembali di masa mendatang. Untuk informasi selengkapnya, lihat Menggunakan konsol S3.
- Menghapus dasbor akan menghapus semua pengaturan konfigurasi yang terkait dengan dasbor.
- Menghapus dasbor akan membuat semua data metrik historis tidak tersedia. Data historis ini masih dapat dipertahankan selama 15 bulan. Untuk mengakses data ini lagi, buat dasbor dengan nama yang sama di Wilayah asal yang sama dengan yang sudah dihapus.

Menggunakan konsol S3

Anda dapat menghapus dasbor Lensa Penyimpanan Amazon S3 dari konsol Amazon S3. Namun, menghapus dasbor mencegahnya menghasilkan metrik di masa mendatang.

Menghapus dasbor Lensa Penyimpanan Amazon S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, pilih Lensa Penyimpanan, Dasbor.
3. Di daftar Dasbor, pilih dasbor yang ingin dihapus, lalu pilih Hapus di bagian atas daftar.
4. Di laman Hapus dasbor, lakukan konfirmasi bahwa Anda ingin menghapus dasbor dengan memasukkan nama dasbor ke kolom teks. Lalu, pilih Konfirmasi.

Menggunakan AWS CLI

Example

Contoh berikut menghapus konfigurasi S3 Storage Lens. Untuk menggunakan contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3control delete-storage-lens-configuration --account-id=222222222222 --region=us-east-1 --config-id=your-configuration-id
```

Menggunakan AWS SDK for Java

Example — Hapus konfigurasi dasbor Lensa Penyimpanan Amazon S3

Contoh berikut menunjukkan cara menghapus konfigurasi S3 Storage Lens menggunakan SDK for Java:

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.DeleteStorageLensConfigurationRequest;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class DeleteDashboard {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "111122223333";
        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.deleteStorageLensConfiguration(new
 DeleteStorageLensConfigurationRequest()
                    .withAccountId(sourceAccountId)
```

```
                    .withConfigId(configurationId)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Daftar dasbor Lensa Penyimpanan Amazon S3

Menggunakan konsol S3

Untuk membuat daftar dasbor Lensa Penyimpanan S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, navigasikan ke Storage Lens.
3. Pilih Dasbor. Anda sekarang dapat melihat dasbor di. Akun AWS

Menggunakan AWS CLI

Example

Contoh perintah berikut mencantumkan dasbor S3 Storage Lens di dasbor Anda. Akun AWS Untuk menggunakan contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3control list-storage-lens-configurations --account-id=222222222222 --region=us-east-1 --next-token=abcdefghij1234
```

Example

Contoh berikut mencantumkan konfigurasi S3 Storage Lens tanpa token berikutnya. Untuk menggunakan contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3control list-storage-lens-configurations --account-id=222222222222 --region=us-
east-1
```

Menggunakan AWS SDK for Java

Example — Daftar konfigurasi dasbor Lensa Penyimpanan S3

Contoh berikut menunjukkan kepada Anda cara membuat daftar konfigurasi Lensa Penyimpanan S3 di SDK for Java. Untuk menggunakan contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.” untuk setiap contoh deskripsi.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.ListStorageLensConfigurationEntry;
import com.amazonaws.services.s3control.model.ListStorageLensConfigurationsRequest;

import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class ListDashboard {

    public static void main(String[] args) {
        String sourceAccountId = "111122223333";
        String nextToken = "nextToken";

        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            final List<ListStorageLensConfigurationEntry> configurations =
                    s3ControlClient.listStorageLensConfigurations(new
 ListStorageLensConfigurationsRequest()
                            .withAccountId(sourceAccountId)
                            .withNextToken(nextToken)
                    ).getStorageLensConfigurationList();
```

```
            System.out.println(configurations.toString());
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Melihat detail konfigurasi dasbor Amazon S3 Storage Lens

Anda dapat melihat dasbor Amazon S3 Storage Lens dari konsol Amazon S3 AWS CLI, dan SDK for Java.

Menggunakan konsol S3

Untuk melihat detail konfigurasi dasbor Lensa Penyimpanan S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, navigasikan ke Storage Lens.
3. Pilih Dasbor.
4. Dari daftar Dasbor, klik dasbor yang ingin Anda lihat. Anda sekarang dapat melihat detail dasbor Lensa Penyimpanan Anda.

Menggunakan AWS CLI

Example

Contoh berikut mengambil konfigurasi S3 Storage Lens sehingga Anda dapat melihat detail konfigurasi. Untuk menggunakan contoh ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3control get-storage-lens-configuration --account-id=222222222222 --config-
id=your-configuration-id --region=us-east-1
```

Menggunakan AWS SDK for Java

Example — Mengambil dan melihat konfigurasi Lensa Penyimpanan S3

Contoh berikut menunjukkan cara mengambil konfigurasi S3 Storage Lens di SDK for Java sehingga Anda dapat melihat detail konfigurasi. Untuk menggunakan contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.GetStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.GetStorageLensConfigurationResult;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class GetDashboard {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "111122223333";

        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            final StorageLensConfiguration configuration =
                    s3ControlClient.getStorageLensConfiguration(new
 GetStorageLensConfigurationRequest()
                            .withAccountId(sourceAccountId)
                            .withConfigId(configurationId)
                    ).getStorageLensConfiguration();

            System.out.println(configuration.toString());
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
```

```
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Mengelola tag AWS sumber daya dengan Lensa Penyimpanan S3

Setiap dasbor Lensa Penyimpanan Amazon S3 dihitung sebagai AWS sumber daya dengan Nama Sumber Daya Amazon (ARN) miliknya sendiri. Oleh karena itu, saat Anda mengonfigurasi dasbor Lensa Penyimpanan, Anda dapat menambahkan tag AWS sumber daya secara opsional ke dasbor. Anda dapat menambahkan hingga 50 tag untuk setiap dasbor Lensa Penyimpanan. Untuk membuat dasbor Lensa Penyimpanan dengan tag, Anda harus memiliki izin Lensa Penyimpanan S3 berikut:

- `s3:ListStorageLensConfigurations`
- `s3:GetStorageLensConfiguration`
- `s3:GetStorageLensConfigurationTagging`
- `s3:PutStorageLensConfiguration`
- `s3:PutStorageLensConfigurationTagging`

Anda dapat menggunakan tag AWS sumber daya untuk mengkategorikan sumber daya menurut departemen, lini bisnis, atau proyek. Hal ini berguna jika Anda memiliki banyak sumber daya dengan jenis yang sama. Dengan menerapkan tag, Anda dapat dengan cepat mengidentifikasi dasbor Lensa Penyimpanan S3 tertentu berdasarkan tag yang telah Anda tetapkan padanya. Anda juga dapat menggunakan tag untuk melacak dan mengalokasikan biaya.

Selain itu, saat menambahkan tag AWS sumber daya ke dasbor Lensa Penyimpanan, Anda mengaktifkan kontrol akses berbasis atribut (ABAC). ABAC adalah strategi otorisasi yang mendefinisikan izin berdasarkan atribut seperti tag. Anda juga dapat menggunakan kondisi yang menentukan tag sumber daya dalam kebijakan IAM Anda untuk mengontrol akses ke AWS sumber daya.

Anda dapat mengedit kunci dan nilai tanda, dan Anda dapat menghapus tanda dari sumber daya kapan pun. Selain itu, perhatikan batasan berikut:

- Kunci tag dan nilai tag peka huruf besar-kecil.
- Jika Anda menambahkan tanda yang memiliki kunci yang sama dengan tanda yang sudah ada di sumber daya tersebut, nilai baru akan menimpa nilai lama.
- Jika sumber daya dihapus, semua tanda untuk sumber daya tersebut juga akan dihapus.
- Jangan sertakan data pribadi atau sensitif dalam tag AWS sumber daya Anda.
- Tag sistem (dengan kunci tag yang dimulai dengan `aws:`) tidak didukung.
- Panjang setiap kunci tag tidak boleh melebihi 128 karakter. Panjang setiap nilai tag tidak boleh melebihi 256 karakter.

Contoh berikut menunjukkan cara menggunakan tag AWS sumber daya dengan dasbor Storage Lens.

Topik

- Tambahkan tag AWS sumber daya ke dasbor Lensa Penyimpanan
- Mengambil tag AWS sumber daya untuk dasbor Lensa Penyimpanan
- Memperbarui tag dasbor Lensa Penyimpanan
- Menghapus tag AWS sumber daya dari dasbor Lensa Penyimpanan S3

Tambahkan tag AWS sumber daya ke dasbor Lensa Penyimpanan

Contoh berikut menunjukkan cara menambahkan tag AWS sumber daya ke dasbor Lensa Penyimpanan S3. Anda dapat menambahkan tag sumber daya dengan menggunakan konsol Amazon S3, AWS Command Line Interface (AWS CLI), dan. AWS SDK untuk Java

Menggunakan konsol S3

Untuk menambahkan tag AWS sumber daya ke dasbor Lensa Penyimpanan

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, arahkan ke Storage Lens di panel navigasi kiri.
3. Pilih Dasbor.
4. Pilih tombol radio untuk dasbor Storage Lens yang ingin Anda perbarui. Kemudian pilih Edit.
5. Di bawah Umum, pilih Tambah tag.
6. Pada halaman Tambah tag, tambahkan pasangan kunci-nilai baru.

> **Note**
>
> Menambahkan tag baru yang memiliki kunci sama dengan tag yang sudah ada akan menimpa nilai tag sebelumnya.

7. (Opsional) Untuk menambahkan lebih dari satu tag baru, pilih Tambah tag lagi untuk melanjutkan penambahan entri baru. Anda dapat menambahkan hingga 50 tag AWS sumber daya ke dasbor Lensa Penyimpanan Anda.
8. (Opsional) Jika Anda ingin menghapus entri yang baru ditambahkan, pilih Hapus di bagian samping tag yang ingin dihapus.
9. Pilih Simpan perubahan.

Menggunakan AWS CLI

Example

Perintah contoh berikut menambahkan tag ke konfigurasi dasbor S3 Storage Lens. Untuk menggunakan contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3control put-storage-lens-configuration-tagging --account-id=222222222222 --
region=us-east-1 --config-id=your-configuration-id --tags=file://./tags.json
```

Menggunakan AWS SDK for Java

Contoh berikut menambahkan tag ke konfigurasi Amazon S3 Storage Lens di SDK for Java. Untuk menggunakan contoh ini, ganti *user input placeholders*dengan informasi Anda sendiri.

Example — Tambahkan tag ke konfigurasi Lensa Penyimpanan S3

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import
 com.amazonaws.services.s3control.model.PutStorageLensConfigurationTaggingRequest;
import com.amazonaws.services.s3control.model.StorageLensTag;
```

```
import java.util.Arrays;
import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class PutDashboardTagging {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "111122223333";

        try {
            List<StorageLensTag> tags = Arrays.asList(
                    new StorageLensTag().withKey("key-1").withValue("value-1"),
                    new StorageLensTag().withKey("key-2").withValue("value-2")
            );

            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.putStorageLensConfigurationTagging(new
 PutStorageLensConfigurationTaggingRequest()
                    .withAccountId(sourceAccountId)
                    .withConfigId(configurationId)
                    .withTags(tags)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Mengambil tag AWS sumber daya untuk dasbor Lensa Penyimpanan

Contoh berikut menunjukkan cara mengambil tag AWS sumber daya untuk dasbor Lensa Penyimpanan S3. Anda bisa mendapatkan tag sumber daya dengan menggunakan konsol Amazon S3, AWS Command Line Interface (AWS CLI), dan. AWS SDK untuk Java

Menggunakan konsol S3

Untuk mengambil tag AWS sumber daya untuk dasbor Lensa Penyimpanan

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, navigasikan ke Storage Lens.
3. Pilih Dasbor.
4. Pilih tombol radio untuk konfigurasi dasbor Lensa Penyimpanan yang ingin Anda lihat. Kemudian, pilih Lihat konfigurasi dasbor.
5. Di bawah Tag, tinjau tag yang terkait dengan dasbor.
6. (Opsional) Jika Anda ingin menambahkan tag baru, pilih Edit. Kemudian, pilih Tambah tag. Pada halaman Tambah tag, tambahkan pasangan kunci-nilai baru.

   > **Note**
   >
   > Menambahkan tag baru yang memiliki kunci sama dengan tag yang sudah ada akan menimpa nilai tag sebelumnya.

7. (Opsional) Jika Anda ingin menghapus entri yang baru ditambahkan, pilih Hapus di bagian samping tag yang ingin dihapus.
8. Pilih Simpan perubahan.

Menggunakan AWS CLI

Example

Contoh perintah berikut mengambil tag untuk konfigurasi dashboard S3 Storage Lens. Untuk menggunakan contoh ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3control get-storage-lens-configuration-tagging --account-id=222222222222 --
region=us-east-1 --config-id=your-configuration-id --tags=file://./tags.json
```

Menggunakan AWS SDK for Java

Example — Dapatkan tag untuk konfigurasi dasbor Lensa Penyimpanan S3

Contoh berikut menunjukkan cara mengambil tag untuk konfigurasi dasbor Lensa Penyimpanan S3 di SDK for Java. Untuk menggunakan contoh ini, ganti *user input placeholders*dengan informasi Anda sendiri.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.DeleteStorageLensConfigurationRequest;
import
 com.amazonaws.services.s3control.model.GetStorageLensConfigurationTaggingRequest;
import com.amazonaws.services.s3control.model.StorageLensTag;

import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class GetDashboardTagging {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "111122223333";
        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            final List<StorageLensTag> s3Tags = s3ControlClient
                    .getStorageLensConfigurationTagging(new
 GetStorageLensConfigurationTaggingRequest()
                            .withAccountId(sourceAccountId)
                            .withConfigId(configurationId)
                    ).getTags();

            System.out.println(s3Tags.toString());
```

```
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Memperbarui tag dasbor Lensa Penyimpanan

Contoh berikut menunjukkan cara memperbarui tag dasbor Lensa Penyimpanan dengan menggunakan konsol Amazon S3, AWS Command Line Interface (AWS CLI), dan. AWS SDK untuk Java

Menggunakan konsol S3

Untuk memperbarui tag AWS sumber daya untuk dasbor Lensa Penyimpanan

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, navigasikan ke Storage Lens.
3. Pilih Dasbor.
4. Pilih tombol radio untuk konfigurasi dasbor Lensa Penyimpanan yang ingin Anda lihat. Kemudian, pilih Lihat konfigurasi dasbor.
5. Di bawah Tag, tinjau tag yang terkait dengan dasbor.
6. (Opsional) Jika Anda ingin menambahkan tag baru, pilih Edit. Kemudian, pilih Tambah tag. Pada halaman Tambah tag, tambahkan pasangan kunci-nilai baru.

   > **Note**
   >
   > Menambahkan tag baru yang memiliki kunci sama dengan tag yang sudah ada akan menimpa nilai tag sebelumnya.

7. (Opsional) Jika Anda ingin menghapus entri yang baru ditambahkan, pilih Hapus di bagian samping tag yang ingin dihapus.

8. Pilih Simpan perubahan.

Menggunakan AWS CLI

Example

Perintah contoh berikut menambahkan atau mengganti tag pada konfigurasi dasbor Lensa Penyimpanan Amazon S3 yang ada. Untuk menggunakan contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3control put-storage-lens-configuration-tagging --account-id=111122223333 --
config-id=example-dashboard-configuration-id --region=us-east-1 --config-id=your-
configuration-id
```

Menggunakan AWS SDK for Java

AWS SDK untuk Java Contoh berikut memperbarui tag AWS sumber daya pada dasbor Lensa Penyimpanan yang ada. Untuk menggunakan contoh ini, ganti *user input placeholders*dengan informasi Anda sendiri.

Example — Perbarui tag pada konfigurasi dasbor Lensa Penyimpanan yang ada

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import
 com.amazonaws.services.s3control.model.PutStorageLensConfigurationTaggingRequest;
import com.amazonaws.services.s3control.model.StorageLensTag;

import java.util.Arrays;
import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class PutDashboardTagging {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
```

```
        String sourceAccountId = "111122223333";

        try {
            List<StorageLensTag> tags = Arrays.asList(
                    new StorageLensTag().withKey("key-1").withValue("value-1"),
                    new StorageLensTag().withKey("key-2").withValue("value-2")
            );

            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.putStorageLensConfigurationTagging(new
 PutStorageLensConfigurationTaggingRequest()
                    .withAccountId(sourceAccountId)
                    .withConfigId(configurationId)
                    .withTags(tags)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Menghapus tag AWS sumber daya dari dasbor Lensa Penyimpanan S3

Contoh berikut menunjukkan cara menghapus tag AWS sumber daya dari dasbor Lensa Penyimpanan yang ada. Anda dapat menghapus tag dengan menggunakan konsol Amazon S3, AWS Command Line Interface (AWS CLI), dan. AWS SDK untuk Java

Menggunakan konsol S3

Untuk menghapus tag AWS sumber daya dari dasbor Lensa Penyimpanan yang ada

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

2. Di panel navigasi kiri, navigasikan ke Storage Lens.
3. Pilih Dasbor.
4. Pilih tombol radio untuk konfigurasi dasbor Lensa Penyimpanan yang ingin Anda lihat. Kemudian, pilih Lihat konfigurasi dasbor.
5. Di bawah Tag, tinjau tag yang terkait dengan dasbor.
6. Pilih Hapus di samping tag yang ingin Anda hapus.
7. Pilih Simpan perubahan.

Menggunakan AWS CLI

AWS CLI Perintah berikut menghapus tag AWS sumber daya dari dasbor Lensa Penyimpanan yang ada. Untuk menggunakan perintah contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

Example

```
aws s3control delete-storage-lens-configuration-tagging --account-id=222222222222 --
config-id=your-configuration-id --region=us-east-1
```

Menggunakan AWS SDK for Java

AWS SDK untuk Java Contoh berikut menghapus tag AWS sumber daya dari dasbor Lensa Penyimpanan menggunakan Amazon Resource Name (ARN) yang Anda tentukan di akun. *111122223333* Untuk menggunakan contoh ini, ganti *user input placeholders*dengan informasi Anda sendiri.

Example — Hapus tag untuk konfigurasi dasbor Lensa Penyimpanan S3

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import
 com.amazonaws.services.s3control.model.DeleteStorageLensConfigurationTaggingRequest;
```

```
import static com.amazonaws.regions.Regions.US_WEST_2;

public class DeleteDashboardTagging {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "111122223333";
        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.deleteStorageLensConfigurationTagging(new
 DeleteStorageLensConfigurationTaggingRequest()
                    .withAccountId(sourceAccountId)
                    .withConfigId(configurationId)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

File pembantu untuk menggunakan Lensa Penyimpanan Amazon S3

Gunakan file JSON berikut dan input kuncinya untuk contoh Anda.

Konfigurasi contoh Lensa Penyimpanan S3 di JSON

Example **`config.json`**

File `config.json` tersebut berisi detail metrik dan konfigurasi rekomendasi tingkat lanjut tingkat Organisasi Lensa Penyimpanan S3. Untuk menggunakan contoh berikut, ganti *`user input placeholders`* dengan informasi Anda sendiri.

> **Note**
>
> Biaya tambahan berlaku untuk metrik dan rekomendasi tingkat lanjut. Untuk informasi selengkapnya, lihat metrik dan rekomendasi tingkat lanjut.

```
{
  "Id": "SampleS3StorageLensConfiguration", //Use this property to identify your S3
 Storage Lens configuration.
  "AwsOrg": { //Use this property when enabling S3 Storage Lens for AWS Organizations.
    "Arn": "arn:aws:organizations::123456789012:organization/o-abcdefgh"
  },
  "AccountLevel": {
    "ActivityMetrics": {
      "IsEnabled":true
    },
    "AdvancedCostOptimizationMetrics": {
      "IsEnabled":true
    },
    "AdvancedDataProtectionMetrics": {
      "IsEnabled":true
    },
    "DetailedStatusCodesMetrics": {
      "IsEnabled":true
    },
    "BucketLevel": {
      "ActivityMetrics": {
        "IsEnabled":true
      },
      "AdvancedDataProtectionMetrics": {
      "IsEnabled":true
      },
      "AdvancedCostOptimizationMetrics": {
        "IsEnabled":true
      },
      "DetailedStatusCodesMetrics": {
        "IsEnabled":true
      },
      "PrefixLevel":{
        "StorageMetrics":{
          "IsEnabled":true,
          "SelectionCriteria":{
```

```
            "MaxDepth":5,
            "MinStorageBytesPercentage":1.25,
            "Delimiter":"/"
          }
        }
      }
    }
  },
  "Exclude": { //Replace with "Include" if you prefer to include Regions.
    "Regions": [
      "eu-west-1"
    ],
    "Buckets": [ //This attribute is not supported for AWS Organizations-level
 configurations.
      "arn:aws:s3:::amzn-s3-demo-source-bucket"
    ]
  },
  "IsEnabled": true, //Whether the configuration is enabled
  "DataExport": { //Details about the metrics export
    "S3BucketDestination": {
      "OutputSchemaVersion": "V_1",
      "Format": "CSV", //You can add "Parquet" if you prefer.
      "AccountId": "111122223333",
      "Arn": "arn:aws:s3:::
amzn-s3-demo-destination-bucket", // The destination bucket for your metrics export
 must be in the same Region as your S3 Storage Lens configuration.
      "Prefix": "prefix-for-your-export-destination",
      "Encryption": {
        "SSES3": {}
      }
    },
    "CloudWatchMetrics": {
      "IsEnabled": true
    }
  }
}
```

Konfigurasi contoh Lensa Penyimpanan S3 dengan grup Lensa Penyimpanan di JSON

Example **`config.json`**

File `config.json` berisi detail yang ingin Anda terapkan ke konfigurasi Lensa Penyimpanan Anda saat menggunakan grup Lensa Penyimpanan. Untuk menggunakan contoh, ganti *`user input placeholders`* dengan informasi Anda sendiri.

Untuk memasang semua grup Lensa Penyimpanan ke dasbor Anda, perbarui konfigurasi Lensa Penyimpanan Anda dengan sintaks berikut:

```
{
  "Id": "ExampleS3StorageLensConfiguration",
  "AccountLevel": {
    "ActivityMetrics": {
      "IsEnabled":true
    },
    "AdvancedCostOptimizationMetrics": {
      "IsEnabled":true
    },
    "AdvancedDataProtectionMetrics": {
      "IsEnabled":true
    },
    "BucketLevel": {
      "ActivityMetrics": {
      "IsEnabled":true
      },
    "StorageLensGroupLevel": {},
  "IsEnabled": true
}
```

Untuk menyertakan hanya dua grup Lensa Penyimpanan dalam konfigurasi dasbor Lensa Penyimpanan (*`slg-1`*dan*`slg-2`*), gunakan sintaks berikut:

```
{
  "Id": "ExampleS3StorageLensConfiguration",
  "AccountLevel": {
    "ActivityMetrics": {
      "IsEnabled":true
    },
    "AdvancedCostOptimizationMetrics": {
      "IsEnabled":true
```

```
    },
    "AdvancedDataProtectionMetrics": {
      "IsEnabled":true
    },
    "BucketLevel": {
      "ActivityMetrics": {
      "IsEnabled":true
      },
   "StorageLensGroupLevel": {
        "SelectionCriteria": {
            "Include": [
                "arn:aws:s3:us-east-1:111122223333:storage-lens-group/slg-1",
                "arn:aws:s3:us-east-1:444455556666:storage-lens-group/slg-2"
            ]
    },
  "IsEnabled": true
}
```

Untuk mengecualikan hanya grup Lensa Penyimpanan tertentu agar tidak dipasang ke konfigurasi dasbor Anda, gunakan sintaks berikut:

```
{
  "Id": "ExampleS3StorageLensConfiguration",
  "AccountLevel": {
    "ActivityMetrics": {
      "IsEnabled":true
    },
    "AdvancedCostOptimizationMetrics": {
      "IsEnabled":true
    },
    "AdvancedDataProtectionMetrics": {
      "IsEnabled":true
    },
    "BucketLevel": {
      "ActivityMetrics": {
      "IsEnabled":true
      },
   "StorageLensGroupLevel": {
        "SelectionCriteria": {
            "Exclude": [
                "arn:aws:s3:us-east-1:111122223333:storage-lens-group/slg-1",
                "arn:aws:s3:us-east-1:444455556666:storage-lens-group/slg-2"
            ]
```

```
    },
  "IsEnabled": true
}
```

Konfigurasi contoh tag Lensa Penyimpanan S3 di JSON

Example **`tags.json`**

File `tags.json` berisi tag yang ingin Anda terapkan ke konfigurasi Lensa Penyimpanan S3 Anda. Untuk menggunakan contoh ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
[
    {
        "Key": "key1",
        "Value": "value1"
    },
    {
        "Key": "key2",
        "Value": "value2"
    }
]
```

Konfigurasi contoh Lensa Penyimpanan S3 untuk izin IAM

Example **`permissions.json`**–Nama dasbor khusus

Kebijakan contoh ini menunjukkan file `permissions.json` IAM Lensa Penyimpanan S3 dengan nama dasbor tertentu yang ditentukan. Ganti *`value1`*, *`us-east-1`*, *`your-dashboard-name`*, dan *`example-account-id`* dengan nilai Anda sendiri.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetStorageLensConfiguration",
                "s3:DeleteStorageLensConfiguration",
                "s3:PutStorageLensConfiguration"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/key1": "value1"
```

```
                }
            },
            "Resource": "arn:aws:s3:us-east-1:example-account-id:storage-lens/your-
dashboard-name"
        }
    ]
}
```

Example **`permissions.json`**–Tidak ada nama dasbor khusus

Kebijakan contoh ini menunjukkan file `permissions.json` IAM Lensa Penyimpanan S3 tanpa nama dasbor tertentu yang ditentukan. Ganti *`value1`*, *`us-east-1`*, dan *`example-account-id`* dengan nilai Anda sendiri.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetStorageLensConfiguration",
                "s3:DeleteStorageLensConfiguration",
                "s3:PutStorageLensConfiguration"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/key1": "value1"
                }
            },
            "Resource": "arn:aws:s3:us-east-1:example-account-id:storage-lens/*"
        }
    ]
}
```

Melihat metrik dengan Lensa Penyimpanan Amazon S3

Lensa Penyimpanan S3 menggabungkan metrik Anda dan menampilkan informasi di bagian Snapshot akun di halaman Bucket konsol Amazon S3. Lensa Penyimpanan S3 juga menyediakan dasbor interaktif yang dapat Anda gunakan untuk memvisualisasikan wawasan dan tren, menandai outlier, serta menerima rekomendasi untuk mengoptimalkan biaya penyimpanan dan menerapkan praktik terbaik perlindungan data. Dasbor Anda memiliki opsi perincian untuk menghasilkan dan

memvisualisasikan wawasan di tingkat organisasi, akun, Wilayah AWS, kelas penyimpanan, bucket, prefiks, atau grup Lensa Penyimpanan. Anda juga dapat mengirimkan ekspor metrik harian dalam format CSV atau Parquet ke bucket S3.

Secara default, semua dasbor dikonfigurasi dengan metrik gratis, yang mencakup metrik yang dapat digunakan untuk memahami penggunaan dan aktivitas di seluruh penyimpanan S3, mengoptimalkan biaya penyimpanan, dan menerapkan praktik terbaik perlindungan data dan manajemen akses. Metrik gratis akan digabungkan ke tingkat bucket. Dengan metrik gratis, data tersedia untuk kueri hingga 14 hari.

Metrik dan rekomendasi lanjutan mencakup fitur tambahan berikut yang dapat digunakan untuk mendapatkan wawasan lebih lanjut tentang penggunaan dan aktivitas di seluruh penyimpanan dan praktik terbaik untuk mengoptimalkan penyimpanan:

- Rekomendasi kontekstual (hanya tersedia di dasbor)
- Metrik lanjutan (termasuk metrik aktivitas yang digabungkan berdasarkan bucket)
- Agregasi awalan
- Agregasi grup Lensa Penyimpanan
- Agregasi grup Lensa Penyimpanan
- CloudWatch Penerbitan Amazon

Data metrik lanjutan tersedia untuk kueri selama 15 bulan. Terdapat biaya tambahan untuk menggunakan Lensa Penyimpanan S3 dengan metrik lanjutan. Untuk informasi selengkapnya, lihat Harga Amazon S3. Untuk informasi selengkapnya tentang metrik gratis dan metrik lanjutan, lihat Pilihan metrik.

Topik

- Melihat metrik Lensa Penyimpanan S3 di dasbor
- Melihat metrik Lensa Penyimpanan Amazon S3 menggunakan ekspor data
- Pantau metrik Lensa Penyimpanan S3 di CloudWatch
- Kasus penggunaan metrik Lensa Penyimpanan Amazon S3

Melihat metrik Lensa Penyimpanan S3 di dasbor

Di konsol Amazon S3, Lensa Penyimpanan S3 menyediakan dasbor default interaktif yang dapat digunakan untuk memvisualisasikan wawasan dan tren dalam data. Anda juga dapat menggunakan

dasbor ini untuk menandai outlier dan menerima rekomendasi untuk mengoptimalkan biaya penyimpanan dan menerapkan praktik terbaik untuk perlindungan data. Dasbor memiliki opsi penelusuran untuk menghasilkan wawasan di tingkat grup akun, bucket, Wilayah AWS, awalan, atau Lensa Penyimpanan. Jika Anda telah mengaktifkan S3 Storage Lens untuk bekerja dengan AWS Organizations, Anda juga dapat menghasilkan wawasan di tingkat organisasi (seperti data untuk semua akun yang merupakan bagian dari AWS Organizations hierarki Anda). Dasbor selalu dimuat untuk tanggal terbaru yang metriknya tersedia.

Dasbor default S3 Storage Lens pada konsol diberi nama default-account-dashboard. Amazon S3 mengonfigurasi dasbor ini sebelumnya untuk memvisualisasikan ringkasan wawasan dan tren untuk keseluruhan akun dan memperbaruinya setiap hari di konsol S3. Anda tidak dapat mengubah cakupan konfigurasi dari dasbor default, tapi Anda dapat meningkatkan pilihan metrik dari metrik gratis ke metrik dan rekomendasi lanjutan berbayar. Dengan metrik dan rekomendasi lanjutan, Anda dapat mengakses metrik dan fitur tambahan. Fitur-fitur ini mencakup kategori metrik lanjutan, agregasi tingkat awalan, rekomendasi kontekstual, dan penerbitan Amazon. CloudWatch

Anda dapat menonaktifkan dasbor default, tetapi tidak dapat menghapusnya. Jika Anda menonaktifkan dasbor default, dasbor itu tidak akan diperbarui lagi. Anda juga tidak akan lagi menerima metrik harian baru di S3 Lensa Penyimpanan atau di bagian Snapshot akun di laman Bucket. Anda masih dapat melihat data historis di dasbor default hingga periode 14 hari untuk kueri data kedaluwarsa. Periodenya adalah 15 bulan jika Anda telah mengaktifkan metrik dan rekomendasi lanjutan. Untuk mengakses data ini, Anda dapat mengaktifkan kembali dasbor default di periode kedaluwarsa.

Anda dapat membuat dasbor Lensa Penyimpanan S3 tambahan dan mencakupnya berdasarkan Wilayah AWS, bucket S3, atau akun. Anda juga dapat membuat cakupan dasbor berdasarkan organisasi jika Anda telah mengaktifkan Storage Lens untuk bekerja dengannya AWS Organizations. Saat membuat atau mengedit dasbor Lensa Penyimpanan S3, tentukan cakupan dasbor dan pemilihan metrik.

Anda dapat menonaktifkan atau menghapus dasbor tambahan apa pun yang dibuat.

- Jika dasbor dinonaktifkan, dasbor tidak lagi diperbarui, dan Anda tidak akan lagi menerima metrik harian baru. Anda masih dapat melihat data historis untuk metrik gratis hingga periode kedaluwarsa 14 hari. Jika metrik dan rekomendasi lanjutan diaktifkan untuk dasbor tersebut, periodenya adalah 15 bulan. Anda dapat mengaktifkan kembali dasbor di dalam periode kedaluwarsa untuk mengakses data ini.

- Jika Anda menghapus dasbor, Anda akan kehilangan semua pengaturan konfigurasi dasbor. Anda tidak akan lagi menerima metrik harian baru, dan Anda juga akan kehilangan akses ke data historis yang terkait dengan dasbor tersebut. Jika Anda ingin mengakses data historis untuk dasbor yang telah dihapus, Anda harus membuat dasbor lain dengan nama yang sama di Wilayah asal yang sama.

Topik

- Melihat dasbor Lensa Penyimpanan Amazon S3
- Memahami dasbor Lensa Penyimpanan S3

Melihat dasbor Lensa Penyimpanan Amazon S3

Prosedur berikut ini menunjukkan cara melihat dasbor Lensa Penyimpanan S3 di konsol S3. Untuk panduan berbasis kasus penggunaan yang menunjukkan cara menggunakan dasbor untuk mengoptimalkan biaya, menerapkan praktik terbaik, dan meningkatkan kinerja aplikasi yang mengakses bucket S3, lihat Kasus penggunaan metrik Lensa Penyimpanan Amazon S3.

> **Note**
>
> Anda tidak dapat menggunakan akun kredensial pengguna root untuk melihat dasbor Lensa Penyimpanan Amazon S3. Untuk mengakses dasbor Lensa Penyimpanan S3, Anda harus memberikan izin yang diperlukan AWS Identity and Access Management (IAM) kepada pengguna IAM baru atau yang sudah ada. Lalu, masuk dengan kredensial pengguna tersebut untuk mengakses dasbor Lensa Penyimpanan S3. Untuk informasi selengkapnya tentang Menyetel izin Lensa Penyimpanan Amazon S3 dan Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, pilih Lensa Penyimpanan, Dasbor.
3. Di daftar Dasbor, pilih dasbor yang ingin Anda lihat.

   Dasbor akan terbuka di Lensa Penyimpanan S3. Bagian Snapshot untuk tanggal menunjukkan tanggal terakhir saat S3 Lensa Penyimpanan mengumpulkan metriknya. Dasbor akan selalu memuat tanggal terbaru yang metriknya tersedia.

4. (Opsional) Untuk mengubah tanggal dasbor Lensa Penyimpanan S3, di pemilih tanggal di bagian kanan atas, pilih tanggal baru.
5. (Opsional) Untuk menerapkan filter sementara untuk lebih membatasi cakupan data dasbor, lakukan hal berikut:
   a. Perluas bagian Filter.
   b. Untuk memfilter berdasarkan akun tertentu Wilayah AWS, kelas penyimpanan, bucket, awalan, atau grup Lensa Penyimpanan, pilih opsi yang akan difilter.

      > **Note**
      >
      > Filter Awalan dan filter grup Lensa Penyimpanan tidak dapat diterapkan secara bersamaan.

   c. Untuk memperbarui filter, pilih Terapkan.
   d. Untuk menghapus filter, klik X di sebelah filter.
6. Di bagian mana pun di dasbor Lensa Penyimpanan S3, untuk melihat data untuk metrik tertentu, untuk Metrik, pilih nama metrik.
7. Dalam bagan atau visualisasi apa pun di dasbor Lensa Penyimpanan S3, Anda dapat menelusuri tingkat agregasi yang lebih dalam dengan menggunakan tab Akun, Wilayah AWS, Kelas Penyimpanan, Bucket, Prefiks, atau Grup Lensa Penyimpanan. Sebagai contoh, lihat Temukan bucket Amazon S3 dingin.

Memahami dasbor Lensa Penyimpanan S3

Dasbor Lensa Penyimpanan S3 memiliki tab Ikhtisar utama, dan hingga lima tab tambahan yang mewakili setiap tingkat agregasi:

- Akun
- Wilayah AWS
- Kelas penyimpanan
- Bucket
- Awalan
- Grup Lensa Penyimpanan

Di tab Ikhtisar, data dasbor akan digabungkan menjadi tiga bagian berbeda: Snapshot untuk tanggal, Tren dan distribusi, dan Ikhtisar N teratas.

Untuk informasi selengkapnya tentang dasbor Lensa Penyimpanan S3, lihat bagian berikut.

Snapshot

Bagian Snapshot untuk tanggal menampilkan metrik ringkasan yang telah digabungkan oleh Lensa Penyimpanan S3 untuk tanggal yang dipilih. Metrik ringkasan ini mencakup metrik berikut:

- Total penyimpanan–Jumlah total penyimpanan yang digunakan dalam byte.
- Jumlah objek–Jumlah total objek di dalam Akun AWS.
- Ukuran objek rata-rata–Ukuran objek rata-rata.
- Bucket aktif–Jumlah total bucket aktif di dalam penggunaan aktif dengan penyimpanan > 0 byte di akun.
- Akun–Jumlah akun yang penyimpanannya di dalam cakupan. Nilai ini adalah 1 kecuali Anda menggunakan AWS Organizations dan Lensa Penyimpanan S3 Anda memiliki akses tepercaya dengan peran terkait layanan yang valid. Untuk informasi selengkapnya, lihat Menggunakan Peran Terkait Layanan untuk Lensa Penyimpanan Amazon S3.
- Bucket–Jumlah total bucket di akun.

Data metrik

Untuk setiap metrik yang muncul di snapshot, Anda dapat melihat data berikut:

- Nama metrik–Nama metriknya.
- Kategori metrik–Kategori tempat metrik dikelola.
- Total untuk tanggal–Perhitungan total untuk tanggal yang dipilih.
- % perubahan–Persentase perubahan dari tanggal snapshot terakhir.
- Tren 30 hari–Garis tren yang menunjukkan perubahan metrik selama periode 30 hari.
- Rekomendasi–Rekomendasi kontekstual berdasarkan data yang disediakan di dalam snapshot. Rekomendasi tersedia dengan metrik dan rekomendasi lanjutan. Untuk informasi selengkapnya, lihat Rekomendasi.

Kategori metrik

Anda dapat secara opsional memperbarui dasbor Snapshot untuk tanggal guna menampilkan metrik untuk kategori lain. Jika ingin melihat data snapshot untuk metrik tambahan, Anda dapat memilih dari Kategori metrik berikut:

- Optimasi biaya
- Perlindungan data
- Aktivitas (tersedia dengan metrik lanjutan)
- Manajemen akses
- Kinerja
- Peristiwa

Bagian Snapshot untuk tanggal hanya menampilkan pilihan metrik untuk setiap kategori saja. Untuk melihat semua metrik untuk kategori tertentu, pilih metrik di bagian Tren dan distribusi atau Ikhtisar N teratas. Untuk informasi selengkapnya tentang kategori metrik, lihat Kategori metrik. Untuk daftar lengkap metrik Lensa Penyimpanan S3, lihat Glosarium metrik Lensa Penyimpanan Amazon S3.

Tren dan distribusi

Bagian kedua dari tab Ikhtisar adalah Tren dan distribusi. Di bagian Tren dan distribusi, Anda dapat membandingkan dua metrik untuk rentang tanggal yang ditentukan. Bagian Tren dan distribusi menunjukkan hubungan antara dua metrik dari waktu ke waktu. Bagian ini menampilkan bagan yang dapat digunakan untuk melihat Kelas penyimpanan dan distribusi Wilayah antara dua tren yang dilacak. Anda dapat secara opsional menelusuri titik data di salah satu grafik untuk analisis yang lebih mendalam.

Untuk panduan yang menggunakan bagian Tren dan distribusi, lihat Identifikasi bucket yang tidak menggunakan enkripsi sisi server AWS KMS untuk enkripsi default (SSE-KMS).

Ikhtisar N teratas

Bagian ketiga dari dasbor Lensa Penyimpanan S3 adalah Ikhtisar N teratas (diurutkan ke dalam urutan menaik atau menurun). Bagian ini menampilkan metrik yang Anda pilih di sejumlah besar akun, bucket Wilayah AWS, awalan, atau grup Lensa Penyimpanan. Jika Anda mengaktifkan S3 Storage Lens untuk bekerja dengan AWS Organizations, Anda juga dapat melihat metrik yang dipilih di seluruh organisasi Anda.

Untuk panduan yang menggunakan bagian ikhtisar N teratas, lihat Identifikasi bucket S3 terbesar Anda.

Telusuri dan analisis berdasarkan opsi

Untuk memberikan pengalaman analisis yang lancar, dasbor Lensa Penyimpanan S3 menyediakan menu tindakan, yang muncul saat Anda memilih nilai bagan mana pun. Untuk menggunakan menu ini, pilih nilai bagan mana pun untuk melihat nilai metrik terkait, lalu pilih nilai bagan dari dua opsi di kotak yang muncul:

- Tindakan Penelusuran menerapkan nilai yang dipilih sebagai filter di semua tab dasbor. Anda kemudian dapat menelusuri ke dalam nilai tersebut untuk melakukan analisis mendalam.
- Tindakan Analisa berdasarkan akan menampilkan tab Dimensi yang dipilih dan menerapkan nilai tab tersebut sebagai filter. Tab ini mencakup Akun, Wilayah AWS, kelas Penyimpanan, Bucket, Awalan (untuk dasbor yang mengaktifkan Metrik lanjutan dan Agregasi awalan), dan grup Lensa Penyimpanan (untuk dasbor yang mengaktifkan Metrik lanjutan dan Agregasi grup Lensa Penyimpanan). Dengan Analisa berdasarkan, Anda dapat melihat data dengan konteks dimensi baru untuk analisis lebih mendalam.

Tindakan Penelusuran dan Analisis berdasarkanmungkin akan dinonaktifkan jika hasilnya memberikan hasil yang tidak logis atau tidak memiliki nilai apa pun. Tindakan Penelusuran dan Analisis berdasarkan menerapkan filter di atas setiap filter yang ada di semua tab dasbor. Anda juga dapat menghapus filter sesuai kebutuhan.

Tab

Tab tingkat dimensi memberikan tampilan terperinci untuk semua nilai di dalam dimensi tertentu. Misalnya, tab Wilayah AWS menampilkan metrik untuk semua Wilayah AWS, dan tab Bucket menampilkan metrik untuk semua bucket. Setiap tab dimensi berisi tata letak identik yang terdiri dari empat bagian:

- Bagan tren yang menampilkan item N teratas di dalam dimensi selama 30 hari terakhir untuk metrik yang dipilih. Secara default, bagan ini menampilkan 10 item teratas, tetapi Anda dapat menguranginya menjadi setidaknya 3 item atau menambahkannya hingga 50 item.
- Bagan histogram yang menampilkan diagram batang vertikal untuk tanggal dan metrik yang dipilih. Jika Anda memiliki banyak item untuk ditampilkan di bagan ini, Anda mungkin perlu menggulir secara horizontal.
- Bagan analisis gelembung yang memplot semua item di dalam dimensi. Bagan ini mewakili metrik pertama di sumbu x dan metrik kedua di sumbu y. Metrik ketiga akan diwakili oleh ukuran gelembung.

- Tampilan kisi metrik yang berisi setiap item di dalam dimensi yang tercantum di dalam baris. Kolom mewakili setiap metrik yang tersedia, disusun di dalam tab kategori metrik untuk navigasi yang lebih mudah.

Melihat metrik Lensa Penyimpanan Amazon S3 menggunakan ekspor data

Metrik Lensa Penyimpanan Amazon S3 dihasilkan setiap hari dalam file ekspor metrik berformat CSV atau Apache Parquet dan ditempatkan dalam bucket S3 di akun Anda. Dari sana, Anda dapat memasukkan ekspor metrik ke alat analisis pilihan Anda, seperti Amazon QuickSight dan Amazon Athena, tempat Anda dapat menganalisis penggunaan penyimpanan dan tren aktivitas.

Topik

- Menggunakan AWS KMS key untuk mengenkripsi ekspor metrik Anda
- Apa yang dimaksud dengan manifes ekspor Lensa Penyimpanan S3?
- Memahami skema ekspor Lensa Penyimpanan Amazon S3

Menggunakan AWS KMS key untuk mengenkripsi ekspor metrik Anda

Untuk memberikan izin kepada Lensa Penyimpanan Amazon S3 guna mengenkripsi ekspor metrik menggunakan CMK, Anda harus menggunakan kebijakan kunci. Untuk memperbarui kebijakan kunci sehingga Anda dapat menggunakan kunci KMS untuk mengenkripsi ekspor metrik Lensa Penyimpanan S3 Anda, ikuti langkah-langkah berikut.

Untuk memberikan izin Lensa Penyimpanan S3 guna mengenkripsi data dengan menggunakan kunci KMS

1. Masuk ke AWS Management Console dengan menggunakan Akun AWS yang memiliki kunci yang dikelola pelanggan.
2. Buka AWS KMS konsol di https://console.aws.amazon.com/kms.
3. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
4. Di panel navigasi kiri, pilih CMK.
5. Di bawah Kunci terkelola pelanggan, pilih kunci yang ingin Anda gunakan untuk mengenkripsi ekspor metrik. AWS KMS keys khusus Wilayah dan harus berada di Wilayah yang sama dengan bucket S3 tujuan ekspor metrik.
6. Di Bawah Kebijakan kunci, pilih Beralih ke tampilan kebijakan.

7. Untuk memperbarui kebijakan kunci, pilih Edit.
8. Di Bawah Edit kebijakan kunci, tambahkan kebijakan kunci berikut ke kebijakan kunci yang ada. Untuk menggunakan kebijakan ini, ganti *user input placeholders* dengan informasi Anda.

```
{
    "Sid": "Allow Amazon S3 Storage Lens use of the KMS key",
     "Effect": "Allow",
    "Principal": {
        "Service": "storage-lens.s3.amazonaws.com"
    },
    "Action": [
        "kms:GenerateDataKey"
    ],
    "Resource": "*",
    "Condition": {
       "StringEquals": {
           "aws:SourceArn": "arn:aws:s3:us-east-1:source-account-id:storage-
lens/your-dashboard-name",
           "aws:SourceAccount": "source-account-id"
        }
     }
}
```

9. Pilih Simpan perubahan.

Untuk informasi selengkapnya tentang cara membuat CMK, dan menggunakan kebijakan kunci, lihat topik berikut di Panduan Developer AWS Key Management Service :

- Memulai
- Menggunakan kebijakan utama di AWS KMS

Anda juga dapat menggunakan AWS KMS PUT key policy API operation (PutKeyPolicy) untuk menyalin kebijakan kunci ke kunci terkelola pelanggan yang ingin Anda gunakan untuk mengenkripsi ekspor metrik menggunakan REST API,, AWS CLI dan. SDKs

Apa yang dimaksud dengan manifes ekspor Lensa Penyimpanan S3?

Mengingat banyaknya jumlah data yang dikumpulkan, ekspor metrik harian Lensa Penyimpanan S3 dapat dibagi menjadi beberapa file. File manifes `manifest.json` menjelaskan lokasi file ekspor metrik untuk hari tersebut. Setiap kali ada ekspor baru yang dikirimkan, ekspor tersebut disertai

dengan manifes baru. Tiap manifes yang terkandung dalam file `manifest.json` menyediakan metadata dan informasi dasar lainnya tentang ekspor.

Informasi manifes mencakup properti berikut:

- `sourceAccountId`–ID akun dari pemilik konfigurasi.
- `configId`–Pengidentifikasi unik untuk dasbor.
- `destinationBucket`–Bucket tujuan Amazon Resource Name (ARN) tempat metrik ekspor ditempatkan.
- `reportVersion`–Versi dari ekspor.
- `reportDate`–Tanggal dari laporan.
- `reportFormat`–Format dari laporan.
- `reportSchema`–Skema dari laporan.
- `reportFiles`–Daftar aktual dari file laporan ekspor yang berada di bucket tujuan.

Berikut ini adalah contoh manifes dalam file `manifest.json` untuk ekspor berformat CSV.

```
{
   "sourceAccountId":"123456789012",
   "configId":"my-dashboard-configuration-id",
   "destinationBucket":"arn:aws:s3:::amzn-s3-demo-destination-bucket",
   "reportVersion":"V_1",
   "reportDate":"2020-11-03",
   "reportFormat":"CSV",

 "reportSchema":"version_number,configuration_id,report_date,aws_account_number,aws_region,stor
   "reportFiles":[
        {
            "key":"DestinationPrefix/StorageLens/123456789012/my-dashboard-
configuration-id/V_1/reports/dt=2020-11-03/a38f6bc4-2e3d-4355-ac8a-e2fdcf3de158.csv",
            "size":1603959,
            "md5Checksum":"2177e775870def72b8d84febe1ad3574"
        }
      ]
}
```

Berikut ini adalah contoh manifes dalam file `manifest.json` untuk ekspor berformat Parquet.

```
{
   "sourceAccountId":"123456789012",
   "configId":"my-dashboard-configuration-id",
   "destinationBucket":"arn:aws:s3:::amzn-s3-demo-destination-bucket",
   "reportVersion":"V_1",
   "reportDate":"2020-11-03",
   "reportFormat":"Parquet",
   "reportSchema":"message s3.storage.lens { required string version_number;
 required string configuration_id; required string report_date; required string
 aws_account_number; required string aws_region; required string storage_class;
 required string record_type; required string record_value; required string
 bucket_name; required string metric_name; required long metric_value; }",
   "reportFiles":[
      {
         "key":"DestinationPrefix/StorageLens/123456789012/my-dashboard-configuration-
id/V_1/reports/dt=2020-11-03/bd23de7c-b46a-4cf4-bcc5-b21aac5be0f5.par",
         "size":14714,
         "md5Checksum":"b5c741ee0251cd99b90b3e8eff50b944"
      }
}
```

Anda dapat mengonfigurasi ekspor metrik agar dihasilkan sebagai bagian dari konfigurasi dasbor di konsol Amazon S3 atau dengan menggunakan Amazon S3 REST API,, dan. AWS CLI SDKs

Memahami skema ekspor Lensa Penyimpanan Amazon S3

Tabel berikut berisi skema ekspor metrik Lensa Penyimpanan S3 Anda.

Nama atribut	Tipe data	Nama kolom	Deskripsi
`VersionNumber`	String	`version_number`	Versi metrik Lensa Penyimpanan S3 yang digunakan.
`ConfigurationId`	String	`configuration_id`	`configuration_id` dari konfigurasi Lensa Penyimpanan S3 Anda.

Nama atribut	Tipe data	Nama kolom	Deskripsi
`ReportDate`	String	`report_date`	Tanggal metrik dilacak.
`AwsAccountNumber`	String	`aws_account_number`	Akun AWS Nomor Anda.
`AwsRegion`	String	`aws_region`	Metrik yang dilacak. Wilayah AWS
`StorageClass`	String	`storage_class`	Kelas penyimpanan bucket yang dimaksud.
`RecordType`	ENUM	`record_type`	Jenis artefak yang sedang dilaporkan (ACCOUNT, BUCKET, atau PREFIX).
`RecordValue`	String	`record_value`	Nilai artefak `RecordType` . **Note** `record_value` dienkodekan dengan URL.
`BucketName`	String	`bucket_name`	Nama bucket yang sedang dilaporkan.
`MetricName`	String	`metric_name`	Nama metrik yang sedang dilaporkan.

Nama atribut	Tipe data	Nama kolom	Deskripsi
`MetricValue`	Long	`metric_value`	Nilai metrik yang sedang dilaporkan.

Contoh ekspor metrik Lensa Penyimpanan S3

Berikut ini adalah contoh ekspor metrik Lensa Penyimpanan S3 berdasarkan skema ini.

> **Note**
>
> Anda dapat mengidentifikasi metrik untuk Grup Lensa Penyimpanan dengan mencari nilai `STORAGE_LENS_GROUP_BUCKET` atau `STORAGE_LENS_GROUP_ACCOUNT` di kolom `record_type`. Kolom `record_value` akan menampilkan Amazon Resource Name (ARN) untuk Grup Lensa Penyimpanan, misalnya, `arn:aws:s3:us-east-1:123456789012:storage-lens-group/slg-1`.

version_r	configuration_id	report_date	aws_account_number	aws_region	storage_class	record_type	record_value	bucket_name	metric_name	metric_value
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			StorageBytes	2478830621
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			ObjectCount	1598962
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			ReplicatedStorageBytes	20000
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			ReplicatedObjectCount	20
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			EncryptedStorageBytes	2478828742
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			EncryptedObjectCount	1598961
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			DeleteMarkerObjectCount	1500
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			ObjectLockEnabledStorageBytes	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			ObjectLockEnabledObjectCount	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			CurrentVersionStorageBytes	2478830621
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			CurrentVersionObjectCount	1598962
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			NonCurrentVersionStorageBytes	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			NonCurrentVersionObjectCount	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			IncompleteMultipartUploadStorageBytes	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			IncompleteMultipartUploadObjectCount	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	StorageBytes	29996800
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	ObjectCount	12264
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	ReplicatedStorageBytes	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	ReplicatedObjectCount	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	EncryptedStorageBytes	29996800
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	EncryptedObjectCount	12264
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	DeleteMarkerObjectCount	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	ObjectLockEnabledStorageBytes	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	ObjectLockEnabledObjectCount	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	CurrentVersionStorageBytes	29996800
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	CurrentVersionObjectCount	12264
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	NonCurrentVersionStorageBytes	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	NonCurrentVersionObjectCount	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	IncompleteMultipartUploadStorageBytes	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	IncompleteMultipartUploadObjectCount	0

Berikut ini adalah contoh ekspor metrik Lensa Penyimpanan S3 dengan data Grup Lensa Penyimpanan.

version_numbe	configuration_id	report_date	aws_account_num	aws_region	storage_class	record_type	record_value	bucket_name	metric_name	metric_value
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		StorageBytes	3128548856
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ObjectCount	4440
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ReplicatedStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ReplicatedObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		EncryptedStorageBytes	3128548856
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		EncryptedObjectCount	4440
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		DeleteMarkerObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ObjectLockEnabledStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ObjectLockEnabledObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		CurrentVersionStorageBytes	3128548856
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		CurrentVersionObjectCount	4440
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		NonCurrentVersionStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		NonCurrentVersionObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		IncompleteMultipartUploadStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		IncompleteMultipartUploadObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		DeleteMarkerStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ReplicatedStorageBytesSource	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ReplicatedObjectCountSource	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		IncompleteMPUStorageBytesOlderThan7Days	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		IncompleteMPUObjectCountOlderThan7Days	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	StorageBytes	676863200
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ObjectCount	3000
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ReplicatedStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ReplicatedObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	EncryptedStorageBytes	676863200
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	EncryptedObjectCount	3000
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	DeleteMarkerObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ObjectLockEnabledStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ObjectLockEnabledObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	CurrentVersionStorageBytes	676863200
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	CurrentVersionObjectCount	3000
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	NonCurrentVersionStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	NonCurrentVersionObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	IncompleteMultipartUploadStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	IncompleteMultipartUploadObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	DeleteMarkerStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ReplicatedStorageBytesSource	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ReplicatedObjectCountSource	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	IncompleteMPUStorageBytesOlderThan7Days	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	IncompleteMPUObjectCountOlderThan7Days	0

Pantau metrik Lensa Penyimpanan S3 di CloudWatch

Anda dapat menerbitkan metrik Lensa Penyimpanan S3 CloudWatch ke Amazon untuk membuat tampilan terpadu kesehatan operasional Anda di dasbor. CloudWatch Anda juga dapat menggunakan CloudWatch fitur, seperti alarm dan tindakan yang dipicu, matematika metrik, dan deteksi anomali, untuk memantau dan mengambil tindakan pada metrik Lensa Penyimpanan S3. Selain itu, operasi CloudWatch API memungkinkan aplikasi, termasuk penyedia pihak ketiga, untuk mengakses metrik Lensa Penyimpanan S3 Anda. Untuk informasi selengkapnya tentang CloudWatch fitur, lihat Panduan CloudWatch Pengguna Amazon.

Anda dapat mengaktifkan opsi CloudWatch penerbitan untuk konfigurasi dasbor baru atau yang sudah ada dengan menggunakan konsol Amazon S3, Amazon S3 REST API, dan. AWS CLI AWS SDKs Dasbor yang ditingkatkan ke metrik dan rekomendasi lanjutan S3 Storage Lens dapat menggunakan opsi penerbitan. CloudWatch Untuk metrik dan rekomendasi lanjutan, lihat Harga Lensa Penyimpanan Amazon S3. Tidak ada biaya penerbitan CloudWatch metrik tambahan yang berlaku; namun, CloudWatch biaya lain, seperti dasbor, alarm, dan panggilan API, berlaku. Untuk informasi lebih lanjut, lihat harga Amazon CloudWatch.

Metrik Lensa Penyimpanan S3 dipublikasikan ke akun yang memiliki konfigurasi CloudWatch Lensa Penyimpanan S3. Setelah mengaktifkan opsi CloudWatch penerbitan dalam metrik dan rekomendasi lanjutan, Anda dapat mengakses metrik tingkat organisasi, akun, dan ember. CloudWatch Metrik tingkat awalan tidak tersedia di. CloudWatch

> **Note**
>
> Metrik Lensa Penyimpanan S3 adalah metrik harian dan dipublikasikan CloudWatch satu kali per hari. Saat Anda menanyakan metrik Lensa Penyimpanan S3 CloudWatch, periode kueri harus 1 hari (86400 detik). Setelah metrik Lensa Penyimpanan S3 harian Anda muncul di dasbor Lensa Penyimpanan S3 Anda di konsol Amazon S3, diperlukan beberapa jam agar metrik yang sama ini muncul. CloudWatch Saat Anda mengaktifkan opsi CloudWatch penerbitan untuk metrik Lensa Penyimpanan S3 untuk pertama kalinya, metrik Anda dapat dipublikasikan hingga 24 jam. CloudWatch

Setelah mengaktifkan opsi CloudWatch penerbitan, Anda dapat menggunakan CloudWatch fitur berikut untuk memantau dan menganalisis data LensStorage Lensa Penyimpanan S3 Anda:

- Dasbor - Gunakan CloudWatch dasbor untuk membuat dasbor Lensa Penyimpanan S3 yang disesuaikan. Bagikan CloudWatch dasbor Anda dengan orang-orang yang tidak memiliki akses langsung ke Anda Akun AWS, lintas tim, dengan pemangku kepentingan, dan dengan orang-orang di luar organisasi Anda.
- Alarm dan tindakan yang dipicu–Konfigurasikan alarm yang memantau metrik dan mengambil tindakan ketika ambang batas dilanggar. Misalnya, Anda dapat mengonfigurasi alarm yang mengirimkan notifikasi Amazon SNS jika metrik Byte Unggahan Multibagian Tidak Lengkap melebihi 1 GB selama tiga hari berturut-turut.
- Deteksi anomali–Aktifkan deteksi anomali untuk terus menganalisis metrik, menentukan garis dasar normal, dan anomali permukaan. Anda dapat membuat alarm deteksi anomali berdasarkan nilai yang diharapkan dari suatu metrik. Misalnya, Anda dapat memantau anomali untuk metrik Byte yang Mengaktifkan Kunci Objek untuk mendeteksi penghapusan pengaturan Kunci Objek yang tidak sah.
- Matematika metrik–Anda juga dapat menggunakan matematika metrik untuk membuat kueri beberapa metrik Lensa Penyimpanan S3 dan menggunakan ekspresi matematika untuk membuat deret waktu baru berdasarkan metrik ini. Misalnya, Anda dapat membuat metrik baru untuk mendapatkan ukuran objek rata-rata dengan membagi `StorageBytes` dengan `ObjectCount`.

Untuk informasi selengkapnya tentang opsi CloudWatch penerbitan untuk metrik Lensa Penyimpanan S3, lihat topik berikut.

Topik

- Metrik dan dimensi Lensa Penyimpanan S3
- Mengaktifkan CloudWatch penerbitan untuk S3 Storage Lens
- Bekerja dengan metrik Lensa Penyimpanan S3 di CloudWatch

Metrik dan dimensi Lensa Penyimpanan S3

Untuk mengirim metrik Lensa Penyimpanan S3 CloudWatch, Anda harus mengaktifkan opsi CloudWatch penerbitan dalam metrik dan rekomendasi lanjutan S3 Storage Lens. Setelah metrik lanjutan diaktifkan, Anda dapat menggunakan CloudWatchdasbor untuk memantau metrik Lensa Penyimpanan S3 bersama metrik aplikasi lainnya dan membuat tampilan terpadu tentang kesehatan operasional Anda. Anda dapat menggunakan dimensi untuk memfilter metrik Lensa Penyimpanan S3 CloudWatch menurut organisasi, akun, bucket, kelas penyimpanan, Wilayah, dan ID konfigurasi metrik.

Metrik Lensa Penyimpanan S3 dipublikasikan ke akun yang memiliki konfigurasi CloudWatch Lensa Penyimpanan S3. Setelah mengaktifkan opsi CloudWatch penerbitan dalam metrik dan rekomendasi lanjutan, Anda dapat mengakses metrik tingkat organisasi, akun, dan ember. CloudWatch Metrik tingkat awalan tidak tersedia di. CloudWatch

> **Note**
>
> Metrik Lensa Penyimpanan S3 adalah metrik harian dan dipublikasikan CloudWatch satu kali per hari. Saat Anda menanyakan metrik Lensa Penyimpanan S3 CloudWatch, periode kueri harus 1 hari (86400 detik). Setelah metrik Lensa Penyimpanan S3 harian Anda muncul di dasbor Lensa Penyimpanan S3 Anda di konsol Amazon S3, diperlukan beberapa jam agar metrik yang sama ini muncul. CloudWatch Saat Anda mengaktifkan opsi CloudWatch penerbitan untuk metrik Lensa Penyimpanan S3 untuk pertama kalinya, metrik Anda dapat dipublikasikan hingga 24 jam. CloudWatch

Untuk informasi selengkapnya tentang metrik dan dimensi Lensa Penyimpanan S3 CloudWatch, lihat topik berikut.

Topik

- Metrik
- Dimensi

Metrik

Metrik Lensa Penyimpanan S3 tersedia sebagai metrik di dalamnya. CloudWatch Metrik Lensa Penyimpanan S3 dipublikasikan ke namespace `AWS/S3/Storage-Lens`. Namespace ini hanya untuk metrik Lensa Penyimpanan S3. Metrik bucket, permintaan, dan replikasi Amazon S3 akan dipublikasikan ke namespace `AWS/S3`.

Metrik Lensa Penyimpanan S3 dipublikasikan ke akun yang memiliki konfigurasi CloudWatch Lensa Penyimpanan S3. Setelah mengaktifkan opsi CloudWatch penerbitan dalam metrik dan rekomendasi lanjutan, Anda dapat mengakses metrik tingkat organisasi, akun, dan ember. CloudWatch Metrik tingkat awalan tidak tersedia di. CloudWatch

Di Lensa Penyimpanan S3, metrik akan digabungkan dan disimpan hanya di Wilayah asal yang sudah ditentukan. Metrik Lensa Penyimpanan S3 juga dipublikasikan CloudWatch di Wilayah beranda yang Anda tentukan dalam konfigurasi Lensa Penyimpanan S3.

Untuk daftar lengkap metrik Lensa Penyimpanan S3, termasuk daftar metrik yang tersedia, lihat. CloudWatch Glosarium metrik Lensa Penyimpanan Amazon S3

> **Note**
>
> Statistik valid untuk metrik Lensa Penyimpanan S3 CloudWatch adalah Rata-rata. Untuk informasi selengkapnya tentang statistik CloudWatch, lihat definisi CloudWatch statistik di Panduan CloudWatch Pengguna Amazon.

Granularitas metrik Lensa Penyimpanan S3 di CloudWatch

Lensa Penyimpanan S3 menawarkan metrik pada granularitas organisasi, akun, bucket, dan awalan. Lensa Penyimpanan S3 menerbitkan metrik Lensa Penyimpanan S3 tingkat organisasi, akun, dan ember ke. CloudWatch Metrik Lensa Penyimpanan S3 tingkat awalan tidak tersedia di. CloudWatch

Untuk informasi selengkapnya tentang perincian metrik Lensa Penyimpanan S3 yang tersedia di CloudWatch, lihat daftar berikut:

- Organisasi–Metrik yang digabungkan dari seluruh akun anggota di organisasi. S3 Storage Lens menerbitkan metrik untuk akun anggota CloudWatch di akun manajemen.
  - Organisasi dan akun–Metrik untuk akun anggota di organisasi.
  - Organisasi dan bucket–Metrik untuk bucket Amazon S3 di akun anggota organisasi.
- Akun (Tingkat non-organisasi)–Metrik yang dikumpulkan di seluruh bucket di akun.

- Bucket (Tingkat non-organisasi)–Metrik untuk bucket tertentu. Pada tahun CloudWatch, S3 Storage Lens menerbitkan metrik ini ke Akun AWS yang membuat konfigurasi Lensa Penyimpanan S3. Lensa Penyimpanan S3 menerbitkan metrik ini hanya untuk konfigurasi non-organisasi.

Dimensi

Saat Lensa Penyimpanan S3 mengirimkan data ke CloudWatch, dimensi dilampirkan ke setiap metrik. Dimensi adalah kategori yang menggambarkan karakteristik metrik. Anda dapat menggunakan dimensi untuk memfilter hasil yang CloudWatch kembali.

Misalnya, semua metrik Lensa Penyimpanan S3 CloudWatch memiliki dimensi. `configuration_id` Anda dapat menggunakan dimensi ini untuk membedakan antara metrik yang terkait dengan konfigurasi Lensa Penyimpanan S3 tertentu. `organization_id` mengidentifikasi metrik tingkat organisasi. Untuk informasi selengkapnya tentang dimensi CloudWatch, lihat Dimensi dalam Panduan CloudWatch Pengguna.

Dimensi yang berbeda tersedia untuk metrik Lensa Penyimpanan S3 tergantung pada granularitas metriknya. Misalnya, Anda dapat menggunakan `organization_id` dimensi untuk memfilter metrik tingkat organisasi berdasarkan ID. AWS Organizations Namun, Anda tidak dapat menggunakan dimensi ini untuk metrik bucket dan metrik tingkat akun. Untuk informasi selengkapnya, lihat Memfilter metrik dengan menggunakan dimensi.

Untuk melihat dimensi mana yang tersedia untuk konfigurasi Lensa Penyimpanan S3 Anda, lihat tabel berikut.

Dimensi	Deskripsi	Bucket	Organisasi	Organisasi dan bucket	Organisasi dan akun
`configuration_id`	Nama dasbor untuk konfigurasi Lensa Penyimpanan S3 yang dilaporkan di dalam metrik	.	.	.	.
`metrics_version`	Versi metrik Lensa Penyimpanan S3. Versi metrik memiliki nilai tetap `1.0`.	.	.	.	.
`organization_id`	AWS Organizations ID untuk metrik	.	.	.	.

Dimensi	Deskripsi	Bucket	Akun	Organisasi	Organisasi dan bucket	Organisasi dan akun
aws_account_number	Akun AWS Yang terkait dengan metrik	.	.	.	.	.
aws_region	Wilayah AWS Untuk metrik	.	.	.	.	.
bucket_name	Nama bucket S3 yang dilaporkan di dalam metrik	.	.	.	.	.
storage_class	Kelas penyimpanan untuk bucket yang dilaporkan di dalam metrik	.	.	.	.	.
record_type	Granularitas metrik: ORGANISASI, AKUN, BUCKET	BUCKET	AKUN	BUCKET	AKUN	ORGANISASI

Mengaktifkan CloudWatch penerbitan untuk S3 Storage Lens

Anda dapat menerbitkan metrik Lensa Penyimpanan S3 CloudWatch ke Amazon untuk membuat tampilan terpadu kesehatan operasional Anda di dasbor. CloudWatch Anda juga dapat menggunakan CloudWatch fitur, seperti alarm dan tindakan yang dipicu, matematika metrik, dan deteksi anomali, untuk memantau dan mengambil tindakan pada metrik Lensa Penyimpanan S3. Selain itu, operasi CloudWatch API memungkinkan aplikasi, termasuk penyedia pihak ketiga, untuk mengakses metrik Lensa Penyimpanan S3 Anda. Untuk informasi selengkapnya tentang CloudWatch fitur, lihat Panduan CloudWatch Pengguna Amazon.

Metrik Lensa Penyimpanan S3 dipublikasikan ke akun yang memiliki konfigurasi CloudWatch Lensa Penyimpanan S3. Setelah mengaktifkan opsi CloudWatch penerbitan dalam metrik dan rekomendasi lanjutan, Anda dapat mengakses metrik tingkat organisasi, akun, dan ember. CloudWatch Metrik tingkat awalan tidak tersedia di. CloudWatch

Anda dapat mengaktifkan CloudWatch dukungan untuk konfigurasi dasbor baru atau yang sudah ada dengan menggunakan konsol S3, Amazon S3 APIs REST AWS CLI, dan. AWS SDKs Opsi CloudWatch penerbitan tersedia untuk dasbor yang ditingkatkan ke metrik dan rekomendasi lanjutan S3 Storage Lens. Untuk metrik dan rekomendasi lanjutan, lihat Harga Lensa Penyimpanan Amazon S3. Tidak ada biaya penerbitan CloudWatch metrik tambahan yang berlaku; namun, CloudWatch biaya lain, seperti dasbor, alarm, dan panggilan API, berlaku.

Untuk mengaktifkan opsi CloudWatch penerbitan metrik Lensa Penyimpanan S3, lihat topik berikut.

> **Note**
>
> Metrik Lensa Penyimpanan S3 adalah metrik harian dan dipublikasikan CloudWatch satu kali per hari. Saat Anda menanyakan metrik Lensa Penyimpanan S3 CloudWatch, periode kueri harus 1 hari (86400 detik). Setelah metrik Lensa Penyimpanan S3 harian Anda muncul di dasbor Lensa Penyimpanan S3 Anda di konsol Amazon S3, diperlukan beberapa jam agar metrik yang sama ini muncul. CloudWatch Saat Anda mengaktifkan opsi CloudWatch penerbitan untuk metrik Lensa Penyimpanan S3 untuk pertama kalinya, metrik Anda dapat dipublikasikan hingga 24 jam. CloudWatch
> Saat ini, metrik Lensa Penyimpanan S3 tidak dapat dikonsumsi melalui CloudWatch aliran.

Menggunakan konsol S3

Saat memperbarui dasbor Lensa Penyimpanan S3, Anda tidak dapat mengubah nama dasbor atau Wilayah asal. Anda juga tidak dapat mengubah cakupan dasbor default, yang mencakup seluruh penyimpanan akun.

Untuk memperbarui dasbor Lensa Penyimpanan S3 untuk mengaktifkan penerbitan CloudWatch

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, pilih Lensa Penyimpanan S3, Dasbor.
3. Pilih dasbor yang ingin diedit, lalu pilih Edit.
4. Di bawah Pilihan metrik, pilih Metrik dan rekomendasi lanjutan.

   Metrik dan rekomendasi lanjutan tersedia dengan biaya tambahan. Metrik dan rekomendasi lanjutan mencakup periode 15 bulan untuk kueri data, metrik penggunaan yang dikumpulkan pada tingkat awalan, metrik aktivitas yang dikumpulkan berdasarkan bucket, opsi CloudWatch

penerbitan, dan rekomendasi kontekstual yang membantu Anda mengoptimalkan biaya penyimpanan dan menerapkan praktik terbaik perlindungan data. Untuk informasi selengkapnya, lihat Harga Amazon S3.

5. Di bawah Pilih Metrik lanjutan dan fitur rekomendasi, pilih CloudWatchpenerbitan.

> **Important**
>
> Jika konfigurasi Anda mengaktifkan agregasi awalan untuk metrik penggunaan, metrik tingkat awalan tidak akan dipublikasikan. CloudWatch Hanya metrik Lensa Penyimpanan S3 tingkat bucket, akun, dan organisasi yang dipublikasikan. CloudWatch

6. Pilih Simpan perubahan.

Untuk membuat dasbor Lensa Penyimpanan S3 baru yang memungkinkan dukungan CloudWatch

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, pilih Lensa Penyimpanan, Dasbor.
3. Pilih Buat dasbor.
4. Di bagian bawah Umum, tentukan opsi konfigurasi berikut:

   a. Untuk Nama dasbor, masukkan nama dasbor Anda.

      Nama dasbor harus kurang dari 65 karakter dan tidak boleh berisi karakter khusus atau spasi. Anda tidak dapat mengubah nama dasbor setelah dasbor dibuat.

   b. Pilih Wilayah Asal untuk dasbor Anda.

      Metrik untuk semua Wilayah yang disertakan di dalam cakupan dasbor ini disimpan secara terpusat di Wilayah asal yang ditunjuk. Di CloudWatch, metrik Lensa Penyimpanan S3 juga tersedia di Wilayah asal. Anda tidak dapat mengubah Wilayah asal setelah dasbor dibuat.

5. (Opsional) Untuk menambahkan tag, Tambahkan tag dan masukkan tag Kunci dan Nilai.

> **Note**
>
> Anda dapat menambahkan hingga 50 tag ke konfigurasi dasbor.

6. Tentukan cakupan untuk konfigurasi Anda:

a. Saat akan membuat konfigurasi tingkat organisasi, pilih akun yang akan disertakan dalam konfigurasi: Sertakan semua akun di dalam konfigurasi atau Batasi cakupan ke akun masuk.

> **Note**
>
> Saat membuat konfigurasi tingkat organisasi yang mencakup semua akun, Anda hanya dapat menyertakan atau mengecualikan Wilayah, bukan bucket.

b. Pilih Wilayah dan bucket yang Anda ingin Lensa Penyimpanan S3 sertakan di dalam konfigurasi dasbor dengan melakukan hal berikut:

- Untuk menyertakan semua Wilayah, pilih Sertakan Wilayah dan bucket.
- Untuk menyertakan Wilayah tertentu, hapus Sertakan semua Wilayah. Di bawah Pilih Wilayah untuk disertakan, pilih Wilayah yang Anda ingin Lensa Penyimpanan S3 sertakan di dasbor.
- Untuk menyertakan bucket tertentu, hapus Sertakan semua ember. Di bagian bawah Pilih bucket untuk disertakan, pilih bucket yang Anda ingin Lensa Penyimpanan S3 sertakan di dasbor.

> **Note**
>
> Anda dapat memilih hingga 50 bucket.

7. Untuk Pilihan metrik, pilih Metrik dan rekomendasi lanjutan.

   Untuk informasi selengkapnya tentang metrik dan rekomendasi lanjutan, lihat harga Amazon S3.

8. Di bagian bawah Fitur metrik dan rekomendasi lanjutan, pilih opsi yang ingin diaktifkan:

   - Metrik lanjutan
   - CloudWatch penerbitan

> **Important**
>
> Jika Anda mengaktifkan agregasi awalan untuk konfigurasi Lensa Penyimpanan S3 Anda, metrik tingkat awalan tidak akan dipublikasikan. CloudWatch Hanya metrik Lensa Penyimpanan S3 tingkat bucket, akun, dan organisasi yang dipublikasikan. CloudWatch

- Agregasi awalan

> **Note**
>
> Untuk informasi selengkapnya tentang fitur metrik dan rekomendasi lanjutan, lihat Pilihan metrik.

9. Jika Anda mengaktifkan Metrik lanjutan, pilih kategori metrik lanjutan yang ingin ditampilkan di dasbor Lensa Penyimpanan S3:

   - Metrik aktivitas
   - Metrik kode status terperinci
   - Metrik pengoptimalan biaya lanjutan
   - Metrik perlindungan data lanjutan

   Untuk informasi selengkapnya tentang kategori metrik, lihat Kategori metrik. Untuk daftar lengkap metrik, lihat Glosarium metrik Lensa Penyimpanan Amazon S3.

10. (Opsional) Konfigurasikan ekspor metrik Anda.

    Untuk informasi selengkapnya tentang cara mengonfigurasi ekspor metrik, lihat langkah Menggunakan konsol S3.

11. Pilih Buat dasbor.

Menggunakan AWS CLI

AWS CLI Contoh berikut memungkinkan opsi CloudWatch penerbitan dengan menggunakan metrik lanjutan dan konfigurasi rekomendasi tingkat organisasi S3 Storage Lens. Untuk menggunakan contoh ini, ganti *`user input placeholders`*dengan informasi Anda sendiri.

```
aws s3control put-storage-lens-configuration --account-id=555555555555 --config-
id=your-configuration-id --region=us-east-1 --storage-lens-configuration=file://./
config.json

config.json
{
  "Id": "SampleS3StorageLensConfiguration", //Use this property to identify your S3
 Storage Lens configuration.
```

```
  "AwsOrg": { //Use this property when enabling S3 Storage Lens for AWS Organizations.
    "Arn": "arn:aws:organizations::123456789012:organization/o-abcdefgh"
  },
  "AccountLevel": {
    "ActivityMetrics": {
      "IsEnabled":true
    },
    "AdvancedCostOptimizationMetrics": {
      "IsEnabled":true
    },
    "AdvancedDataProtectionMetrics": {
      "IsEnabled":true
    },
    "DetailedStatusCodesMetrics": {
      "IsEnabled":true
    },
    "BucketLevel": {
      "ActivityMetrics": {
        "IsEnabled":true //Mark this as false if you want only free metrics.
      },
      "ActivityMetrics": {
        "IsEnabled":true //Mark this as false if you want only free metrics.
      },
      "AdvancedCostOptimizationMetrics": {
        "IsEnabled":true //Mark this as false if you want only free metrics.
      },
      "DetailedStatusCodesMetrics": {
        "IsEnabled":true //Mark this as false if you want only free metrics.
      },
      "PrefixLevel":{
        "StorageMetrics":{
          "IsEnabled":true, //Mark this as false if you want only free metrics.
          "SelectionCriteria":{
            "MaxDepth":5,
            "MinStorageBytesPercentage":1.25,
            "Delimiter":"/"
          }
        }
      }
    }
  },
  "Exclude": { //Replace with "Include" if you prefer to include Regions.
    "Regions": [
      "eu-west-1"
```

```
    ],
    "Buckets": [ //This attribute is not supported for AWS Organizations-level
 configurations.
      "arn:aws:s3:::amzn-s3-demo-source-bucket "
    ]
  },
  "IsEnabled": true, //Whether the configuration is enabled
  "DataExport": { //Details about the metrics export
    "S3BucketDestination": {
      "OutputSchemaVersion": "V_1",
      "Format": "CSV", //You can add "Parquet" if you prefer.
      "AccountId": "111122223333",
      "Arn": "arn:aws:s3:::amzn-s3-demo-destination-bucket", // The destination
 bucket for your metrics export must be in the same Region as your S3 Storage Lens
 configuration.
      "Prefix": "prefix-for-your-export-destination",
      "Encryption": {
        "SSES3": {}
      }
    },
    "CloudWatchMetrics": {
      "IsEnabled": true //Mark this as false if you want to export only free metrics.
    }
  }
}
```

Menggunakan AWS SDK for Java

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.AccountLevel;
import com.amazonaws.services.s3control.model.ActivityMetrics;
import com.amazonaws.services.s3control.model.BucketLevel;
import com.amazonaws.services.s3control.model.CloudWatchMetrics;
import com.amazonaws.services.s3control.model.Format;
import com.amazonaws.services.s3control.model.Include;
import com.amazonaws.services.s3control.model.OutputSchemaVersion;
import com.amazonaws.services.s3control.model.PrefixLevel;
```

```
import com.amazonaws.services.s3control.model.PrefixLevelStorageMetrics;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.S3BucketDestination;
import com.amazonaws.services.s3control.model.SSES3;
import com.amazonaws.services.s3control.model.SelectionCriteria;
import com.amazonaws.services.s3control.model.StorageLensAwsOrg;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensDataExport;
import com.amazonaws.services.s3control.model.StorageLensDataExportEncryption;
import com.amazonaws.services.s3control.model.StorageLensTag;

import java.util.Arrays;
import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class CreateAndUpdateDashboard {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "Source Account ID";
        String exportAccountId = "Destination Account ID";
        String exportBucketArn = "arn:aws:s3:::amzn-s3-demo-destination-bucket"; //
 The destination bucket for your metrics export must be in the same Region as your S3
 Storage Lens configuration.
        String awsOrgARN = "arn:aws:organizations::123456789012:organization/o-
abcdefgh";
        Format exportFormat = Format.CSV;

        try {
            SelectionCriteria selectionCriteria = new SelectionCriteria()
                    .withDelimiter("/")
                    .withMaxDepth(5)
                    .withMinStorageBytesPercentage(10.0);
            PrefixLevelStorageMetrics prefixStorageMetrics = new
 PrefixLevelStorageMetrics()
                    .withIsEnabled(true)
                    .withSelectionCriteria(selectionCriteria);
            BucketLevel bucketLevel = new BucketLevel()
                    .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
                    .withAdvancedCostOptimizationMetrics(new
 AdvancedCostOptimizationMetrics().withIsEnabled(true))
                    .withAdvancedDataProtectionMetrics(new
 AdvancedDataProtectionMetrics().withIsEnabled(true))
```

```
                    .withDetailedStatusCodesMetrics(new
DetailedStatusCodesMetrics().withIsEnabled(true))
                    .withPrefixLevel(new
PrefixLevel().withStorageMetrics(prefixStorageMetrics));
            AccountLevel accountLevel = new AccountLevel()
                    .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
                    .withAdvancedCostOptimizationMetrics(new
AdvancedCostOptimizationMetrics().withIsEnabled(true))
                    .withAdvancedDataProtectionMetrics(new
AdvancedDataProtectionMetrics().withIsEnabled(true))
                    .withDetailedStatusCodesMetrics(new
DetailedStatusCodesMetrics().withIsEnabled(true))
                    .withBucketLevel(bucketLevel);

            Include include = new Include()
                    .withBuckets(Arrays.asList("arn:aws:s3:::amzn-s3-demo-bucket"))
                    .withRegions(Arrays.asList("us-west-2"));

            StorageLensDataExportEncryption exportEncryption = new
StorageLensDataExportEncryption()
                    .withSSES3(new SSES3());
            S3BucketDestination s3BucketDestination = new S3BucketDestination()
                    .withAccountId(exportAccountId)
                    .withArn(exportBucketArn)
                    .withEncryption(exportEncryption)
                    .withFormat(exportFormat)
                    .withOutputSchemaVersion(OutputSchemaVersion.V_1)
                    .withPrefix("Prefix");
            CloudWatchMetrics cloudWatchMetrics = new CloudWatchMetrics()
                    .withIsEnabled(true);
            StorageLensDataExport dataExport = new StorageLensDataExport()
                    .withCloudWatchMetrics(cloudWatchMetrics)
                    .withS3BucketDestination(s3BucketDestination);

            StorageLensAwsOrg awsOrg = new StorageLensAwsOrg()
                    .withArn(awsOrgARN);

            StorageLensConfiguration configuration = new StorageLensConfiguration()
                    .withId(configurationId)
                    .withAccountLevel(accountLevel)
                    .withInclude(include)
                    .withDataExport(dataExport)
                    .withAwsOrg(awsOrg)
                    .withIsEnabled(true);
```

```
            List<StorageLensTag> tags = Arrays.asList(
                    new StorageLensTag().withKey("key-1").withValue("value-1"),
                    new StorageLensTag().withKey("key-2").withValue("value-2")
            );

            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.putStorageLensConfiguration(new
 PutStorageLensConfigurationRequest()
                    .withAccountId(sourceAccountId)
                    .withConfigId(configurationId)
                    .withStorageLensConfiguration(configuration)
                    .withTags(tags)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Penggunaan API REST

Untuk mengaktifkan opsi CloudWatch penerbitan dengan menggunakan Amazon S3 REST API, Anda dapat menggunakannya. PutStorageLensConfiguration

Langkah selanjutnya

Setelah mengaktifkan opsi CloudWatch penerbitan, Anda dapat mengakses metrik Lensa Penyimpanan S3 Anda di. CloudWatch Anda juga dapat memanfaatkan CloudWatch fitur untuk memantau dan menganalisis data Lensa CloudWatch Penyimpanan S3 Anda. Untuk informasi selengkapnya, lihat topik berikut:

- Metrik dan dimensi Lensa Penyimpanan S3

- Bekerja dengan metrik Lensa Penyimpanan S3 di CloudWatch

Bekerja dengan metrik Lensa Penyimpanan S3 di CloudWatch

Anda dapat menerbitkan metrik Lensa Penyimpanan S3 CloudWatch ke Amazon untuk membuat tampilan terpadu kesehatan operasional Anda di dasbor. CloudWatch Anda juga dapat menggunakan CloudWatch fitur, seperti alarm dan tindakan yang dipicu, matematika metrik, dan deteksi anomali, untuk memantau dan mengambil tindakan pada metrik Lensa Penyimpanan S3. Selain itu, operasi CloudWatch API memungkinkan aplikasi, termasuk penyedia pihak ketiga, untuk mengakses metrik Lensa Penyimpanan S3 Anda. Untuk informasi selengkapnya tentang CloudWatch fitur, lihat Panduan CloudWatch Pengguna Amazon.

Anda dapat mengaktifkan opsi CloudWatch penerbitan untuk konfigurasi dasbor baru atau yang sudah ada dengan menggunakan konsol Amazon S3, Amazon S3 REST, dan. APIs AWS CLI AWS SDKs Opsi CloudWatch penerbitan tersedia untuk dasbor yang ditingkatkan ke metrik dan rekomendasi lanjutan S3 Storage Lens. Untuk metrik dan rekomendasi lanjutan, lihat Harga Lensa Penyimpanan Amazon S3. Tidak ada biaya penerbitan CloudWatch metrik tambahan yang berlaku; namun, CloudWatch biaya lain, seperti dasbor, alarm, dan panggilan API, berlaku. Untuk informasi selengkapnya, lihat CloudWatch harga Amazon.

Metrik Lensa Penyimpanan S3 dipublikasikan ke akun yang memiliki konfigurasi CloudWatch Lensa Penyimpanan S3. Setelah mengaktifkan opsi CloudWatch penerbitan dalam metrik dan rekomendasi lanjutan, Anda dapat mengakses metrik tingkat organisasi, akun, dan ember. CloudWatch Metrik tingkat awalan tidak tersedia di. CloudWatch

> **Note**
>
> Metrik Lensa Penyimpanan S3 adalah metrik harian dan dipublikasikan CloudWatch satu kali per hari. Saat Anda menanyakan metrik Lensa Penyimpanan S3 CloudWatch, periode kueri harus 1 hari (86400 detik). Setelah metrik Lensa Penyimpanan S3 harian Anda muncul di dasbor Lensa Penyimpanan S3 Anda di konsol Amazon S3, diperlukan beberapa jam agar metrik yang sama ini muncul. CloudWatch Saat Anda mengaktifkan opsi CloudWatch penerbitan untuk metrik Lensa Penyimpanan S3 untuk pertama kalinya, metrik Anda dapat dipublikasikan hingga 24 jam. CloudWatch
> Saat ini, metrik Lensa Penyimpanan S3 tidak dapat dikonsumsi melalui CloudWatch aliran.

Untuk informasi selengkapnya tentang bekerja dengan metrik Lensa Penyimpanan S3 di CloudWatch, lihat topik berikut.

Topik

- Bekerja dengan CloudWatch dasbor
- Mengatur alarm, memicu tindakan, dan menggunakan deteksi anomali
- Memfilter metrik dengan menggunakan dimensi
- Menghitung metrik baru dengan matematika metrik
- Menggunakan ekspresi pencarian di bagan

Bekerja dengan CloudWatch dasbor

Anda dapat menggunakan CloudWatch dasbor untuk memantau metrik Lensa Penyimpanan S3 bersama metrik aplikasi lainnya dan membuat tampilan terpadu tentang kesehatan operasional Anda. Dasbor adalah halaman beranda yang dapat disesuaikan di CloudWatch konsol yang dapat Anda gunakan untuk memantau sumber daya Anda dalam satu tampilan.

CloudWatch memiliki kontrol izin luas yang tidak mendukung pembatasan akses ke kumpulan metrik atau dimensi tertentu. Pengguna di akun atau organisasi Anda yang memiliki akses ke CloudWatch akan memiliki akses ke metrik untuk semua konfigurasi Lensa Penyimpanan S3 di mana opsi CloudWatch dukungan diaktifkan. Anda tidak dapat mengelola izin untuk dasbor tertentu seperti yang Anda bisa lakukan di Lensa Penyimpanan S3. Untuk informasi selengkapnya tentang CloudWatch izin, lihat Mengelola izin akses ke CloudWatch sumber daya Anda di CloudWatch Panduan Pengguna Amazon.

Untuk informasi selengkapnya tentang menggunakan CloudWatch dasbor dan mengonfigurasi izin, lihat Menggunakan CloudWatch dasbor Amazon dan CloudWatch dasbor Berbagi di Panduan Pengguna Amazon. CloudWatch

Mengatur alarm, memicu tindakan, dan menggunakan deteksi anomali

Anda dapat mengonfigurasi CloudWatch alarm yang menonton metrik Lensa Penyimpanan S3 CloudWatch dan mengambil tindakan saat ambang batas dilanggar. Misalnya, Anda dapat mengonfigurasi alarm yang mengirimkan notifikasi Amazon SNS jika metrik Byte Unggahan Multibagian Tidak Lengkap melebihi 1 GB selama tiga hari berturut-turut.

Anda juga dapat mengaktifkan deteksi anomali untuk terus menganalisis metrik Lensa Penyimpanan S3, menentukan garis dasar normal, dan anomali permukaan. Anda dapat membuat alarm deteksi

anomali berdasarkan nilai yang diharapkan dari metrik. Misalnya, Anda dapat memantau anomali untuk metrik Byte yang Mengaktifkan Kunci Objek untuk mendeteksi penghapusan pengaturan Kunci Objek yang tidak sah.

Untuk informasi dan contoh selengkapnya, lihat Menggunakan CloudWatch alarm Amazon dan Membuat alarm dari metrik pada grafik di Panduan CloudWatch Pengguna Amazon.

Memfilter metrik dengan menggunakan dimensi

Anda dapat menggunakan dimensi untuk memfilter metrik Lensa Penyimpanan S3 di konsol. CloudWatch Misalnya, Anda dapat memfilter berdasarkan `configuration_id`, `aws_account_number`, `aws_region`, `bucket_name`, dan lainnya.

Lensa Penyimpanan S3 mendukung beberapa konfigurasi dasbor per akun. Artinya, konfigurasi yang berbeda dapat menyertakan bucket yang sama. Saat metrik ini dipublikasikan CloudWatch, bucket akan memiliki metrik duplikat di dalamnya. CloudWatch Untuk melihat metrik hanya untuk konfigurasi Lensa Penyimpanan S3 tertentu di CloudWatch, Anda dapat menggunakan dimensi. `configuration_id` Saat memfilter berdasarkan `configuration_id`, Anda hanya melihat metrik yang terkait dengan konfigurasi yang diidentifikasi.

Untuk informasi selengkapnya tentang pemfilteran berdasarkan ID konfigurasi, lihat Mencari metrik yang tersedia di CloudWatch Panduan Pengguna Amazon.

Menghitung metrik baru dengan matematika metrik

Anda dapat menggunakan matematika metrik untuk mengkueri beberapa metrik Lensa Penyimpanan S3 dan menggunakan ekspresi matematika untuk membuat deret waktu baru berdasarkan metrik ini. Misalnya, Anda dapat membuat metrik baru untuk objek yang tidak terenkripsi dengan mengurangi Objek Terenkripsi dari Perhitungan Objek. Anda juga dapat membuat metrik untuk mendapatkan ukuran objek rata-rata dengan membagi `StorageBytes` dengan `ObjectCount` atau jumlah byte yang diakses pada satu hari dengan membagi `BytesDownloaded` dengan `StorageBytes`.

Untuk informasi selengkapnya, lihat Menggunakan matematika metrik di Panduan CloudWatch Pengguna Amazon.

Menggunakan ekspresi pencarian di bagan

Dengan metrik Lensa Penyimpanan S3, Anda dapat membuat ekspresi pencarian. Misalnya, Anda dapat membuat ekspresi penelusuran untuk semua metrik yang diberi nama IncompleteMultipartUploadStorageBytesdan ditambahkan `SUM` ke ekspresi. Dengan ekspresi

pencarian ini, Anda dapat melihat total byte unggahan multibagian yang tidak lengkap di semua dimensi penyimpanan di dalam satu metrik.

Contoh ini menunjukkan sintaks yang akan Anda gunakan untuk membuat ekspresi pencarian untuk semua metrik bernama. IncompleteMultipartUploadStorageBytes

```
SUM(SEARCH('{AWS/S3/Storage-
Lens,aws_account_number,aws_region,configuration_id,metrics_version,record_type,storage_class}
 MetricName="IncompleteMultipartUploadStorageBytes"', 'Average',86400))
```

Untuk informasi selengkapnya tentang sintaks ini, lihat sintaks ekspresi CloudWatch penelusuran di CloudWatch Panduan Pengguna Amazon. Untuk membuat CloudWatch grafik dengan ekspresi penelusuran, lihat Membuat CloudWatch grafik dengan ekspresi penelusuran di Panduan CloudWatch Pengguna Amazon.

Kasus penggunaan metrik Lensa Penyimpanan Amazon S3

Anda dapat menggunakan dasbor Lensa Penyimpanan Amazon S3 untuk memvisualisasikan wawasan dan tren, menandai outlier, dan menerima rekomendasi. Metrik Lensa Penyimpanan S3 disusun ke dalam kategori yang selaras dengan kasus penggunaan utama. Anda dapat menggunakan metrik ini untuk melakukan hal-hal berikut:

- Mengdentifikasi peluang optimalisasi biaya
- Menerapkan praktik terbaik untuk perlindungan data
- Menerapkan praktik terbaik untuk manajemen akses
- Meningkatkan kinerja beban kerja aplikasi

Misalnya, dengan metrik optimisasi biaya, Anda dapat mengidentifikasi peluang untuk mengurangi biaya penyimpanan Amazon S3. Anda dapat mengidentifikasi bucket dengan unggahan multibagian yang berumur lebih dari 7 hari atau bucket yang mengumpulkan versi yang tidak terkini.

Anda juga dapat menggunakan metrik perlindungan data untuk mengidentifikasi bucket yang tidak mengikuti praktik terbaik untuk perlindungan data di dalam organisasi. Misalnya, Anda dapat mengidentifikasi bucket yang tidak menggunakan AWS Key Management Service kunci (SSE-KMS) untuk enkripsi default atau tidak mengaktifkan Versi S3.

Dengan metrik manajemen akses S3 Storage Lens, Anda dapat mengidentifikasi setelan bucket untuk Kepemilikan Objek S3 sehingga Anda dapat memigrasikan izin daftar kontrol akses (ACL) ke kebijakan bucket dan menonaktifkannya. ACLs

Jika Anda mengaktifkan Metrik lanjutan Lensa Penyimpanan S3, Anda dapat menggunakan metrik kode status terperinci untuk mendapatkan jumlah permintaan yang berhasil atau gagal yang dapat digunakan untuk memecahkan masalah akses atau kinerja.

Dengan metrik lanjutan, Anda juga dapat mengakses metrik optimisasi biaya dan perlindungan data tambahan yang dapat digunakan untuk mengidentifikasi peluang untuk lebih mengurangi biaya penyimpanan S3 secara keseluruhan dan lebih selaras dengan praktik terbaik untuk melindungi data. Misalnya, metrik optimisasi biaya lanjutan menyertakan jumlah aturan siklus hidup yang dapat digunakan untuk mengidentifikasi bucket yang tidak memiliki aturan siklus hidup untuk menghentikan masa berlaku unggahan multibagian yang tidak lengkap dan berumur lebih dari 7 hari. Metrik perlindungan data lanjutan mencakup jumlah aturan replikasi.

Untuk informasi selengkapnya tentang kategori metrik, lihat Kategori metrik. Untuk daftar lengkap metrik Lensa Penyimpanan S3, lihat Glosarium metrik Lensa Penyimpanan Amazon S3.

Topik

- Menggunakan Lensa Penyimpanan Amazon S3 untuk mengoptimalkan biaya penyimpanan Anda
- Menggunakan Lensa Penyimpanan S3 untuk melindungi data
- Menggunakan Lensa Penyimpanan S3 untuk mengaudit pengaturan Kepemilikan Objek
- Menggunakan metrik Lensa Penyimpanan S3 untuk meningkatkan kinerja

Menggunakan Lensa Penyimpanan Amazon S3 untuk mengoptimalkan biaya penyimpanan Anda

Anda dapat menggunakan metrik pengoptimalan biaya Lensa Penyimpanan S3 untuk mengurangi biaya keseluruhan penyimpanan S3 Anda. Metrik pengoptimalan biaya dapat membantu Anda mengonfirmasi bahwa Anda telah mengonfigurasi Amazon S3 secara efektif dan sesuai dengan praktik terbaik. Misalnya, Anda dapat mengidentifikasi peluang pengoptimalan biaya berikut:

- Bucket dengan unggahan multipart yang tidak lengkap lebih dari 7 hari
- Bucket yang mengumpulkan banyak versi tidak terkini
- Bucket yang tidak memiliki aturan siklus hidup untuk membatalkan unggahan multipart yang tidak lengkap
- Bucket yang tidak memiliki aturan siklus hidup untuk kedaluwarsa objek versi noncurrent
- Bucket yang tidak memiliki aturan siklus hidup untuk mentransisikan objek ke kelas penyimpanan yang berbeda

Anda kemudian dapat menggunakan data ini untuk menambahkan aturan siklus hidup tambahan ke bucket Anda.

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan metrik pengoptimalan biaya di dasbor S3 Lensa Penyimpanan Anda untuk mengoptimalkan biaya penyimpanan.

Topik

- Identifikasi bucket S3 terbesar Anda
- Temukan bucket Amazon S3 dingin
- Temukan unggahan multipart yang tidak lengkap
- Kurangi jumlah versi noncurrent yang dipertahankan
- Identifikasi bucket yang tidak memiliki aturan siklus hidup dan tinjau jumlah aturan siklus hidup

Identifikasi bucket S3 terbesar Anda

Anda membayar untuk menyimpan objek dalam bucket S3. Tarif yang dikenakan biaya tergantung pada ukuran objek Anda, berapa lama Anda menyimpan objek, dan kelas penyimpanannya. Dengan Lensa Penyimpanan S3, Anda mendapatkan tampilan terpusat dari semua bucket di akun Anda. Untuk melihat semua bucket di semua akun organisasi, Anda dapat mengonfigurasi dasbor Lensa Penyimpanan S3 AWS Organizations tingkat -level. Dari tampilan dasbor ini, Anda dapat mengidentifikasi bucket terbesar Anda.

Langkah 1: Identifikasi bucket terbesar Anda

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, pilih Lensa Penyimpanan, Dasbor.
3. Di daftar Dasbor, pilih dasbor yang ingin Anda lihat.

   Saat dasbor terbuka, Anda dapat melihat tanggal terakhir S3 Lensa Penyimpanan mengumpulkan metrik. Dasbor Anda selalu dimuat ke tanggal terbaru yang memiliki metrik yang tersedia.
4. Untuk melihat peringkat bucket terbesar Anda berdasarkan metrik Total penyimpanan untuk rentang tanggal yang dipilih, gulir ke bawah ke bagian ikhtisar N Teratas untuk tanggal.

   Anda dapat mengaktifkan urutan pengurutan untuk menampilkan bucket terkecil. Anda juga dapat menyesuaikan pilihan Metrik untuk menentukan peringkat bucket berdasarkan metrik

mana pun yang tersedia. Ikhtisar N Teratas untuk bagian tanggal juga menunjukkan persentase perubahan dari hari atau minggu sebelumnya dan garis percikan untuk memvisualisasikan tren. Tren ini adalah tren 14 hari untuk metrik gratis dan tren 30 hari untuk metrik dan rekomendasi lanjutan.

> **Note**
>
> Dengan metrik dan rekomendasi lanjutan Lensa Penyimpanan S3, metrik tersedia untuk kueri selama 15 bulan. Untuk informasi selengkapnya, lihat Pilihan metrik.

5. Untuk wawasan yang lebih mendetail tentang bucket Anda, gulir ke atas ke bagian atas halaman, lalu pilih tab Bucket.

   Pada tab Bucket, Anda dapat melihat detail seperti tingkat pertumbuhan terbaru, ukuran objek rata-rata, awalan terbesar, dan jumlah objek.

Langkah 2: Arahkan ke bucket Anda dan selidiki

Setelah mengidentifikasi bucket S3 terbesar, Anda dapat menavigasi ke setiap bucket dalam konsol S3 untuk melihat objek di bucket, memahami beban kerja terkait, dan mengidentifikasi pemilik internalnya. Anda dapat menghubungi pemilik bucket untuk mengetahui apakah pertumbuhan diharapkan atau apakah pertumbuhan membutuhkan pemantauan dan kontrol lebih lanjut.

Temukan bucket Amazon S3 dingin

Jika Anda mengaktifkan metrik lanjutan Lensa Penyimpanan S3, Anda dapat menggunakan metrik aktivitas untuk memahami seberapa dingin bucket S3 Anda. Bucket “dingin” adalah bucket yang penyimpanannya tidak lagi diakses (atau sangat jarang diakses). Kurangnya aktivitas ini biasanya menunjukkan bahwa objek bucket tidak sering diakses.

Metrik aktivitas, seperti Permintaan GET dan Byte Unduhan, menunjukkan seberapa sering bucket Anda diakses setiap hari. Untuk memahami konsistensi pola akses dan menemukan bucket yang tidak lagi diakses sama sekali, Anda dapat membuat tren data ini selama beberapa bulan. Metrik tingkat pengambilan, yang dihitung sebagai Download byte/Total storage, menunjukkan proporsi penyimpanan dalam bucket yang diakses setiap hari.

> **Note**
>
> Byte unduhan diduplikasi dalam kasus di mana objek yang sama diunduh beberapa kali di siang hari.

Prasyarat

Untuk melihat metrik aktivitas di dasbor Lensa Penyimpanan S3, Anda harus mengaktifkan metrik dan rekomendasi lanjutan Lensa Penyimpanan S3, lalu pilih Metrik aktivitas. Untuk informasi selengkapnya, lihat Menggunakan konsol S3.

Langkah 1: Identifikasi bucket aktif

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, pilih Lensa Penyimpanan, Dasbor.
3. Di daftar Dasbor, pilih dasbor yang ingin Anda lihat.
4. Pilih tab Bucket, lalu gulir ke bawah ke bagian Analisis gelembung dengan ember untuk tanggal.

   Di bagian Analisis gelembung berdasarkan bucket untuk tanggal, Anda dapat memplot bucket pada beberapa dimensi dengan menggunakan tiga metrik mana pun untuk mewakili sumbu X, sumbu Y, dan Ukuran gelembung.
5. Untuk menemukan bucket yang sudah dingin, untuk sumbu X, sumbu Y, dan Ukuran, pilih metrik Total storage, % retrieval rate, dan Average object size.
6. Di bagian Analisis gelembung dengan bucket untuk tanggal, temukan bucket apa pun dengan tingkat pengambilan nol (atau mendekati nol) dan ukuran penyimpanan relatif yang lebih besar, dan pilih gelembung yang mewakili bucket.

   Sebuah kotak berisi pilihan untuk wawasan yang lebih terperinci akan ditampilkan. Lakukan salah satu hal berikut ini:

   a. Untuk memperbarui tab Bucket agar hanya menampilkan metrik untuk bucket yang dipilih, pilih Telusuri, lalu pilih Terapkan.

   b. Untuk menggabungkan data tingkat ember menurut akun, kelas penyimpanan, atau bucket Wilayah AWS, pilih Analyze by, lalu tentukan pilihan untuk Dimensi. Misalnya, untuk agregat berdasarkan kelas penyimpanan, pilih kelas Penyimpanan untuk Dimensi.

Untuk menemukan bucket yang kedinginan, lakukan analisis gelembung menggunakan metrik Total storage, % retrieval rate, dan Average object size. Cari bucket apa pun dengan tingkat pengambilan nol (atau mendekati nol) dan ukuran penyimpanan relatif yang lebih besar.

Tab Bucket di dasbor Anda diperbarui untuk menampilkan data untuk agregasi atau filter yang Anda pilih. Jika Anda digabungkan berdasarkan kelas penyimpanan atau dimensi lain, tab baru itu terbuka di dasbor Anda (misalnya, tab kelas Penyimpanan).

Langkah 2: Selidiki bucket dingin

Dari sini, Anda dapat mengidentifikasi pemilik bucket dingin di akun atau organisasi Anda dan mencari tahu apakah penyimpanan itu masih diperlukan. Anda kemudian dapat mengoptimalkan biaya dengan mengonfigurasi konfigurasi kedaluwarsa siklus hidup untuk bucket ini atau mengarsipkan data di salah satu kelas penyimpanan.Amazon S3 Glacier

Untuk menghindari masalah bucket dingin di masa depan, Anda dapat secara otomatis mentransisikan data Anda dengan menggunakan konfigurasi Siklus Hidup S3 untuk bucket Anda, atau Anda dapat mengaktifkan pengarsipan otomatis dengan S3 Intelligent-Tiering.

Anda juga dapat menggunakan langkah 1 untuk mengidentifikasi bucket panas. Kemudian, Anda dapat memastikan bahwa bucket ini menggunakan kelas penyimpanan S3 yang benar untuk memastikan bahwa mereka melayani permintaan mereka dengan paling efektif dalam hal kinerja dan biaya.

Temukan unggahan multipart yang tidak lengkap

Anda dapat menggunakan unggahan multipart untuk mengunggah objek yang sangat besar (hingga 5 TB) sebagai satu set bagian untuk meningkatkan throughput dan pemulihan yang lebih cepat dari masalah jaringan. Jika proses pengunggahan multibagian tidak selesai, bagian yang tidak lengkap tetap berada di bucket (dalam keadaan tidak dapat digunakan). Suku cadang yang tidak lengkap ini menimbulkan biaya penyimpanan sampai proses pengunggahan selesai, atau sampai bagian yang tidak lengkap dilepas. Untuk informasi selengkapnya, lihat Mengunggah dan menyalin objek menggunakan unggahan multipart di Amazon S3.

Dengan Lensa Penyimpanan S3, Anda dapat mengidentifikasi jumlah byte unggahan multipart yang tidak lengkap di akun Anda atau di seluruh organisasi Anda, termasuk unggahan multipart yang tidak lengkap yang berusia lebih dari 7 hari. Untuk daftar lengkap metrik unggahan multibagian yang tidak lengkap, lihat. Glosarium metrik Lensa Penyimpanan Amazon S3

Sebagai praktik terbaik, kami menyarankan untuk mengonfigurasi aturan siklus hidup untuk mengakhiri unggahan multibagian yang tidak lengkap yang lebih lama dari jumlah hari tertentu. Saat Anda membuat aturan siklus hidup untuk mengakhiri unggahan multibagian yang tidak lengkap, kami merekomendasikan 7 hari sebagai titik awal yang baik.

Langkah 1: Tinjau tren keseluruhan untuk unggahan multipart yang tidak lengkap

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, pilih Lensa Penyimpanan, Dasbor.
3. Di daftar Dasbor, pilih dasbor yang ingin Anda lihat.
4. Di bagian Snapshot untuk tanggal, di bawah kategori Metrik, pilih Pengoptimalan biaya.

   Bagian Snapshot untuk tanggal diperbarui untuk menampilkan metrik pengoptimalan biaya, yang mencakup byte unggahan multibagian tidak lengkap yang berusia lebih dari 7 hari.

   Di bagan apa pun di dasbor S3 Lensa Penyimpanan, Anda dapat melihat metrik unggahan multipart yang tidak lengkap. Anda dapat menggunakan metrik ini untuk menilai lebih lanjut dampak byte unggahan multibagian yang tidak lengkap pada penyimpanan Anda, termasuk kontribusinya terhadap tren pertumbuhan secara keseluruhan. Anda juga dapat menelusuri ke tingkat agregasi yang lebih dalam, menggunakan tab Akun, Wilayah AWS, Bucket, atau kelas Penyimpanan untuk analisis data yang lebih dalam. Sebagai contoh, lihat Temukan bucket Amazon S3 dingin.

Langkah 2: Identifikasi bucket yang memiliki byte unggahan multibagian paling tidak lengkap tetapi tidak memiliki aturan siklus hidup untuk membatalkan unggahan multibagian yang tidak lengkap

Prasyarat

Untuk melihat metrik Batalkan penghitungan aturan siklus hidup unggahan multibagian yang tidak lengkap di dasbor Lensa Penyimpanan S3, Anda harus mengaktifkan Metrik dan rekomendasi Lanjutan Lensa Penyimpanan S3, lalu pilih Metrik pengoptimalan biaya lanjutan. Untuk informasi selengkapnya, lihat Menggunakan konsol S3.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, pilih Lensa Penyimpanan, Dasbor.
3. Di daftar Dasbor, pilih dasbor yang ingin Anda lihat.

4. Untuk mengidentifikasi bucket tertentu yang mengumpulkan unggahan multipart yang tidak lengkap lebih dari 7 hari, buka bagian Ikhtisar N Teratas untuk tanggal.

   Secara default, ikhtisar N Teratas untuk bagian tanggal menampilkan metrik untuk 3 bucket teratas. Anda dapat menambah atau mengurangi jumlah bucket di bidang N Atas. Ikhtisar N Teratas untuk bagian tanggal juga menunjukkan persentase perubahan dari hari atau minggu sebelumnya dan garis percikan untuk memvisualisasikan tren. (Tren ini adalah tren 14 hari untuk metrik gratis dan tren 30 hari untuk metrik dan rekomendasi lanjutan.)

   > **Note**
   >
   > Dengan metrik dan rekomendasi lanjutan Lensa Penyimpanan S3, metrik tersedia untuk kueri selama 15 bulan. Untuk informasi selengkapnya, lihat Pilihan metrik.

5. Untuk Metric, pilih Unggahan byte multibagian tidak lengkap yang berumur lebih dari 7 hari dalam kategori Pengoptimalan biaya.

   Di bawah Bucket nomor teratas, Anda dapat melihat bucket dengan byte penyimpanan unggahan multibagian paling tidak lengkap yang berusia lebih dari 7 hari.

6. Untuk melihat metrik tingkat ember yang lebih detail untuk unggahan multibagian yang tidak lengkap, gulir ke bagian atas halaman, lalu pilih tab Bucket.
7. Gulir ke bawah ke bagian Bucket. Untuk kategori Metrik, pilih Optimasi biaya. Kemudian hapus Ringkasan.

   Daftar Bucket diperbarui untuk menampilkan semua metrik pengoptimalan Biaya yang tersedia untuk bucket yang ditampilkan.

8. Untuk memfilter daftar Bucket agar hanya menampilkan metrik pengoptimalan biaya tertentu, pilih ikon preferensi ().

9. Hapus sakelar untuk semua metrik pengoptimalan biaya hingga hanya Byte unggahan multibagian tidak lengkap yang berumur lebih dari 7 hari dan Batalkan jumlah aturan siklus hidup unggahan multibagian yang tidak lengkap tetap dipilih.
10. (Opsional) Di bawah Ukuran halaman, pilih jumlah bucket yang akan ditampilkan di daftar.
11. Pilih Konfirmasi.

    Daftar Bucket diperbarui untuk menampilkan metrik tingkat ember untuk unggahan multibagian yang tidak lengkap dan jumlah aturan siklus hidup. Anda dapat menggunakan data ini untuk

mengidentifikasi bucket yang memiliki byte unggahan multibagian paling tidak lengkap yang berusia lebih dari 7 hari dan tidak memiliki aturan siklus hidup untuk membatalkan unggahan multibagian yang tidak lengkap. Kemudian, Anda dapat menavigasi ke bucket ini di konsol S3 dan menambahkan aturan siklus hidup untuk menghapus unggahan multibagian yang tidak lengkap yang ditinggalkan.

Langkah 3: Tambahkan aturan siklus hidup untuk menghapus unggahan multipart yang tidak lengkap setelah 7 hari

Untuk mengelola unggahan multibagian yang tidak lengkap secara otomatis, Anda dapat menggunakan konsol S3 untuk membuat konfigurasi siklus hidup untuk menghapus byte unggahan multibagian yang tidak lengkap dari bucket setelah beberapa hari tertentu. Untuk informasi selengkapnya, lihat Mengonfigurasi siklus hidup bucket untuk menghapus unggahan multibagian yang tidak lengkap.

Kurangi jumlah versi noncurrent yang dipertahankan

Saat diaktifkan, S3 Penentuan Versi menyimpan beberapa salinan berbeda dari objek yang sama yang dapat Anda gunakan untuk memulihkan data dengan cepat jika suatu objek secara tidak sengaja dihapus atau ditimpa. Jika Anda telah mengaktifkan Pembuatan Versi S3 tanpa mengonfigurasi aturan siklus hidup untuk transisi atau kedaluwarsa versi noncurrent, sejumlah besar versi noncurrent sebelumnya dapat terakumulasi, yang dapat memiliki implikasi biaya penyimpanan. Untuk informasi selengkapnya, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.

Langkah 1: Identifikasi bucket dengan versi objek paling tidak terkini

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, pilih Lensa Penyimpanan, Dasbor.
3. Di daftar Dasbor, pilih dasbor yang ingin Anda lihat.
4. Di bagian Snapshot untuk tanggal, di bawah Kategori metrik, pilih Optimasi biaya.

   Bagian Snapshot untuk tanggal diperbarui untuk menampilkan metrik pengoptimalan biaya, yang mencakup metrik untuk% byte versi noncurrent. Metrik% noncurrent version byte mewakili proporsi total byte penyimpanan Anda yang dikaitkan dengan versi noncurrent, dalam lingkup dasbor dan untuk tanggal yang dipilih.

> **Note**
>
> Jika% byte versi noncurrent Anda lebih besar dari 10 persen penyimpanan Anda di tingkat akun, Anda mungkin menyimpan terlalu banyak versi objek.

5. Untuk mengidentifikasi bucket tertentu yang mengumpulkan sejumlah besar versi noncurrent:

   a. Gulir ke bawah ke bagian ikhtisar N Teratas untuk tanggal. Untuk Top N, masukkan jumlah bucket yang ingin Anda lihat datanya.

   b. Untuk Metric, pilih% byte bukan versi terkini.

      Di bawah Bucket nomor teratas, Anda dapat melihat bucket (untuk nomor yang Anda tentukan) dengan byte versi noncurrent% tertinggi. Ikhtisar N Teratas untuk bagian tanggal juga menunjukkan persentase perubahan dari hari atau minggu sebelumnya dan garis percikan untuk memvisualisasikan tren. Tren ini adalah tren 14 hari untuk metrik gratis dan tren 30 hari untuk metrik dan rekomendasi lanjutan.

      > **Note**
      >
      > Dengan metrik dan rekomendasi lanjutan Lensa Penyimpanan S3, metrik tersedia untuk kueri selama 15 bulan. Untuk informasi selengkapnya, lihat Pilihan metrik.

   c. Untuk melihat metrik tingkat ember yang lebih detail untuk versi objek noncurrent, gulir ke bagian atas halaman, lalu pilih tab Bucket.

      Dalam bagan atau visualisasi apa pun di dasbor Lensa Penyimpanan S3, Anda dapat menelusuri ke tingkat agregasi yang lebih dalam, menggunakan tab Akun, Wilayah AWS, kelas Penyimpanan, atau Bucket. Sebagai contoh, lihat Temukan bucket Amazon S3 dingin.

   d. Di bagian Bucket, untuk kategori Metrik, pilih Optimasi biaya. Kemudian, jelas Ringkasan.

      Anda sekarang dapat melihat metrik byte versi noncurrent%, bersama dengan metrik lain yang terkait dengan versi noncurrent.

Langkah 2: Identifikasi bucket yang tidak memiliki aturan siklus hidup transisi dan kedaluwarsa untuk mengelola versi noncurrent

Prasyarat

Untuk melihat Jumlah aturan siklus hidup transisi versi noncurrent dan metrik Hitungan aturan siklus hidup kedaluwarsa versi Noncurrent di dasbor Lensa Penyimpanan S3, Anda harus mengaktifkan Metrik dan rekomendasi lanjutan Lensa Penyimpanan S3, lalu pilih Metrik pengoptimalan biaya lanjutan. Untuk informasi selengkapnya, lihat Menggunakan konsol S3.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, pilih Lensa Penyimpanan, Dasbor.
3. Di daftar Dasbor, pilih dasbor yang ingin Anda lihat.
4. Di dasbor Lensa Penyimpanan, pilih tab Bucket.
5. Gulir ke bawah ke bagian Bucket. Untuk kategori Metrik, pilih Optimasi biaya. Kemudian hapus Ringkasan.

   Daftar Bucket diperbarui untuk menampilkan semua metrik pengoptimalan Biaya yang tersedia untuk bucket yang ditampilkan.
6. Untuk memfilter daftar Bucket agar hanya menampilkan metrik pengoptimalan biaya tertentu, pilih ikon preferensi ().

7. Hapus sakelar untuk semua metrik pengoptimalan biaya hingga hanya yang berikut yang tetap dipilih:

   - % byte bukan versi terkini
   - Jumlah aturan siklus hidup transisi versi tidak saat ini
   - Jumlah aturan siklus hidup kedaluwarsa bukan versi terkini
8. (Opsional) Di bawah Ukuran halaman, pilih jumlah bucket yang akan ditampilkan di daftar.
9. Pilih Konfirmasi.

   Daftar Bucket diperbarui untuk menampilkan metrik untuk byte versi noncurrent dan jumlah aturan siklus hidup versi noncurrent. Anda dapat menggunakan data ini untuk mengidentifikasi bucket yang memiliki persentase byte versi noncurrent yang tinggi namun tidak memiliki aturan siklus hidup transisi dan kedaluwarsa. Kemudian, Anda dapat menavigasi ke bucket ini di konsol S3 dan menambahkan aturan siklus hidup ke bucket ini.

Langkah 3: Tambahkan aturan siklus hidup untuk transisi atau kedaluwarsa versi objek noncurrent

Setelah menentukan bucket mana yang memerlukan penyelidikan lebih lanjut, Anda dapat menavigasi ke bucket dalam konsol S3 dan menambahkan aturan siklus hidup untuk kedaluwarsa versi noncurrent setelah beberapa hari tertentu. Atau, untuk mengurangi biaya sambil tetap mempertahankan versi noncurrent, Anda dapat mengonfigurasi aturan siklus hidup untuk mentransisikan versi noncurrent ke salah satu kelas penyimpanan. Amazon S3 Glacier Untuk informasi selengkapnya, lihat Menentukan aturan siklus aktif untuk bucket yang diaktifkan versioning-nya.

Identifikasi bucket yang tidak memiliki aturan siklus hidup dan tinjau jumlah aturan siklus hidup

Lensa Penyimpanan S3 menyediakan metrik jumlah aturan Siklus Hidup S3 yang dapat Anda gunakan untuk mengidentifikasi bucket yang tidak memiliki aturan siklus hidup. Untuk menemukan bucket yang tidak memiliki aturan siklus hidup, Anda dapat menggunakan metrik Total bucket tanpa aturan siklus hidup. Bucket tanpa konfigurasi Siklus Hidup S3 mungkin memiliki penyimpanan yang tidak lagi Anda perlukan atau dapat bermigrasi ke kelas penyimpanan berbiaya lebih rendah. Anda juga dapat menggunakan metrik jumlah aturan siklus hidup untuk mengidentifikasi bucket yang tidak memiliki jenis aturan siklus hidup tertentu, seperti aturan kedaluwarsa atau transisi.

Prasyarat

Untuk melihat metrik jumlah aturan siklus hidup dan metrik Total bucket tanpa aturan siklus hidup di dasbor Lensa Penyimpanan S3, Anda harus mengaktifkan Metrik dan rekomendasi S3 Lensa Penyimpanan S3, lalu memilih Metrik pengoptimalan biaya lanjutan. Untuk informasi selengkapnya, lihat Menggunakan konsol S3.

Langkah 1: Identifikasi bucket tanpa aturan siklus hidup

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, pilih Lensa Penyimpanan, Dasbor.
3. Di daftar Dasbor, pilih dasbor yang ingin Anda lihat.
4. Untuk mengidentifikasi bucket tertentu tanpa aturan siklus hidup, gulir ke bawah ke bagian Ikhtisar N Teratas untuk tanggal.

   Secara default, ikhtisar N Teratas untuk bagian tanggal menampilkan metrik untuk 3 bucket teratas. Di bidang N teratas, Anda dapat menambah jumlah bucket. Ikhtisar N Teratas untuk

bagian tanggal juga menunjukkan persentase perubahan dari hari atau minggu sebelumnya dan garis percikan untuk memvisualisasikan tren. Tren ini adalah tren 14 hari untuk metrik gratis dan tren 30 hari untuk metrik dan rekomendasi lanjutan.

> **Note**
>
> Dengan metrik dan rekomendasi lanjutan Lensa Penyimpanan S3, metrik tersedia untuk kueri selama 15 bulan. Untuk informasi selengkapnya, lihat Pilihan metrik.

5. Untuk Metrik, pilih Total bucket tanpa aturan siklus hidup dari kategori Pengoptimalan biaya.
6. Tinjau data berikut untuk Total bucket tanpa aturan siklus hidup:

   - Akun nomor teratas-Lihat akun mana yang memiliki bucket terbanyak tanpa aturan siklus hidup.
   - Wilayah nomor teratas-Lihat rincian bucket tanpa aturan siklus hidup menurut Wilayah.
   - Bucket nomor teratas-Lihat bucket mana yang tidak memiliki aturan siklus hidup.

   Dalam bagan atau visualisasi apa pun di dasbor Lensa Penyimpanan S3, Anda dapat menelusuri ke tingkat agregasi yang lebih dalam, menggunakan tab Akun, Wilayah AWS, kelas Penyimpanan, atau Bucket. Sebagai contoh, lihat Temukan bucket Amazon S3 dingin.

   Setelah mengidentifikasi bucket mana yang tidak memiliki aturan siklus hidup, Anda juga dapat meninjau jumlah aturan siklus hidup tertentu untuk bucket Anda.

Langkah 2: Tinjau jumlah aturan siklus hidup untuk bucket Anda

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, pilih Lensa Penyimpanan, Dasbor.
3. Di daftar Dasbor, pilih dasbor yang ingin Anda lihat.
4. Di dasbor S3 Lensa Penyimpanan Anda, pilih tab Bucket.
5. Gulir ke bawah ke bagian Bucket. Di bawah kategori Metrik, pilih Optimasi biaya. Kemudian hapus Ringkasan.

   Daftar Bucket diperbarui untuk menampilkan semua metrik pengoptimalan Biaya yang tersedia untuk bucket yang ditampilkan.

6. Untuk memfilter daftar Bucket agar hanya menampilkan metrik pengoptimalan biaya tertentu, pilih ikon preferensi ().

7. Hapus sakelar untuk semua metrik pengoptimalan biaya hingga hanya yang berikut yang tetap dipilih:

   - Jumlah aturan siklus hidup transisi
   - Jumlah aturan siklus hidup kedaluwarsa
   - Jumlah aturan siklus hidup transisi versi tidak saat ini
   - Jumlah aturan siklus hidup kedaluwarsa bukan versi terkini
   - Batalkan jumlah aturan siklus hidup unggahan multibagian yang tidak lengkap
   - Jumlah aturan siklus hidup total

8. (Opsional) Di bawah Ukuran halaman, pilih jumlah bucket yang akan ditampilkan di daftar.
9. Pilih Konfirmasi.

   Daftar Bucket diperbarui untuk menampilkan metrik jumlah aturan siklus hidup untuk bucket Anda. Anda dapat menggunakan data ini untuk mengidentifikasi bucket tanpa aturan siklus hidup atau bucket yang tidak memiliki jenis aturan siklus hidup tertentu, misalnya aturan kedaluwarsa atau transisi. Kemudian, Anda dapat menavigasi ke bucket ini di konsol S3 dan menambahkan aturan siklus hidup ke bucket ini.

Langkah 3: Tambahkan aturan siklus hidup

Setelah mengidentifikasi bucket tanpa aturan siklus hidup, Anda dapat menambahkan aturan siklus hidup. Untuk informasi selengkapnya, lihat Menyetel konfigurasi Siklus Hidup S3 pada bucket dan Contoh konfigurasi Siklus Hidup S3.

Menggunakan Lensa Penyimpanan S3 untuk melindungi data

Anda dapat menggunakan metrik perlindungan data Lensa Penyimpanan Amazon S3 untuk mengidentifikasi bucket yang belum menerapkan praktik terbaik perlindungan data. Anda dapat menggunakan metrik ini untuk mengambil tindakan dan menerapkan setelan standar yang selaras dengan praktik terbaik untuk melindungi data di seluruh bucket di akun atau organisasi Anda. Misalnya, Anda dapat menggunakan metrik perlindungan data untuk mengidentifikasi bucket yang tidak menggunakan kunci AWS Key Management Service (AWS KMS) (SSE-KMS) untuk enkripsi default atau permintaan yang menggunakan AWS Signature Version 2 (SigV2).

Kasus penggunaan berikut menyediakan strategi untuk menggunakan dasbor Lensa Penyimpanan S3 dalam mengidentifikasi outlier dan menerapkan praktik terbaik perlindungan data di seluruh bucket S3 Anda.

Topik

- Identifikasi bucket yang tidak menggunakan enkripsi sisi server AWS KMS untuk enkripsi default (SSE-KMS)
- Identifikasi bucket yang mengaktifkan Penentuan Versi S3
- Identifikasi permintaan yang menggunakan AWS Signature Version 2 (SigV2)
- Hitung jumlah total aturan replikasi untuk setiap bucket
- Identifikasi persentase byte Kunci Objek

Identifikasi bucket yang tidak menggunakan enkripsi sisi server AWS KMS untuk enkripsi default (SSE-KMS)

Dengan enkripsi default Amazon S3, Anda dapat menyetel perilaku enkripsi default untuk bucket S3. Untuk informasi selengkapnya, lihat the section called “Mengatur enkripsi bucket default”.

Anda dapat menggunakan jumlah bucket berkemampuan SSE-KMS dan metrik bucket berkemampuan% SSE-KMS untuk mengidentifikasi bucket yang menggunakan enkripsi sisi server dengan kunci (SSE-KMS) untuk enkripsi default. AWS KMS Lensa Penyimpanan S3 juga menyediakan metrik untuk byte yang tidak terenkripsi, objek yang tidak terenkripsi, byte yang terenkripsi, dan objek yang terenkripsi. Untuk daftar lengkap metrik, lihat Glosarium metrik Lensa Penyimpanan Amazon S3.

Anda dapat menganalisis metrik enkripsi SSE-KMS di dalam konteks metrik enkripsi umum untuk mengidentifikasi bucket yang tidak menggunakan SSE-KMS. Jika Anda ingin menggunakan SSE-KMS untuk semua bucket di akun atau organisasi, Anda kemudian dapat memperbarui pengaturan enkripsi default untuk bucket ini agar menggunakan SSE-KMS. Selain SSE-KMS, Anda juga dapat menggunakan enkripsi di sisi server dengan kunci yang dikelola Amazon S3 (SSE-S3) atau kunci yang disediakan pelanggan (SSE-C). Untuk informasi selengkapnya, lihat Melindungi data dengan enkripsi.

Langkah 1: Identifikasi bucket mana yang menggunakan SSE-KMS untuk enkripsi default

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

2. Di panel navigasi bagian kiri, pilih Lensa Penyimpanan, Dasbor.
3. Di daftar Dasbor, pilih nama dasbor yang ingin dilihat.
4. Di bagian Tren dan distribusi, pilih% jumlah bucket yang mendukung SSE-KMS untuk metrik primer dan % byte terenkripsi untuk metrik sekunder.

   Tren untuk bagan tanggal diperbarui dengan tujuan menampilkan tren untuk SSE-KMS dan byte terenkripsi.
5. Untuk melihat wawasan tingkat bucket yang lebih terperinci untuk SSE-KMS:

   a. Pilih satu titik pada bagan. Sebuah kotak berisi pilihan untuk wawasan yang lebih terperinci akan ditampilkan.
   b. Pilih dimensi Bucket. Lalu, pilih Terapkan.
6. Di bagan Distribusi berdasarkan bucket untuk tanggal, pilih metrik jumlah bucket yang mendukung SSE-KMS.
7. Anda sekarang dapat melihat bucket mana yang mengaktifkan SSE-KMS dan bucket mana yang tidak.

Langkah 2: Perbarui pengaturan enkripsi default bucket

Karena Anda sudah menentukan bucket mana yang menggunakan SSE-KMS di dalam konteks % yang byte terenkripsi, Anda dapat mengidentifikasi bucket yang tidak menggunakan SSE-KMS. Kemudian, Anda dapat menavigasi secara opsional ke bucket ini di dalam konsol S3 dan memperbarui pengaturan enkripsi defaultnya untuk menggunakan SSE-KMS atau SSE-S3. Untuk informasi selengkapnya, lihat Mengonfigurasi enkripsi default.

Identifikasi bucket yang mengaktifkan Penentuan Versi S3

Saat diaktifkan, fitur S3 Penentuan Versi akan mempertahankan beberapa versi dari objek yang sama yang dapat digunakan untuk memulihkan data dengan cepat jika suatu objek dihapus atau ditimpa secara tidak sengaja. Anda dapat menggunakan metrik jumlah bucket yang mengaktifkan Penentuan Versi untuk melihat bucket mana yang menggunakan Penentuan Versi S3. Kemudian, Anda dapat mengambil tindakan di konsol S3 untuk mengaktifkan Penentuan Versi S3 untuk bucket lain.

Langkah 1: Identifikasi bucket yang mengaktifkan Penentuan Versi S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

2. Di panel navigasi, pilih Lensa Penyimpanan, Dasbor.
3. Di daftar Dasbor, pilih nama dasbor yang ingin dilihat.
4. Di bagian Tren dan distribusi, pilih jumlah bucket yang mengaktifkan Penentuan Versi untuk metrik primer dan Bucket untuk metrik sekunder.

   Tren untuk bagan tanggal diperbarui dengan tujuan menampilkan tren untuk bucket yang mengaktifkan Penentuan Versi S3. Tepat di bawah garis tren, Anda dapat melihat subbagian Distribusi kelas penyimpanan dan Distribusi wilayah.
5. Untuk melihat wawasan yang lebih terperinci untuk setiap bucket yang Anda lihat di bagan Tren untuk tanggal agar dapat melakukan analisis yang lebih mendalam, lakukan hal-hal berikut:

   a. Pilih satu titik pada bagan. Sebuah kotak berisi pilihan untuk wawasan yang lebih terperinci akan ditampilkan.
   b. Pilih dimensi yang akan diterapkan ke data Anda untuk analisis yang lebih mendalam: Akun, Wilayah AWS, Kelas penyimpanan, atau Bucket. Lalu, pilih Terapkan.
6. Di bagian Analisis gelembung berdasarkan bucket untuk tanggal, pilih jumlah bucket yang mengaktifkan Penentuan Versi, Bucket, dan metrik Bucket aktif.

   Analisis gelembung berdasarkan bucket untuk tanggal diperbarui dengan tujuan menampilkan data untuk metrik yang dipilih. Anda dapat menggunakan data ini untuk melihat bucket mana yang mengaktifkan Penentuan Versi S3 di dalam konteks jumlah total bucket Anda. Di bagian Analisis gelembung berdasarkan bucket untuk tanggal, Anda dapat memplot bucket pada beberapa dimensi dengan menggunakan tiga metrik mana pun untuk mewakili sumbu X, sumbu Y, dan Ukuran gelembung.

Langkah 2: Aktifkan Penentuan Versi S3

Setelah mengidentifikasi bucket yang mengaktifkan Penentuan Versi S3, Anda dapat mengidentifikasi bucket yang belum pernah mengaktifkan Penentuan Versi S3 atau yang mengangguhkan Penentuan Versi. Kemudian, Anda dapat mengaktifkan Penentuan Versi untuk bucket ini secara opsional di konsol S3. Untuk informasi selengkapnya, lihat Mengaktifkan Penentuan Versi pada bucket.

Identifikasi permintaan yang menggunakan AWS Signature Version 2 (SigV2)

Anda dapat menggunakan metrik Semua permintaan tanda tangan yang tidak didukung untuk mengidentifikasi permintaan yang menggunakan AWS Signature Version 2 (SigV2). Data ini dapat membantu mengidentifikasi aplikasi tertentu yang menggunakan SigV2. Anda kemudian dapat memigrasikan aplikasi ini ke AWS Signature Version 4 (SigV4).

SigV4 adalah metode penandatanganan yang direkomendasikan untuk semua aplikasi S3 yang baru. SiGv4 memberikan keamanan yang lebih baik dan didukung di semua. Wilayah AWS Untuk informasi selengkapnya, lihat Pembaruan Amazon S3-Periode penghentian SigV2 yang diperpanjang & dimodifikasi.

Prasyarat

Untuk melihat Semua permintaan tanda tangan yang tidak didukung di dasbor Lensa Penyimpanan S3, Anda harus mengaktifkan Metrik dan rekomendasi lanjutan Lensa Penyimpanan S3 dan pilih Metrik perlindungan data lanjutan. Untuk informasi selengkapnya, lihat Menggunakan konsol S3.

Langkah 1: Periksa tren penandatanganan SigV2 berdasarkan Akun AWS, Wilayah, dan ember

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, pilih Lensa Penyimpanan, Dasbor.
3. Di daftar Dasbor, pilih nama dasbor yang ingin dilihat.
4. Untuk mengidentifikasi bucket, akun, dan Wilayah tertentu dengan permintaan yang menggunakan SigV2:

    a. Di bawah Ikhtisar N Teratas untuk tanggal, di N Teratas, masukkan jumlah bucket yang ingin Anda lihat datanya.

    b. Untuk Metrik, pilih Semua permintaan tanda tangan yang tidak didukung dari kategori Perlindungan data.

    Ikhtisar N Teratas untuk pembaruan tanggal guna menampilkan data permintaan SigV2 berdasarkan akun, Wilayah AWS, dan bucket. Ikhtisar N Teratas untuk bagian tanggal juga menunjukkan persentase perubahan dari hari atau minggu sebelumnya dan garis percikan untuk memvisualisasikan tren. Tren ini adalah tren 14 hari untuk metrik gratis dan tren 30 hari untuk metrik dan rekomendasi lanjutan.

    > **Note**
    >
    > Dengan metrik dan rekomendasi lanjutan Lensa Penyimpanan S3, metrik tersedia untuk kueri selama 15 bulan. Untuk informasi selengkapnya, lihat Pilihan metrik.

Langkah 2: Identifikasi bucket yang diakses oleh aplikasi melalui permintaan SigV2

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, pilih Lensa Penyimpanan, Dasbor.
3. Di daftar Dasbor, pilih nama dasbor yang ingin dilihat.
4. Di dasbor Lensa Penyimpanan, pilih tab Bucket.
5. Gulir ke bawah ke bagian Bucket. Di bawah Kategori metrik, pilih Perlindungan data. Kemudian hapus Ringkasan.

   Daftar Bucket diperbarui guna menampilkan semua metrik Perlindungan data yang tersedia untuk bucket yang ditampilkan.
6. Untuk memfilter daftar Bucket agar menampilkan hanya metrik perlindungan data tertentu, pilih ikon preferensi
   ( ).
7. Hapus tombol untuk semua metrik perlindungan data hingga hanya metrik berikut yang tetap dipilih:

   - Semua permintaan tanda tangan yang tidak didukung
   - % semua permintaan tanda tangan yang tidak didukung
8. (Opsional) Di bawah Ukuran halaman, pilih jumlah bucket yang akan ditampilkan di daftar.
9. Pilih Konfirmasi.

   Daftar Bucket diperbarui guna menampilkan metrik tingkat bucket untuk permintaan SigV2. Anda dapat menggunakan data ini untuk mengidentifikasi bucket tertentu yang memiliki permintaan SigV2. Kemudian, Anda dapat menggunakan informasi ini untuk memigrasikan aplikasi ke SigV4. Untuk informasi selengkapnya, lihat Mengautentikasi Permintaan (Versi AWS Tanda Tangan 4) di Referensi API Amazon Simple Storage Service.

Hitung jumlah total aturan replikasi untuk setiap bucket

Replikasi S3 memungkinkan penyalinan objek secara otomatis dan asinkron di seluruh bucket Amazon S3. Bucket yang dikonfigurasi untuk replikasi objek dapat dimiliki oleh Akun AWS yang sama atau oleh akun yang berbeda. Untuk informasi selengkapnya, lihat Mereplikasi objek di dalam dan di seluruh Wilayah.

Anda dapat menggunakan metrik jumlah aturan replikasi Lensa Penyimpanan S3 untuk mendapatkan informasi terperinci per bucket tentang bucket yang dikonfigurasi untuk replikasi. Informasi ini mencakup aturan replikasi di dalam dan di seluruh bucket dan Wilayah.

Prasyarat

Untuk melihat metrik jumlah aturan replikasi di dasbor Lensa Penyimpanan S3, Anda harus mengaktifkan Metrik dan rekomendasi lanjutan Lensa Penyimpanan S3, lalu pilih Metrik perlindungan data lanjutan. Untuk informasi selengkapnya, lihat Menggunakan konsol S3.

Langkah 1: Hitung jumlah total aturan replikasi untuk setiap bucket

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, pilih Lensa Penyimpanan, Dasbor.
3. Di daftar Dasbor, pilih nama dasbor yang ingin dilihat.
4. Di dasbor Lensa Penyimpanan, pilih tab Bucket.
5. Gulir ke bawah ke bagian Bucket. Di bawah Kategori metrik, pilih Perlindungan data. Kemudian hapus Ringkasan.
6. Untuk memfilter daftar Bucket agar menampilkan hanya metrik jumlah aturan replikasi, pilih ikon preferensi
   ( ).
7. Hapus tombol untuk semua metrik perlindungan data hingga hanya metrik jumlah aturan replikasi yang tetap dipilih:
   - Jumlah aturan Replikasi Wilayah Sama
   - Jumlah aturan Replikasi Lintas Wilayah
   - Jumlah aturan replikasi akun sama
   - Jumlah aturan replikasi lintas akun
   - Jumlah total aturan replikasi
8. (Opsional) Di bawah Ukuran halaman, pilih jumlah bucket yang akan ditampilkan di daftar.
9. Pilih Konfirmasi.

Langkah 2: Menambahkan aturan replikasi

Setelah memiliki jumlah aturan replikasi per bucket, Anda dapat membuat aturan replikasi tambahan secara opsional. Untuk informasi selengkapnya, lihat Contoh untuk mengonfigurasi replikasi langsung.

Identifikasi persentase byte Kunci Objek

Dengan S3 Object Lock, Anda dapat menyimpan objek dengan menggunakan model write-once-read-many (WORM). Anda dapat menggunakan Kunci Objek untuk membantu mencegah objek terhapus atau tertimpa selama jangka waktu tertentu atau tanpa batas waktu. Anda dapat mengaktifkan Kunci Objek hanya ketika Anda membuat bucket dan juga mengaktifkan Penentuan Versi S3. Namun, Anda dapat mengedit periode retensi untuk versi objek individual atau menerapkan penahanan hukum untuk bucket yang mengaktifkan Kunci Objek. Untuk informasi selengkapnya, lihat Mengunci objek dengan Object Lock.

Anda dapat menggunakan metrik Kunci Objek di Lensa Penyimpanan S3 untuk melihat metrik % byte Kunci Objek untuk akun atau organisasi Anda. Anda dapat menggunakan informasi ini untuk mengidentifikasi bucket di akun atau organisasi Anda yang tidak mengikuti praktik terbaik perlindungan data.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, pilih Lensa Penyimpanan, Dasbor.
3. Di daftar Dasbor, pilih nama dasbor yang ingin dilihat.
4. Di bagian Snapshot, di bawah Kategori metrik, pilih Perlindungan data.

   Bagian Snapshot diperbarui guna menampilkan metrik perlindungan data, termasuk metrik % byte Kunci Objek. Anda dapat melihat persentase keseluruhan byte Kunci Objek untuk akun atau organisasi Anda.
5. Untuk melihat % byte Kunci Objek per bucket, gulir ke bawah ke bagian Ringkasan N teratas.

   Untuk mendapatkan data tingkat objek untuk Kunci Objek, Anda juga dapat menggunakan jumlah objek Kunci Objek dan metrik % objek Kunci Objek.
6. Untuk Metrik, pilih% byte Kunci Objek dari kategori Perlindungan data.

   Secara default, ikhtisar N Teratas untuk bagian tanggal menampilkan metrik untuk 3 bucket teratas. Di bidang N teratas, Anda dapat menambah jumlah bucket. Ikhtisar N Teratas untuk

bagian tanggal juga menunjukkan persentase perubahan dari hari atau minggu sebelumnya dan garis percikan untuk memvisualisasikan tren. Tren ini adalah tren 14 hari untuk metrik gratis dan tren 30 hari untuk metrik dan rekomendasi lanjutan.

> **Note**
>
> Dengan metrik dan rekomendasi lanjutan Lensa Penyimpanan S3, metrik tersedia untuk kueri selama 15 bulan. Untuk informasi selengkapnya, lihat Pilihan metrik.

7. Tinjau data berikut untuk% byte Kunci Objek:

   - Akun nomor teratas-Lihat akun mana yang memiliki % byte Kunci Objek tertinggi dan terendah.
   - Wilayah nomor teratas-Lihat rincian % byte Kunci Objek berdasarkan Wilayah.
   - Bucket nomor teratas-Lihat bucket mana yang memiliki % byte Kunci Objek tertinggi dan terendah.

Menggunakan Lensa Penyimpanan S3 untuk mengaudit pengaturan Kepemilikan Objek

Amazon S3 Object Ownership adalah setelan tingkat ember S3 yang dapat Anda gunakan untuk menonaktifkan daftar kontrol akses (ACLs) dan mengontrol kepemilikan objek di bucket Anda. Jika Anda menyetel Kepemilikan Objek ke pemilik bucket yang diberlakukan, Anda dapat menonaktifkan daftar kontrol akses (ACLs) dan mengambil kepemilikan setiap objek di bucket Anda. Pendekatan ini menyederhanakan pengelolaan akses untuk data yang disimpan di Amazon S3.

Secara default, ketika orang lain Akun AWS mengunggah objek ke bucket S3 Anda, akun tersebut (penulis objek) memiliki objek tersebut, memiliki akses ke objek tersebut, dan dapat memberi pengguna lain akses ke objek tersebut. ACLs Gunakan Kepemilikan Objek untuk mengubah perilaku default ini.

Mayoritas kasus penggunaan modern di Amazon S3 tidak lagi memerlukan penggunaan. ACLs Oleh karena itu, kami menyarankan Anda menonaktifkan ACLs, kecuali dalam keadaan yang tidak biasa di mana Anda harus mengontrol akses untuk setiap objek secara individual. Dengan menyetel Kepemilikan Objek ke pemilik bucket yang diberlakukan, Anda dapat menonaktifkan ACLs dan mengandalkan kebijakan untuk kontrol akses. Untuk informasi selengkapnya, lihat Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.

Dengan metrik manajemen akses S3 Storage Lens, Anda dapat mengidentifikasi bucket yang tidak dinonaktifkan. ACLs Setelah mengidentifikasi bucket ini, Anda dapat memigrasikan izin ACL ke kebijakan dan menonaktifkan ACLs bucket tersebut.

Topik

- Langkah 1: Identifikasi tren umum untuk setelan Kepemilikan Objek
- Langkah 2: Identifikasi tren tingkat bucket untuk setelan Kepemilikan Objek
- Langkah 3: Perbarui setelan Kepemilikan Objek Anda ke pemilik bucket yang diberlakukan untuk menonaktifkan ACLs

Langkah 1: Identifikasi tren umum untuk setelan Kepemilikan Objek

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, pilih Lensa Penyimpanan, Dasbor.
3. Di daftar Dasbor, pilih nama dasbor yang ingin dilihat.
4. Di bagian Snapshot untuk tanggal, di bawah kategori Metrik, pilih Manajemen akses.

   Bagian Snapshot untuk tanggal diperbarui untuk menampilkan metrik % Kepemilikan Objek pemilik bucket yang diberlakukan. Anda dapat melihat persentase keseluruhan bucket di akun atau organisasi yang menggunakan pengaturan yang diberlakukan pemilik bucket untuk menonaktifkan Kepemilikan Objek. ACLs

Langkah 2: Identifikasi tren tingkat bucket untuk setelan Kepemilikan Objek

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, pilih Lensa Penyimpanan, Dasbor.
3. Di daftar Dasbor, pilih nama dasbor yang ingin dilihat.
4. Untuk melihat metrik tingkat bucket yang lebih detail, pilih tab Bucket.
5. Di bagian Distribusi berdasarkan bucket untuk tanggal, pilih metrik % Kepemilikan Objek pemilik bucket yang diberlakukan.

Bagan diperbarui untuk menampilkan rincian per bucket untuk% Kepemilikan Objek pemilik bucket yang diberlakukan. Anda dapat melihat bucket mana yang menggunakan setelan yang diberlakukan pemilik bucket untuk menonaktifkan Kepemilikan Objek. ACLs

6. Untuk melihat konteks setelan pemilik bucket yang diberlakukan, gulir ke bawah ke bagian Bucket. Untuk Kategori metrik, pilih Manajemen akses. Kemudian hapus Ringkasan.

   Daftar Bucket menampilkan data untuk ketiga setelan Kepemilikan Objek: pemilik bucket yang diberlakukan, pemilik bucket yang dipilih, dan penulis objek.

7. Untuk memfilter daftar Bucket agar dapat menampilkan metrik hanya untuk setelan Kepemilikan Objek tertentu, pilih ikon preferensi ( ).

8. Hapus metrik yang tidak ingin dilihat.
9. (Opsional) Di bawah Ukuran halaman, pilih jumlah bucket yang akan ditampilkan di daftar.
10. Pilih Konfirmasi.

Langkah 3: Perbarui setelan Kepemilikan Objek Anda ke pemilik bucket yang diberlakukan untuk menonaktifkan ACLs

Setelah mengidentifikasi bucket yang menggunakan setelan penulis objek dan pemilik bucket yang dipilih untuk Kepemilikan Objek, Anda dapat memigrasikan izin ACL ke kebijakan bucket. Setelah selesai memigrasikan izin ACL, Anda dapat memperbarui setelan Kepemilikan Objek ke pemilik bucket yang diberlakukan untuk menonaktifkan. ACLs Untuk informasi selengkapnya, lihat Prasyarat untuk menonaktifkan ACLs.

Menggunakan metrik Lensa Penyimpanan S3 untuk meningkatkan kinerja

Jika Anda mengaktifkan Metrik lanjutan Lensa Penyimpanan S3, Anda dapat menggunakan metrik kode status terperinci untuk mendapatkan jumlah permintaan yang berhasil atau gagal. Anda dapat menggunakan informasi ini untuk memecahkan masalah akses atau kinerja. Metrik kode status terperinci menunjukkan jumlah kode status HTTP, seperti 403 Forbidden dan 503 Service Unavailable. Anda dapat memeriksa tren keseluruhan untuk metrik kode status terperinci di seluruh bucket, akun, dan organisasi S3. Kemudian, Anda dapat menelusuri metrik tingkat bucket untuk mengidentifikasi beban kerja yang saat ini mengakses bucket tersebut dan menyebabkan kesalahan.

Misalnya, Anda dapat melihat metrik jumlah kesalahan 403 Forbidden untuk mengidentifikasi beban kerja yang mengakses bucket dengan izin yang tidak sesuai. Setelah mengidentifikasi beban kerja

ini, Anda dapat mempelajari lebih dalam di luar Lensa Penyimpanan S3 untuk memecahkan masalah kesalahan 403 Forbidden.

Contoh ini menunjukkan cara melakukan analisis tren untuk kesalahan 403 Forbidden dengan menggunakan jumlah kesalahan 403 Forbidden dan metrik % kesalahan 403 Forbidden. Anda dapat menggunakan metrik ini untuk mengidentifikasi beban kerja yang mengakses bucket dengan izin yang tidak sesuai. Anda dapat melakukan analisis tren serupa untuk salah satu Metrik kode status terperinci lainnya. Untuk informasi selengkapnya, lihat Glosarium metrik Lensa Penyimpanan Amazon S3.

Prasyarat

Untuk melihat Metrik kode status terperinci di dasbor Lensa Penyimpanan S3, Anda harus mengaktifkan Metrik dan rekomendasi lanjutan Lensa Penyimpanan S3, lalu pilih Metrik kode status terperinci. Untuk informasi selengkapnya, lihat Menggunakan konsol S3.

Topik

- Langkah 1: Lakukan analisis tren untuk sebuah kode status HTTP individual
- Langkah 2: Analisis jumlah kesalahan berdasarkan bucket
- Langkah 3: Memecahkan masalah kesalahan

Langkah 1: Lakukan analisis tren untuk sebuah kode status HTTP individual

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, pilih Lensa Penyimpanan, Dasbor.
3. Di daftar Dasbor, pilih nama dasbor yang ingin dilihat.
4. Di bagian Tren dan distribusi, untuk Metrik primer, pilih Jumlah kesalahan 403 Forbidden dari kategori Kode status terperinci. Untuk Metrik sekunder, pilih% kesalahan 403 Forbidden.
5. Gulir ke bawah ke bagian Ikhtisar N teratas untuk tanggal. Untuk Metrik, pilih Jumlah kesalahan 403 Forbidden atau % kesalahan 403 Forbidden dari kategori Kode status terperinci.

   Ikhtisar N Teratas untuk pembaruan bagian tanggal untuk menampilkan 403 jumlah kesalahan Terlarang teratas berdasarkan akun, Wilayah AWS, dan bucket.

Langkah 2: Analisis jumlah kesalahan berdasarkan bucket

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, pilih Lensa Penyimpanan, Dasbor.
3. Di daftar Dasbor, pilih nama dasbor yang ingin dilihat.
4. Di dasbor Lensa Penyimpanan, pilih tab Bucket.
5. Gulir ke bawah ke bagian Bucket. Untuk Kategori metrik, pilih metrik Kode status terperinci. Kemudian hapus Ringkasan.

   Daftar Bucket diperbarui guna menampilkan semua metrik kode status terperinci yang tersedia. Anda dapat menggunakan informasi ini untuk melihat bucket mana yang memiliki kode status HTTP tertentu dengan proporsi besar dan kode status mana yang umum di seluruh bucket.
6. Untuk memfilter daftar Bucket agar hanya menampilkan metrik kode status terperinci tertentu, pilih ikon preferensi
   ( ).
7. Hapus tombol untuk setiap metrik kode status terperinci yang tidak ingin dilihat di daftar Bucket.
8. (Opsional) Di bawah Ukuran halaman, pilih jumlah bucket yang akan ditampilkan di daftar.
9. Pilih Konfirmasi.

   Daftar Bucket akan menampilkan metrik jumlah kesalahan untuk jumlah bucket yang sudah ditentukan. Anda dapat menggunakan informasi ini untuk mengidentifikasi bucket tertentu yang mengalami banyak kesalahan dan memecahkan masalah kesalahan dengan menggunakan bucket.

Langkah 3: Memecahkan masalah kesalahan

Setelah mengidentifikasi bucket dengan proporsi kode status HTTP spesifik yang tinggi, Anda dapat memecahkan masalah kesalahan ini. Untuk informasi selengkapnya, lihat berikut ini:

- Mengapa saya mendapatkan kesalahan 403 Forbidden ketika mencoba mengunggah file di Amazon S3?
- Mengapa saya mendapatkan kesalahan 403 Forbidden ketika mencoba mengubah kebijakan bucket di Amazon S3?
- Bagaimana cara memecahkan masalah kesalahan 403 Forbidden dari bucket Amazon S3 yang semua sumber dayanya berasal dari yang sama Akun AWS?

- Bagaimana cara memecahkan masalah kesalahan HTTP 500 atau HTTP 503 dari Amazon S3?

Menggunakan Lensa Penyimpanan Amazon S3 dengan AWS Organizations

Lensa Penyimpanan Amazon S3 adalah fitur analisis penyimpanan cloud yang dapat Anda gunakan untuk mendapatkan visibilitas seluruh organisasi ke dalam penggunaan dan aktivitas penyimpanan objek. Anda dapat menggunakan metrik Lensa Penyimpanan S3 untuk menghasilkan wawasan ringkasan, seperti mencari tahu berapa banyak penyimpanan yang Anda miliki di seluruh organisasi atau bucket dan prefiks yang paling cepat berkembang. Anda juga dapat menggunakan Amazon S3 Storage Lens untuk mengumpulkan metrik penyimpanan dan data penggunaan untuk semua Akun AWS yang merupakan bagian dari hierarki Anda. AWS Organizations Untuk melakukan ini, Anda harus menggunakan AWS Organizations, dan Anda harus mengaktifkan akses tepercaya S3 Storage Lens dengan menggunakan akun AWS Organizations manajemen Anda.

Setelah mengaktifkan akses tepercaya, tambahkan akses administrator yang didelegasikan ke akun di organisasi Anda. Akun administrator yang didelegasikan digunakan untuk membuat konfigurasi dan dasbor Lensa Penyimpanan S3 yang mengumpulkan metrik penyimpanan dan data pengguna di seluruh organisasi. Untuk informasi lebih lanjut tentang mengaktifkan akses tepercaya, lihat Lensa Penyimpanan Amazon S3 dan AWS Organizations di Panduan Pengguna AWS Organizations .

Topik

- Mengaktifkan akses tepercaya untuk Lensa Penyimpanan S3
- Menonaktifkan akses tepercaya untuk Lensa Penyimpanan S3
- Mendaftarkan administrator yang didelegasikan untuk Lensa Penyimpanan S3
- Membatalkan pendaftaran administrator yang didelegasikan untuk Lensa Penyimpanan S3

Mengaktifkan akses tepercaya untuk Lensa Penyimpanan S3

Dengan mengaktifkan akses tepercaya, Anda mengizinkan Amazon S3 Storage Lens mengakses hierarki, keanggotaan, dan struktur AWS Organizations AWS Organizations Anda melalui operasi API. Lensa Penyimpanan S3 kemudian menjadi layanan tepercaya untuk seluruh struktur organisasi.

Setiap kali konfigurasi dasbor dibuat, Lensa Penyimpanan S3 menciptakan peran terkait layanan di manajemen organisasi atau akun administrator yang didelegasikan. Peran terkait layanan memberikan izin S3 Storage Lens untuk melakukan tindakan berikut:

- Jelaskan organisasi
- Cantumkan akun
- Verifikasi daftar Layanan AWS akses untuk organisasi
- Dapatkan administrator yang didelegasikan untuk organisasi

S3 Storage Lens kemudian dapat memastikan bahwa ia memiliki akses untuk mengumpulkan metrik lintas akun untuk akun di organisasi Anda. Untuk informasi selengkapnya, lihat Menggunakan peran terkait layanan untuk Lensa Penyimpanan Amazon S3.

Setelah mengaktifkan akses tepercaya, Anda dapat menetapkan akses administrator delegasi ke akun di organisasi. Ketika akun ditandai sebagai administrator yang didelegasikan untuk suatu layanan, akun tersebut akan menerima otorisasi untuk mengakses semua operasi API organisasi hanya-baca. Akses ini memberikan visibilitas administrator yang didelegasikan kepada anggota dan struktur organisasi sehingga mereka juga dapat membuat dasbor Lensa Penyimpanan S3.

> **Note**
>
> - Akses tepercaya hanya dapat diaktifkan oleh akun manajemen.
> - Hanya akun manajemen dan administrator yang didelegasikan yang dapat membuat dasbor atau konfigurasi S3 Storage Lens untuk organisasi Anda.

Menggunakan konsol S3

Untuk mengaktifkan S3 Storage Lens memiliki akses AWS Organizations tepercaya

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, navigasikan ke Storage Lens.
3. Pilih AWS Organizations pengaturan. AWS Organizations Akses untuk menampilkan halaman Storage Lens.
4. Di bawah akses AWS Organizations tepercaya, pilih Edit.

   Halaman AWS Organizations akses ditampilkan.
5. Pilih Aktifkan untuk mengaktifkan akses tepercaya untuk dasbor Lensa Penyimpanan S3 Anda.
6. Pilih Simpan perubahan.

Menggunakan AWS CLI

Example

Contoh berikut menunjukkan cara mengaktifkan akses AWS Organizations tepercaya untuk S3 Storage Lens di AWS CLI.

```
aws organizations enable-aws-service-access --service-principal storage-
lens.s3.amazonaws.com
```

Menggunakan AWS SDK for Java

Example — Aktifkan akses AWS Organizations tepercaya untuk S3 Storage Lens menggunakan SDK for Java

Contoh berikut menunjukkan cara mengaktifkan akses tepercaya untuk S3 Storage Lens di SDK for Java. Untuk menggunakan contoh ini, ganti *user input placeholders*dengan informasi Anda sendiri.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.organizations.AWSOrganizations;
import com.amazonaws.services.organizations.AWSOrganizationsClient;
import com.amazonaws.services.organizations.model.EnableAWSServiceAccessRequest;

public class EnableOrganizationsTrustedAccess {
 private static final String S3_STORAGE_LENS_SERVICE_PRINCIPAL = "storage-
lens.s3.amazonaws.com";

 public static void main(String[] args) {
  try {
            AWSOrganizations organizationsClient = AWSOrganizationsClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(Regions.US_EAST_1)
                .build();

            organizationsClient.enableAWSServiceAccess(new
 EnableAWSServiceAccessRequest()
                .withServicePrincipal(S3_STORAGE_LENS_SERVICE_PRINCIPAL));
        } catch (AmazonServiceException e) {
```

```
            // The call was transmitted successfully, but AWS Organizations couldn't
 process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // AWS Organizations couldn't be contacted for a response, or the client
            // couldn't parse the response from AWS Organizations.
            e.printStackTrace();
        }
 }
}
```

Menonaktifkan akses tepercaya untuk Lensa Penyimpanan S3

Menghapus akun sebagai administrator yang didelegasikan atau menonaktifkan akses tepercaya membatasi metrik dasbor Lensa Penyimpanan S3 pemilik akun agar hanya berfungsi pada tingkat akun. Setiap pemegang akun kemudian hanya dapat melihat manfaat S3 Storage Lens melalui cakupan terbatas akun mereka, dan bukan seluruh organisasi mereka.

Saat Anda menonaktifkan akses tepercaya di Lensa Penyimpanan S3, dasbor apa pun yang memerlukan akses tepercaya tidak lagi diperbarui. Dasbor organisasi apa pun yang dibuat juga tidak lagi diperbarui. Sebagai gantinya, Anda hanya dapat menanyakan data historis untuk dasbor Lensa Penyimpanan S3, sementara data masih tersedia.

> **Note**
>
> - Menonaktifkan akses tepercaya untuk Lensa Penyimpanan S3 juga secara otomatis akan menghentikan semua dasbor tingkat organisasi untuk mengumpulkan dan menggabungkan metrik penyimpanan. Ini karena S3 Storage Lens tidak lagi memiliki akses tepercaya ke akun organisasi.
> - Akun administrator manajemen dan delegasi Anda masih dapat melihat data historis untuk setiap dasbor yang dinonaktifkan. Mereka juga dapat menanyakan data historis ini saat masih tersedia.

Menggunakan konsol S3

Untuk menonaktifkan akses tepercaya untuk Lensa Penyimpanan S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, navigasikan ke Storage Lens.
3. Pilih AWS Organizations pengaturan. AWS Organizations Akses untuk menampilkan halaman Storage Lens.
4. Di bawah akses AWS Organizations tepercaya, pilih Edit.

   Halaman AWS Organizations akses ditampilkan.
5. Pilih Nonaktifkan untuk menonaktifkan akses tepercaya untuk dasbor Lensa Penyimpanan S3 Anda.
6. Pilih Simpan perubahan.

Menggunakan AWS CLI

Example

Contoh berikut menonaktifkan akses tepercaya untuk S3 Storage Lens menggunakan. AWS CLI

```
aws organizations disable-aws-service-access --service-principal storage-
lens.s3.amazonaws.com
```

Menggunakan AWS SDK for Java

Example - Nonaktifkan akses AWS Organizations tepercaya untuk Lensa Penyimpanan S3

Contoh berikut menunjukkan cara menonaktifkan akses AWS Organizations tepercaya untuk S3 Storage Lens di SDK for Java. Untuk menggunakan contoh ini, ganti *`user input placeholders`*dengan informasi Anda sendiri.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.organizations.AWSOrganizations;
```

```
import com.amazonaws.services.organizations.AWSOrganizationsClient;
import com.amazonaws.services.organizations.model.DisableAWSServiceAccessRequest;

public class DisableOrganizationsTrustedAccess {
 private static final String S3_STORAGE_LENS_SERVICE_PRINCIPAL = "storage-
lens.s3.amazonaws.com";

 public static void main(String[] args) {
  try {
            AWSOrganizations organizationsClient = AWSOrganizationsClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(Regions.US_EAST_1)
                .build();

            // Make sure to remove any existing delegated administrator for S3 Storage
 Lens
            // before disabling access; otherwise, the request will fail.
            organizationsClient.disableAWSServiceAccess(new
 DisableAWSServiceAccessRequest()
                .withServicePrincipal(S3_STORAGE_LENS_SERVICE_PRINCIPAL));
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but AWS Organizations couldn't
 process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // AWS Organizations couldn't be contacted for a response, or the client
            // couldn't parse the response from AWS Organizations.
            e.printStackTrace();
        }
 }
}
```

Mendaftarkan administrator yang didelegasikan untuk Lensa Penyimpanan S3

Anda dapat membuat dasbor tingkat organisasi dengan menggunakan akun manajemen organisasi atau akun administrator yang didelegasikan. Akun administrator yang didelegasikan mengizinkan akun lain selain akun manajemen untuk membuat dasbor tingkat organisasi. Hanya akun manajemen organisasi yang dapat mendaftarkan dan membatalkan pendaftaran akun lain sebagai administrator yang didelegasikan untuk organisasi.

Setelah mengaktifkan akses tepercaya, Anda dapat mendaftarkan akses administrator delegasi ke akun di organisasi menggunakan AWS Organizations REST API AWS CLI, atau SDKs dari akun

manajemen. (Untuk informasi selengkapnya, lihat RegisterDelegatedAdministratordi Referensi AWS Organizations API.) Ketika akun terdaftar sebagai administrator yang didelegasikan, akun akan menerima otorisasi untuk mengakses semua operasi API hanya-baca AWS Organizations . Hal ini memberikan visibilitas kepada anggota dan struktur organisasi Anda sehingga mereka dapat membuat dasbor Lensa Penyimpanan S3 atas nama Anda.

> **Note**
>
> Sebelum Anda dapat menunjuk administrator yang didelegasikan dengan menggunakan AWS Organizations REST API, AWS CLI, atau SDKs, Anda harus memanggil operasi. EnableAWSOrganizationsAccess

Menggunakan konsol S3

Untuk mendaftarkan administrator yang didelegasikan untuk Lensa Penyimpanan S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, navigasikan ke Storage Lens.
3. Pilih AWS Organizations pengaturan.
4. Di bawah Administrator yang didelegasikan, pilih Daftar akun.
5. Tambahkan Akun AWS ID untuk mendaftarkan akun sebagai administrator yang didelegasikan. Administrator yang didelegasikan dapat membuat dasbor tingkat organisasi untuk semua akun dan penyimpanan di organisasi Anda.
6. Pilih Daftar akun.

Menggunakan AWS CLI

Example

Contoh berikut menunjukkan cara mendaftarkan administrator yang didelegasikan Organizations untuk S3 Storage Lens menggunakan. AWS CLI Untuk menggunakan contoh ini, ganti *user input placeholders*dengan informasi Anda sendiri.

```
aws organizations register-delegated-administrator --service-principal storage-lens.s3.amazonaws.com --account-id 111122223333
```

Menggunakan AWS SDK for Java

Example — Daftarkan administrator yang didelegasikan Organizations untuk S3 Storage Lens

Contoh berikut menunjukkan cara mendaftarkan administrator yang AWS Organizations didelegasikan untuk S3 Storage Lens di SDK for Java. Untuk menggunakan contoh ini, ganti *user input placeholders*dengan informasi Anda sendiri.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.organizations.AWSOrganizations;
import com.amazonaws.services.organizations.AWSOrganizationsClient;
import
 com.amazonaws.services.organizations.model.RegisterDelegatedAdministratorRequest;

public class RegisterOrganizationsDelegatedAdministrator {
 private static final String S3_STORAGE_LENS_SERVICE_PRINCIPAL = "storage-
lens.s3.amazonaws.com";

 public static void main(String[] args) {
  try {
            String delegatedAdminAccountId = "111122223333"; // Account Id for the
 delegated administrator.
            AWSOrganizations organizationsClient = AWSOrganizationsClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(Regions.US_EAST_1)
                .build();

            organizationsClient.registerDelegatedAdministrator(new
 RegisterDelegatedAdministratorRequest()
                .withAccountId(delegatedAdminAccountId)
                .withServicePrincipal(S3_STORAGE_LENS_SERVICE_PRINCIPAL));
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but AWS Organizations couldn't
 process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // AWS Organizations couldn't be contacted for a response, or the client
            // couldn't parse the response from AWS Organizations.
            e.printStackTrace();
```

```
        }
 }
}
```

Membatalkan pendaftaran administrator yang didelegasikan untuk Lensa Penyimpanan S3

Setelah mengaktifkan akses tepercaya, Anda juga dapat membatalkan pendaftaran akses administrator delegasi ke akun di organisasi Anda. Akun administrator yang didelegasikan memungkinkan akun lain selain akun manajemen Anda untuk membuat dasbor tingkat organisasi. Hanya akun manajemen organisasi yang dapat membatalkan pendaftaran akun sebagai administrator yang didelegasikan untuk organisasi.

Anda dapat membatalkan pendaftaran administrator yang didelegasikan dengan menggunakan AWS Organizations AWS Management Console, REST API AWS CLI, atau AWS SDKS dari akun manajemen. Untuk informasi selengkapnya, lihat DeregisterDelegatedAdministrator di dalam Referensi API AWS Organizations .

Ketika akun dideregistrasi sebagai administrator yang didelegasikan, akun kehilangan akses ke hal-hal berikut:

- Semua operasi AWS Organizations API hanya-baca yang memberikan visibilitas ke anggota dan struktur organisasi Anda.
- Semua dasbor tingkat organisasi yang dibuat oleh administrator yang didelegasikan. Membatalkan pendaftaran administrator yang didelegasikan juga secara otomatis akan menghentikan semua dasbor tingkat organisasi yang dibuat oleh administrator yang didelegasikan tersebut untuk menggabungkan metrik penyimpanan baru.

> **Note**
>
> Administrator delegasi yang dideregistrasi masih dapat melihat data historis untuk dasbor yang dinonaktifkan yang mereka buat jika data masih tersedia untuk kueri.

Menggunakan konsol S3

Untuk membatalkan pendaftaran administrator yang didelegasikan untuk S3 Storage Lens

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, navigasikan ke Storage Lens.
3. Pilih AWS Organizations pengaturan.
4. Di bawah Administrator yang didelegasikan, pilih akun yang ingin Anda deregister.
5. Pilih De-register akun. Akun yang dideregistrasi bukan lagi administrator yang didelegasikan dan sekarang tidak dapat membuat dasbor tingkat organisasi untuk semua akun dan penyimpanan di organisasi Anda.
6. Pilih Daftar akun.

Menggunakan AWS CLI

Example

Contoh berikut menunjukkan cara membatalkan pendaftaran administrator yang didelegasikan Organizations untuk S3 Storage Lens menggunakan. AWS CLI Untuk menggunakan contoh ini, ganti *111122223333* dengan Akun AWS ID Anda sendiri.

```
aws organizations deregister-delegated-administrator --service-principal storage-lens.s3.amazonaws.com --account-id 111122223333
```

Menggunakan AWS SDK for Java

Example — Administrator delegasi Organisasi Deregister untuk S3 Storage Lens

Contoh berikut menunjukkan cara membatalkan pendaftaran administrator yang didelegasikan Organizations untuk S3 Storage Lens menggunakan SDK for Java. Untuk menggunakan contoh ini, ganti *user input placeholders*dengan informasi Anda sendiri.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.organizations.AWSOrganizations;
```

```
import com.amazonaws.services.organizations.AWSOrganizationsClient;
import
 com.amazonaws.services.organizations.model.DeregisterDelegatedAdministratorRequest;

public class DeregisterOrganizationsDelegatedAdministrator {
 private static final String S3_STORAGE_LENS_SERVICE_PRINCIPAL = "storage-
lens.s3.amazonaws.com";

 public static void main(String[] args) {
  try {
            String delegatedAdminAccountId = "111122223333"; // Account Id for the
 delegated administrator.
            AWSOrganizations organizationsClient = AWSOrganizationsClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(Regions.US_EAST_1)
                .build();

            organizationsClient.deregisterDelegatedAdministrator(new
 DeregisterDelegatedAdministratorRequest()
                .withAccountId(delegatedAdminAccountId)
                .withServicePrincipal(S3_STORAGE_LENS_SERVICE_PRINCIPAL));
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but AWS Organizations couldn't
 process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // AWS Organizations couldn't be contacted for a response, or the client
            // couldn't parse the response from AWS Organizations.
            e.printStackTrace();
        }
 }
}
```

Bekerja dengan grup Lensa Penyimpanan S3 untuk memfilter dan mengumpulkan metrik

Grup Lensa Penyimpanan Amazon S3 menggabungkan metrik dengan menggunakan filter khusus berdasarkan metadata objek. Grup Lensa Penyimpanan membantu menelusuri karakteristik data, seperti distribusi objek berdasarkan usia, jenis file yang paling umum, dan banyak lagi. Misalnya, Anda dapat memfilter metrik berdasarkan tag objek untuk mengidentifikasi set data yang tumbuh paling cepat, atau memvisualisasikan penyimpanan berdasarkan ukuran dan usia objek untuk

menginformasikan strategi arsip penyimpanan. Dengan begitu, grup Lensa Penyimpanan Amazon S3 membantu Anda untuk lebih memahami dan mengoptimalkan penyimpanan S3.

Saat menggunakan grup Lensa Penyimpanan, Anda dapat menganalisis dan memfilter metrik Lensa Penyimpanan S3 dengan menggunakan metadata objek seperti awalan, akhiran, tag objek, ukuran objek, atau usia objek. Anda juga dapat menerapkan kombinasi filter tersebut. Setelah melampirkan grup Lensa Penyimpanan ke dasbor Lensa Penyimpanan S3, Anda dapat melihat metrik Lensa Penyimpanan S3 yang dikumpulkan oleh grup Lensa Penyimpanan Amazon S3 secara langsung di dasbor.

Misalnya, Anda juga dapat memfilter metrik berdasarkan ukuran objek atau kelompok usia untuk menentukan bagian penyimpanan mana yang terdiri dari objek kecil. Lalu, Anda dapat menggunakan informasi ini dengan S3 Intelligent-Tiering atau S3 Lifecycle untuk mentransisikan objek kecil ke kelas penyimpanan yang berbeda untuk pengoptimalan biaya dan penyimpanan.

Topik

- Cara kerja grup Lensa Penyimpanan S3
- Menggunakan grup Lensa Penyimpanan

Cara kerja grup Lensa Penyimpanan S3

Anda dapat menggunakan grup Lensa Penyimpanan untuk menggabungkan metrik dengan menggunakan filter khusus berdasarkan metadata objek. Saat Anda menentukan filter kustom, Anda dapat menggunakan awalan, akhiran, tag objek, ukuran objek, usia objek, atau kombinasi dari filter kustom tersebut. Selama pembuatan grup Lensa Penyimpanan, Anda juga dapat menyertakan satu atau beberapa syarat filter. Untuk menentukan beberapa syarat filter, Anda dapat menggunakan `And` atau operator logis `Or`.

Saat Anda membuat dan mengonfigurasi grup Lensa Penyimpanan, grup Lensa Penyimpanan itu sendiri bertindak sebagai filter kustom di dasbor tempat grup dilampirkan. Di dasbor, Anda kemudian dapat menggunakan filter grup Lensa Penyimpanan untuk mendapatkan metrik penyimpanan berdasarkan filter kustom yang ditentukan di dalam grup.

Untuk melihat data untuk grup Lensa Penyimpanan di dasbor Lensa Penyimpanan S3, Anda harus melampirkan grup ke dasbor setelah grup tersebut dibuat. Setelah grup Lensa Penyimpanan terpasang ke dasbor Lensa Penyimpanan, dasbor akan mengumpulkan metrik penggunaan penyimpanan dalam waktu 48 jam. Anda kemudian dapat memvisualisasikan data tersebut di dasbor Lensa Penyimpanan atau mengekspornya melalui ekspor metrik. Jika Anda lupa melampirkan

grup Lensa Penyimpanan ke dasbor, data grup Lensa Penyimpanan Anda tidak akan diambil atau ditampilkan di mana pun.

> **Note**
>
> - Saat Anda membuat grup Lensa Penyimpanan S3, Anda membuat AWS sumber daya. Oleh karena itu, setiap grup Lensa Penyimpanan memiliki Amazon Resource Name (ARN) sendiri, yang dapat ditentukan saat melampirkannya atau mengecualikannya dari dasbor Lensa Penyimpanan S3.
> - Jika grup Lensa Penyimpanan tidak dilampirkan ke dasbor, Anda tidak akan dikenakan biaya tambahan untuk membuat grup Lensa Penyimpanan.
> - Lensa Penyimpanan S3 menggabungkan metrik penggunaan untuk objek di bawah semua grup Lensa Penyimpanan yang cocok. Oleh karena itu, jika suatu objek cocok dengan syarat filter untuk dua atau lebih grup Lensa Penyimpanan, Anda akan melihat penghitungan berulang untuk objek yang sama di seluruh penggunaan penyimpanan.

Anda dapat membuat grup Lensa Penyimpanan di tingkat akun di Wilayah beranda tertentu (dari daftar yang didukung Wilayah AWS). Lalu, Anda dapat melampirkan grup Lensa Penyimpanan ke beberapa dasbor Lensa Penyimpanan, selama dasbor berada di Akun AWS dan Wilayah asal yang sama. Anda dapat membuat hingga 50 grup Lensa Penyimpanan per Wilayah asal di masing-masing Akun AWS.

Anda dapat membuat dan mengelola grup Lensa Penyimpanan S3 menggunakan konsol Amazon S3 AWS Command Line Interface ,AWS CLI() AWS SDKs, atau Amazon S3 REST API.

Topik

- Melihat metrik grup Lensa Penyimpanan yang digabungkan
- Izin grup Lensa Penyimpanan
- Konfigurasi grup Lensa Penyimpanan
- AWS tag sumber daya
- Ekspor metrik grup Lensa Penyimpanan

Melihat metrik grup Lensa Penyimpanan yang digabungkan

Anda dapat melihat metrik yang digabungkan untuk grup Lensa Penyimpanan dengan melampirkan grup ke dasbor. Grup Lensa Penyimpanan yang ingin dilampirkan harus berada di Wilayah asal yang ditunjuk di akun dasbor.

Untuk melampirkan grup Lensa Penyimpanan ke dasbor, Anda harus menentukan grup di bagian Agregasi grup Lensa Penyimpanan pada konfigurasi dasbor. Jika Anda memiliki beberapa grup Lensa Penyimpanan, Anda dapat memfilter hasil Agregasi grup Lensa Penyimpanan untuk menyertakan atau mengecualikan hanya grup yang diinginkan saja. Untuk informasi selengkapnya tentang melampirkan grup ke dasbor, lihat the section called “Hubungkan atau hapus grup Lensa Penyimpanan”.

Setelah melampirkan grup, Anda akan melihat data agregasi grup Lensa Penyimpanan tambahan di dasbor dalam waktu 48 jam.

> **Note**
>
> Untuk melihat metrik yang digabungkan untuk grup Lensa Penyimpanan, Anda harus melampirkan grup ke dasbor Lensa Penyimpanan S3.

Izin grup Lensa Penyimpanan

Grup Lensa Penyimpanan memerlukan izin tertentu di AWS Identity and Access Management (IAM) untuk mengotorisasi akses ke tindakan grup Lensa Penyimpanan S3. Untuk memberikan izin tersebut, Anda dapat menggunakan kebijakan IAM berbasis identitas. Anda dapat melampirkan kebijakan ini ke pengguna, grup, atau peran IAM untuk diberikan izin. Izin tersebut dapat mencakup kemampuan untuk membuat atau menghapus grup Lensa Penyimpanan, melihat konfigurasinya, atau mengelola tag.

Pengguna atau peran AM yang diberikan izin harus dimiliki oleh akun yang membuat atau memiliki grup Lensa Penyimpanan.

Untuk dapat menggunakan grup Lensa Penyimpanan dan untuk melihat metrik grup Lensa Penyimpanan, Anda harus terlebih dahulu memiliki izin yang sesuai untuk menggunakan Lensa Penyimpanan S3. Untuk informasi selengkapnya, lihat the section called “Mengatur izin ”.

Untuk membuat dan mengelola grup Lensa Penyimpanan S3, Anda harus memiliki izin IAM berikut, tergantung pada tindakan yang ingin dilakukan:

Tindakan	Izin IAM
Buat grup Lensa Penyimpanan baru	`s3:CreateStorageLensGroup`
Buat grup Lensa Penyimpanan baru dengan tag	`s3:CreateStorageLensGroup` , `s3:TagResource`
Perbarui grup Lensa Penyimpanan yang sudah ada	`s3:UpdateStorageLensGroup`
Tampilkan detail konfigurasi grup Lensa Penyimpanan	`s3:GetStorageLensGroup`
Cantumkan semua grup Lensa Penyimpanan di Wilayah asal	`s3:ListStorageLensGroups`
Hapus grup Lensa Penyimpanan	`s3:DeleteStorageLensGroup`
Cantumkan daftar tag yang ditambahkan ke grup Lensa Penyimpanan	`s3:ListTagsForResource`
Tambahkan atau perbarui tag grup Lensa Penyimpanan untuk grup Lensa Penyimpanan yang sudah ada	`s3:TagResource`
Hapus tag dari grup Lensa Penyimpanan	`s3:UntagResource`

Berikut adalah contoh cara mengonfigurasi kebijakan IAM di akun yang membuat grup Lensa Penyimpanan. Untuk menggunakan kebijakan ini, ganti *`us-east-1`* dengan Wilayah asal tempat grup Lensa Penyimpanan Anda berada. Ganti *`111122223333`* dengan ID Akun AWS Anda, dan ganti *`example-storage-lens-group`* dengan nama grup Lensa Penyimpanan Anda. Untuk menerapkan izin ini ke semua grup Lensa Penyimpanan, ganti *`example-storage-lens-group`* dengan *.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EXAMPLE-Statement-ID",
```

```
            "Effect": "Allow",
            "Action": [
                "s3:CreateStorageLensGroup",
                "s3:UpdateStorageLensGroup",
                "s3:GetStorageLensGroup",
                "s3:ListStorageLensGroups",
                "s3:DeleteStorageLensGroup,
                "s3:TagResource",
                "s3:UntagResource",
                "s3:ListTagsForResource"
                ],
            "Resource": "arn:aws:s3:us-east-1:111122223333:storage-lens-group/example-storage-lens-group"
        }
    ]
}
```

Untuk informasi selengkapnya tentang izin Lensa Penyimpanan S3, lihat Menyetel izin Lensa Penyimpanan Amazon S3. Untuk informasi selengkapnya tentang bahasa kebijakan IAM, lihat Kebijakan dan izin di Amazon S3.

Konfigurasi grup Lensa Penyimpanan

Nama grup Lensa Penyimpanan S3

Sebaiknya berikan nama grup Lensa Penyimpanan sesuai dengan tujuannya sehingga Anda dapat dengan mudah menentukan grup mana yang ingin dilampirkan ke dasbor. Untuk dapat melampirkan grup Lensa Penyimpanan ke dasbor, Anda harus menentukan grup di bagian agregasi grup Lensa Penyimpanan pada konfigurasi dasbor.

Nama grup Lensa Penyimpanan harus unik di dalam akun tersebut. Nama tidak boleh melebihi 64 karakter, dan dapat berisi huruf (a-z, A-Z), angka (0-9), tanda hubung (-), dan garis bawah (_).

Wilayah Asal

Wilayah rumah adalah Wilayah AWS tempat grup Lensa Penyimpanan Anda dibuat dan dipelihara. Grup Lensa Penyimpanan Anda dibuat di Wilayah asal yang sama dengan dasbor Lensa Penyimpanan Amazon S3 Anda. Konfigurasi dan metrik grup Lensa Penyimpanan juga disimpan di Wilayah ini. Anda dapat membuat hingga 50 grup Lensa Penyimpanan di Wilayah asal.

Setelah Anda membuat grup Lensa Penyimpanan, Anda tidak dapat mengedit Wilayah asal.

Cakupan

Untuk menyertakan objek di dalam grup Lensa Penyimpanan, objek tersebut harus berada di cakupan dasbor Lensa Penyimpanan Amazon S3. Cakupan dasbor Lensa Penyimpanan ditentukan oleh bucket yang disertakan di dalam Cakupan dasbor untuk konfigurasi dasbor Lensa Penyimpanan S3 Anda.

Anda dapat menggunakan filter yang berbeda untuk objek guna menentukan cakupan grup Lensa Penyimpanan. Untuk melihat metrik grup Lensa Penyimpanan ini di dasbor Lensa Penyimpanan S3, objek harus cocok dengan filter yang disertakan di dalam grup Lensa Penyimpanan. Misalnya, grup Lensa Penyimpanan Anda menyertakan objek dengan awalan `marketing` dan akhiran `.png`, tetapi tidak ada objek yang cocok dengan kriteria tersebut. Dalam hal ini, metrik untuk grup Lensa Penyimpanan tersebut tidak akan dihasilkan di dalam ekspor metrik harian Anda, dan tidak ada metrik untuk grup ini yang akan terlihat di dasbor.

Filter

Anda dapat menggunakan filter berikut ke dalam grup Lensa Penyimpanan S3:

- Awalan–Menentukan awalan objek yang disertakan, yang merupakan string karakter di awal nama kunci objek tersebut. Misalnya, nilai `images` untuk filter Awalan akan mencakup objek dengan salah satu awalan berikut: `images/`, `images-marketing`, dan `images/production`. Panjang maksimum awalan adalah 1.024 byte.
- Akhiran-Menentukan akhiran objek yang disertakan (misalnya, `.png`, `.jpeg`, atau `.csv`). Panjang maksimum akhiran adalah 1.024 byte.
- Tag objek–Menentukan daftar tag objek yang ingin difilter. Kunci tag tidak boleh melebihi 128 karakter Unicode, dan nilai tag tidak boleh melebihi 256 karakter Unicode. Perhatikan bahwa jika bidang nilai tag objek dibiarkan kosong, grup Lensa Penyimpanan S3 hanya mencocokkan objek dengan objek lain yang juga memiliki nilai tag kosong.
- Usia-Menentukan rentang usia objek dari objek yang disertakan dalam hitungan hari. Hanya bilangan bulat yang didukung.
- Ukuran–Menentukan rentang ukuran objek dari objek disertakan dalam hitungan byte. Hanya bilangan bulat yang didukung. Nilai maksimum yang diperbolehkan adalah 5 TB.

Tag objek grup Lensa Penyimpanan

Anda dapat membuat grup Lensa Penyimpanan yang menyertakan hingga 10 filter tag objek. Contoh berikut mencakup dua pasangan nilai-kunci tag objek sebagai filter untuk grup Lensa Penyimpanan

yang dinamai *Marketing-Department*. Untuk menggunakan contoh ini, ganti *Marketing-Department* dengan nama grup Anda, dan ganti *object-tag-key-1*, *object-tag-value-1*, dan seterusnya dengan pasangan nilai kunci tag objek yang ingin difilter.

```
{
    "Name": "Marketing-Department",
    "Filter": {
     "MatchAnyTag":[
                {
                    "Key": "object-tag-key-1",
                    "Value": "object-tag-value-1"
                },
                {
                    "Key": "object-tag-key-2",
                    "Value": "object-tag-value-2"
                }
            ]
    }
}
```

Operator logis (**And** atau **Or**)

Untuk menyertakan beberapa kondisi filter di grup Lensa Penyimpanan, Anda dapat menggunakan operator logis (And atau Or). Di contoh berikut, grup Lensa Penyimpanan yang dinamai *Marketing-Department* memiliki operator And yang berisi filter Prefix, ObjectAge, dan ObjectSize. Karena operator And digunakan, hanya objek yang cocok dengan semua syarat filter ini yang akan disertakan di dalam cakupan grup Lensa Penyimpanan.

Untuk menggunakan contoh ini, ganti *user input placeholders* dengan nilai yang ingin difilter.

```
{
    "Name": "Marketing-Department",
    "Filter": {
        "And": {
            "MatchAnyPrefix": [
                "prefix-1",
                "prefix-2",
                "prefix-3/sub-prefix-1"
            ],
             "MatchObjectAge": {
                "DaysGreaterThan": 10,
                "DaysLessThan": 60
```

```
            },
            "MatchObjectSize": {
                "BytesGreaterThan": 10,
                "BytesLessThan": 60
            }
        }
    }
}
```

> **Note**
>
> Jika Anda ingin menyertakan objek yang cocok dengan salah satu syarat filter, ganti operator logis `And` dengan operator logis `Or` dalam contoh ini.

AWS tag sumber daya

Setiap grup Lensa Penyimpanan S3 dihitung sebagai AWS sumber daya dengan Nama Sumber Daya Amazon (ARN) miliknya sendiri. Oleh karena itu, Anda mengonfigurasi grup Lensa Penyimpanan, Anda dapat menambahkan tag sumber daya AWS ke grup secara opsional. Anda dapat menambahkan hingga 50 tag per grup Lensa Penyimpanan. Untuk membuat grup Lensa Penyimpanan dengan tag, Anda harus memiliki izin `s3:CreateStorageLensGroup` dan `s3:TagResource`.

Anda dapat menggunakan tag AWS sumber daya untuk mengkategorikan sumber daya menurut departemen, lini bisnis, atau proyek. Langkah ini berguna ketika Anda memiliki banyak sumber daya dengan jenis yang sama. Dengan menerapkan tag, Anda dapat dengan cepat mengidentifikasi grup Lensa Penyimpanan tertentu berdasarkan tag yang telah ditetapkan. Anda juga dapat menggunakan tag untuk melacak dan mengalokasikan biaya.

Selain itu, saat menambahkan tag AWS sumber daya ke grup Lensa Penyimpanan, Anda mengaktifkan kontrol akses berbasis atribut (ABAC). ABAC adalah strategi otorisasi yang mendefinisikan izin berdasarkan atribut, dalam hal ini tag. Anda juga dapat menggunakan kondisi yang menentukan tag sumber daya dalam kebijakan IAM Anda untuk mengontrol akses ke AWS sumber daya.

Anda dapat mengedit kunci dan nilai tanda, dan Anda dapat menghapus tanda dari sumber daya kapan pun. Selain itu, perhatikan batasan berikut:

- Kunci tag dan nilai tag peka huruf besar-kecil.

- Jika Anda menambahkan tanda yang memiliki kunci yang sama dengan tanda yang sudah ada di sumber daya tersebut, nilai baru akan menimpa nilai lama.
- Jika sumber daya dihapus, semua tanda untuk sumber daya tersebut juga akan dihapus.
- Jangan sertakan data pribadi atau sensitif dalam tag AWS sumber daya Anda.
- Tag sistem (atau tag dengan kunci tag yang dimulai dengan `aws:`) tidak didukung.
- Panjang setiap kunci tag tidak boleh melebihi 128 karakter. Panjang setiap nilai tag tidak boleh melebihi 256 karakter.

Ekspor metrik grup Lensa Penyimpanan

Metrik grup Lensa Penyimpanan S3 disertakan di dalam ekspor metrik Lensa Penyimpanan Amazon S3 untuk dasbor tempat grup Lensa Penyimpanan dilampirkan. Untuk informasi umum tentang fitur ekspor metrik Lensa Penyimpanan, lihat Melihat metrik Lensa Penyimpanan Amazon S3 menggunakan ekspor data.

Ekspor metrik untuk grup Lensa Penyimpanan mencakup metrik Lensa Penyimpanan S3 yang berada di dalam cakupan dasbor tempat grup Lensa Penyimpanan dilampirkan. Ekspor ini juga mencakup data metrik tambahan untuk grup Lensa Penyimpanan.

Setelah membuat grup Lensa Penyimpanan, ekspor metrik akan dikirim setiap hari ke bucket yang dipilih saat Anda mengonfigurasi ekspor metrik untuk dasbor tempat grup dilampirkan. Anda memerlukan waktu hingga 48 jam untuk dapat menerima ekspor metrik pertama.

Untuk menghasilkan metrik di ekspor harian, objek harus cocok dengan filter yang disertakan di dalam grup Lensa Penyimpanan. Jika tidak ada objek yang cocok dengan filter yang disertakan di dalam grup Lensa Penyimpanan, tidak ada metrik yang akan dihasilkan. Namun, jika suatu objek cocok dengan dua atau lebih grup Lensa Penyimpanan, objek tersebut akan dicantumkan secara terpisah untuk setiap grup saat muncul di ekspor metrik.

Anda dapat mengidentifikasi metrik untuk grup Lensa Penyimpanan dengan mencari salah satu nilai berikut di kolom `record_type` ekspor metrik untuk dasbor:

- `STORAGE_LENS_GROUP_BUCKET`
- `STORAGE_LENS_GROUP_ACCOUNT`

Kolom `record_value` menampilkan ARN sumber daya untuk grup Lensa Penyimpanan (misalnya, `arn:aws:s3:us-east-1:111122223333:storage-lens-group/Marketing-Department`).

Menggunakan grup Lensa Penyimpanan

Grup Lensa Penyimpanan Amazon S3 menggabungkan metrik dengan menggunakan filter khusus berdasarkan metadata objek. Anda dapat menganalisis dan memfilter metrik Lensa Penyimpanan S3 dengan menggunakan awalan, akhiran, tag objek, ukuran objek, atau usia objek. Dengan grup Lensa Penyimpanan Amazon S3, Anda juga dapat mengkategorikan penggunaan di dalam dan di seluruh bucket Amazon S3. Dengan begitu, Anda akan dapat lebih memahami dan mengoptimalkan penyimpanan S3 Anda.

Untuk mulai memvisualisasikan data untuk grup Lensa Penyimpanan, Anda harus terlebih dahulu melampirkan grup Lensa Penyimpanan ke dasbor Lensa Penyimpanan S3. Jika Anda perlu mengelola grup Lensa Penyimpanan di dasbor, Anda dapat mengedit konfigurasi dasbor. Untuk memeriksa grup Lensa Penyimpanan mana yang berada di akun, Anda dapat mencantumkannya. Untuk memeriksa grup Lensa Penyimpanan mana yang dilampirkan ke dasbor, Anda selalu dapat memeriksa tab Grup Lensa Penyimpanan di dasbor. Untuk meninjau atau memperbarui cakupan grup Lensa Penyimpanan yang sudah ada, Anda dapat melihat detailnya. Anda juga dapat menghapus grup Lensa Penyimpanan secara permanen.

Untuk mengelola izin, Anda dapat membuat dan menambahkan tag AWS sumber daya yang ditentukan pengguna ke grup Lensa Penyimpanan Anda. Anda dapat menggunakan tag AWS sumber daya untuk mengkategorikan sumber daya menurut departemen, lini bisnis, atau proyek. Langkah ini berguna ketika Anda memiliki banyak sumber daya dengan jenis yang sama. Dengan menerapkan tag, Anda dapat dengan cepat mengidentifikasi grup Lensa Penyimpanan tertentu berdasarkan tag yang telah ditetapkan.

Selain itu, saat menambahkan tag AWS sumber daya ke grup Lensa Penyimpanan, Anda mengaktifkan kontrol akses berbasis atribut (ABAC). ABAC adalah strategi otorisasi yang mendefinisikan izin berdasarkan atribut, dalam hal ini tag. Anda juga dapat menggunakan kondisi yang menentukan tag sumber daya dalam kebijakan IAM Anda untuk mengontrol akses ke AWS sumber daya.

Topik

- Membuat grup Lensa Penyimpanan
- Menghubungkan atau menghapus grup Lensa Penyimpanan S3 ke atau dari dasbor
- Memvisualisasikan data grup Lensa Penyimpanan
- Memperbarui grup Lensa Penyimpanan
- Mengelola tag AWS sumber daya dengan grup Lensa Penyimpanan

- Mencantumkan semua grup Lensa Penyimpanan
- Melihat detail grup Lensa Penyimpanan
- Menghapus grup Lensa Penyimpanan

Membuat grup Lensa Penyimpanan

Contoh berikut menunjukkan cara membuat grup Lensa Penyimpanan Amazon S3 dengan menggunakan konsol Amazon S3 AWS Command Line Interface ,AWS CLI(), dan. AWS SDK untuk Java

Menggunakan konsol S3

Untuk membuat grup Lensa Penyimpanan

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi di bagian atas halaman, pilih nama AWS Wilayah yang saat ini ditampilkan. Selanjutnya, pilih Wilayah yang ingin Anda alihkan.
3. Di panel navigasi bagian kiri, pilih Grup Lensa Penyimpanan.
4. Pilih Buat grup Lensa Penyimpanan.
5. Di bawah Umum, lihat Wilayah Rumah Anda dan masukkan nama grup Lensa Penyimpanan Anda.
6. Di bagian bawah Cakupan, pilih filter yang ingin Anda terapkan ke grup Lensa Penyimpanan. Untuk menerapkan beberapa filter, pilih filter Anda, lalu pilih operator logika AND atau OR.
   - Untuk filter Awalan, pilih Awalan, dan masukkan string awalan. Untuk menambahkan beberapa awalan, pilih Tambah awalan. Untuk menghapus awalan, pilih Hapus di bagian samping awalan yang ingin dihapus.
   - Untuk filter Tag objek, pilih Tag objek, dan masukkan pasangan nilai-kunci untuk objek Anda. Kemudian, pilih Tambah tag. Untuk menghapus tag, pilih Hapus yang ada di bagian samping tag yang ingin dihapus.
   - Untuk filter Akhiran, pilih Akhiran, dan masukkan string akhiran. Untuk menambahkan beberapa akhiran, pilih Tambah akhiran. Untuk menghapus akhiran, pilih Hapus di bagian samping akhiran yang ingin dihapus.

- Untuk filter Usia, tentukan rentang usia objek dalam hitungan hari. Pilih Tentukan usia objek minimum, dan masukkan usia objek minimum. Lalu, pilih Tentukan usia objek maksimum, dan masukkan usia objek maksimum.
- Untuk filter Ukuran, tentukan rentang ukuran objek dan unit pengukurannya. Pilih Tentukan ukuran objek minimum, dan masukkan ukuran objek minimum. Pilih Tentukan ukuran objek maksimum, dan masukkan ukuran objek maksimum.

7. (Opsional) Untuk tag AWS sumber daya, tambahkan pasangan kunci-nilai, lalu pilih Tambah tag.
8. Pilih Buat grup Lensa Penyimpanan.

Menggunakan AWS CLI

Contoh AWS CLI perintah berikut membuat grup Storage Lens. Untuk menggunakan perintah contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3control create-storage-lens-group --account-id 111122223333 \
--region us-east-1 --storage-lens-group=file://./marketing-department.json
```

Contoh AWS CLI perintah berikut membuat grup Storage Lens dengan dua tag AWS sumber daya. Untuk menggunakan perintah contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3control create-storage-lens-group --account-id 111122223333 \
--region us-east-1 --storage-lens-group=file://./marketing-department.json \
--tags Key=k1,Value=v1 Key=k2,Value=v2
```

Untuk contoh konfigurasi JSON, lihat Konfigurasi grup Lensa Penyimpanan.

Menggunakan AWS SDK for Java

AWS SDK untuk Java Contoh berikut membuat grup Storage Lens. Untuk menggunakan contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

Example — Buat grup Lensa Penyimpanan dengan satu filter

Contoh berikut membuat grup Lensa Penyimpanan dengan nama *Marketing-Department*. Grup ini memiliki filter usia objek yang menentukan rentang usia *30* hingga *90* hari. Untuk menggunakan contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
package aws.example.s3control;
```

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.CreateStorageLensGroupRequest;
import software.amazon.awssdk.services.s3control.model.MatchObjectAge;
import software.amazon.awssdk.services.s3control.model.StorageLensGroup;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupFilter;

public class CreateStorageLensGroupWithObjectAge {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            StorageLensGroupFilter objectAgeFilter = StorageLensGroupFilter.builder()
                    .matchObjectAge(MatchObjectAge.builder()
                            .daysGreaterThan(30)
                            .daysLessThan(90)
                            .build())
                    .build();

            StorageLensGroup storageLensGroup = StorageLensGroup.builder()
                    .name(storageLensGroupName)
                    .filter(objectAgeFilter)
                    .build();

            CreateStorageLensGroupRequest createStorageLensGroupRequest =
 CreateStorageLensGroupRequest.builder()
                    .storageLensGroup(storageLensGroup)
                    .accountId(accountId).build();

            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            s3ControlClient.createStorageLensGroup(createStorageLensGroupRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
```

```
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Example — Buat grup Lensa Penyimpanan dengan operator **AND** yang menyertakan beberapa filter

Contoh berikut membuat grup Lensa Penyimpanan dengan nama *Marketing-Department*. Grup ini menggunakan operator AND untuk menunjukkan bahwa objek harus cocok dengan semua kondisi filter. Untuk menggunakan contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.CreateStorageLensGroupRequest;
import software.amazon.awssdk.services.s3control.model.MatchObjectAge;
import software.amazon.awssdk.services.s3control.model.MatchObjectSize;
import software.amazon.awssdk.services.s3control.model.S3Tag;
import software.amazon.awssdk.services.s3control.model.StorageLensGroup;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupAndOperator;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupFilter;


public class CreateStorageLensGroupWithAndFilter {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            // Create object tags.
            S3Tag tag1 = S3Tag.builder()
                    .key("object-tag-key-1")
                    .value("object-tag-value-1")
                    .build();
            S3Tag tag2 = S3Tag.builder()
                    .key("object-tag-key-2")
```

```
                    .value("object-tag-value-2")
                    .build();

            StorageLensGroupAndOperator andOperator =
StorageLensGroupAndOperator.builder()
                    .matchAnyPrefix("prefix-1", "prefix-2", "prefix-3/sub-prefix-1")
                    .matchAnySuffix(".png", ".gif", ".jpg")
                    .matchAnyTag(tag1, tag2)
                    .matchObjectAge(MatchObjectAge.builder()
                            .daysGreaterThan(30)
                            .daysLessThan(90).build())
                    .matchObjectSize(MatchObjectSize.builder()
                            .bytesGreaterThan(1000L)
                            .bytesLessThan(6000L).build())
                    .build();

            StorageLensGroupFilter andFilter = StorageLensGroupFilter.builder()
                    .and(andOperator)
                    .build();

            StorageLensGroup storageLensGroup = StorageLensGroup.builder()
                    .name(storageLensGroupName)
                    .filter(andFilter)
                    .build();

            CreateStorageLensGroupRequest createStorageLensGroupRequest =
CreateStorageLensGroupRequest.builder()
                    .storageLensGroup(storageLensGroup)
                    .accountId(accountId).build();

            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            s3ControlClient.createStorageLensGroup(createStorageLensGroupRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
```

```
    }
}
```

Example — Buat grup Lensa Penyimpanan dengan operator **OR** yang menyertakan beberapa filter

Contoh berikut membuat grup Lensa Penyimpanan dengan nama *Marketing-Department*. Grup ini menggunakan operator OR untuk menerapkan filter awalan (*prefix-1*, *prefix-2*, *prefix3/sub-prefix-1*) atau filter ukuran objek dengan rentang ukuran antara *1000* byte dan *6000* byte. Untuk menggunakan contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.CreateStorageLensGroupRequest;
import software.amazon.awssdk.services.s3control.model.MatchObjectSize;
import software.amazon.awssdk.services.s3control.model.StorageLensGroup;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupFilter;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupOrOperator;

public class CreateStorageLensGroupWithOrFilter {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            StorageLensGroupOrOperator orOperator =
 StorageLensGroupOrOperator.builder()
                    .matchAnyPrefix("prefix-1", "prefix-2", "prefix-3/sub-prefix-1")
                    .matchObjectSize(MatchObjectSize.builder()
                            .bytesGreaterThan(1000L)
                            .bytesLessThan(6000L)
                            .build())
                    .build();

            StorageLensGroupFilter orFilter = StorageLensGroupFilter.builder()
                    .or(orOperator)
                    .build();

            StorageLensGroup storageLensGroup = StorageLensGroup.builder()
```

```
                    .name(storageLensGroupName)
                    .filter(orFilter)
                    .build();

            CreateStorageLensGroupRequest createStorageLensGroupRequest =
 CreateStorageLensGroupRequest.builder()
                    .storageLensGroup(storageLensGroup)
                    .accountId(accountId).build();

            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            s3ControlClient.createStorageLensGroup(createStorageLensGroupRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Example — Buat grup Lensa Penyimpanan dengan satu filter dan dua tag AWS sumber daya

Contoh berikut membuat grup Lensa Penyimpanan dengan nama *Marketing-Department* yang memiliki filter akhiran. Contoh ini juga menambahkan dua tag AWS sumber daya ke grup Storage Lens. Untuk menggunakan contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.CreateStorageLensGroupRequest;
import software.amazon.awssdk.services.s3control.model.StorageLensGroup;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupFilter;
```

```
import software.amazon.awssdk.services.s3control.model.Tag;

public class CreateStorageLensGroupWithResourceTags {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            // Create AWS resource tags.
            Tag resourceTag1 = Tag.builder()
                    .key("resource-tag-key-1")
                    .value("resource-tag-value-1")
                    .build();
            Tag resourceTag2 = Tag.builder()
                    .key("resource-tag-key-2")
                    .value("resource-tag-value-2")
                    .build();

            StorageLensGroupFilter suffixFilter = StorageLensGroupFilter.builder()
                    .matchAnySuffix(".png", ".gif", ".jpg")
                    .build();

            StorageLensGroup storageLensGroup = StorageLensGroup.builder()
                    .name(storageLensGroupName)
                    .filter(suffixFilter)
                    .build();

            CreateStorageLensGroupRequest createStorageLensGroupRequest =
 CreateStorageLensGroupRequest.builder()
                    .storageLensGroup(storageLensGroup)
                    .tags(resourceTag1, resourceTag2)
                    .accountId(accountId).build();

            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            s3ControlClient.createStorageLensGroup(createStorageLensGroupRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
```

```
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Untuk contoh konfigurasi JSON, lihat Konfigurasi grup Lensa Penyimpanan.

Menghubungkan atau menghapus grup Lensa Penyimpanan S3 ke atau dari dasbor

Setelah memperbarui ke tingkat lanjutan di Lensa Penyimpanan Amazon S3, Anda dapat menghubungkan grup Lensa Penyimpanan ke dasbor Anda. Jika memiliki beberapa grup Lensa Penyimpanan, Anda dapat menyertakan atau mengecualikan grup yang Anda inginkan.

Grup Lensa Penyimpanan Anda harus berada di dalam Wilayah asal yang ditunjuk di akun dasbor. Setelah menghubungkan grup Lensa Penyimpanan ke dasbor, Anda akan menerima data agregasi grup Lensa Penyimpanan tambahan di dalam ekspor metrik dalam waktu 48 jam.

> **Note**
>
> Jika Anda ingin melihat metrik gabungan untuk grup Lensa Penyimpanan, Anda harus melampirkannya ke dasbor Lensa Penyimpanan. Untuk contoh file konfigurasi JSON grup Lensa Penyimpanan, lihat Konfigurasi contoh Lensa Penyimpanan S3 dengan grup Lensa Penyimpanan di JSON.

Menggunakan konsol S3

Untuk menghubungkan grup Lensa Penyimpanan ke dasbor Lensa Penyimpanan

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, di bawah Lensa Penyimpanan, pilih Dasbor.
3. Pilih tombol opsi untuk dasbor Lensa Penyimpanan yang ingin Anda pasangkan ke grup Lensa Penyimpanan.
4. Pilih Edit.
5. Di bawah Pilihan metrik, pilih Metrik dan rekomendasi lanjutan.
6. Pilih Agregasi grup Lensa Penyimpanan.

> **Note**
>
> Secara default, Metrik lanjutan juga akan dipilih. Namun, pilihan pengaturan ini dapat dibatalkan karena tidak diperlukan untuk menggabungkan data grup Lensa Penyimpanan.

7. Gulir ke bawah ke Agregasi grup Lensa Penyimpanan dan tentukan grup atau grup Lensa Penyimpanan yang ingin disertakan atau dikecualikan di dalam agregasi data. Anda dapat menggunakan opsi pemfilteran berikut:
   - Jika Anda ingin menyertakan grup Lensa Penyimpanan tertentu, pilih Sertakan grup Lensa Penyimpanan. Di bawah Grup Lensa Penyimpanan yang akan disertakan, pilih grup Lensa Penyimpanan Anda.
   - Jika Anda ingin menyertakan semua grup Lensa Penyimpanan, pilih Sertakan semua grup Lensa Penyimpanan di wilayah asal di akun ini.
   - Jika Anda ingin mengecualikan grup Lensa Penyimpanan tertentu, pilih Kecualikan grup Lensa Penyimpanan. Di bawah grup Lensa Penyimpanan yang akan dikecualikan, pilih grup Lensa Penyimpanan yang ingin Anda kecualikan.
8. Pilih Simpan perubahan. Jika selesai mengonfigurasi grup Lensa Penyimpanan dengan benar, Anda akan melihat data agregasi grup Lensa Penyimpanan tambahan di dasbor dalam waktu 48 jam.

Untuk menghapus grup Lensa Penyimpanan dari dasbor Lensa Penyimpanan S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, di bawah Lensa Penyimpanan, pilih Dasbor.
3. Pilih tombol opsi untuk dasbor Lensa Penyimpanan tempat Anda ingin menghapus grup Lensa Penyimpanan.
4. Pilih Lihat konfigurasi dasbor.
5. Pilih Edit.
6. Gulir ke bawah ke bagian Pilihan metrik.
7. Di bawah Agregasi grup Lensa Penyimpanan, pilih X di sebelah grup Lensa Penyimpanan yang ingin dihapus. Langkah ini akan menghapus grup Lensa Penyimpanan Anda.

Jika Anda menyertakan semua grup Lensa Penyimpanan di dasbor, hapus kotak centang di samping Sertakan semua grup Lensa Penyimpanan di Wilayah asal di akun ini.

8. Pilih Simpan perubahan.

> **Note**
>
> Diperlukan waktu hingga 48 jam untuk dasbor Anda dapat menampilkan pembaruan konfigurasi.

Menggunakan AWS SDK for Java

Example — Pasang semua grup Lensa Penyimpanan ke dasbor

Contoh SDK for Java berikut ini melampirkan semua grup Storage Lens di *111122223333* akun ke *DashBoardConfigurationId* dasbor:

```
package aws.example.s3control;


import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.BucketLevel;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationRequest;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.model.AccountLevel;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensGroupLevel;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class CreateDashboardWithStorageLensGroups {
    public static void main(String[] args) {
        String configurationId = "ExampleDashboardConfigurationId";
        String sourceAccountId = "111122223333";

        try {
            StorageLensGroupLevel storageLensGroupLevel = new StorageLensGroupLevel();
```

```
            AccountLevel accountLevel = new AccountLevel()
                    .withBucketLevel(new BucketLevel())
                    .withStorageLensGroupLevel(storageLensGroupLevel);

            StorageLensConfiguration configuration = new StorageLensConfiguration()
                    .withId(configurationId)
                    .withAccountLevel(accountLevel)
                    .withIsEnabled(true);

            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.putStorageLensConfiguration(new
 PutStorageLensConfigurationRequest()
                    .withAccountId(sourceAccountId)
                    .withConfigId(configurationId)
                    .withStorageLensConfiguration(configuration)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Example — Pasang dua grup Lensa Penyimpanan ke dasbor

AWS SDK untuk Java Contoh berikut melampirkan dua grup Storage Lens (*StorageLensGroupName1*dan*StorageLensGroupName2*) ke *ExampleDashboardConfigurationId* dasbor.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
```

```
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.AccountLevel;
import com.amazonaws.services.s3control.model.BucketLevel;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensGroupLevel;
import com.amazonaws.services.s3control.model.StorageLensGroupLevelSelectionCriteria;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class CreateDashboardWith2StorageLensGroups {
    public static void main(String[] args) {
        String configurationId = "ExampleDashboardConfigurationId";
        String storageLensGroupName1 = "StorageLensGroupName1";
        String storageLensGroupName2 = "StorageLensGroupName2";
        String sourceAccountId = "111122223333";

        try {
            StorageLensGroupLevelSelectionCriteria selectionCriteria = new
 StorageLensGroupLevelSelectionCriteria()
                    .withInclude(
                            "arn:aws:s3:" + US_WEST_2.getName() + ":" + sourceAccountId
 + ":storage-lens-group/" + storageLensGroupName1,
                            "arn:aws:s3:" + US_WEST_2.getName() + ":" + sourceAccountId
 + ":storage-lens-group/" + storageLensGroupName2);

            System.out.println(selectionCriteria);
            StorageLensGroupLevel storageLensGroupLevel = new StorageLensGroupLevel()
                    .withSelectionCriteria(selectionCriteria);

            AccountLevel accountLevel = new AccountLevel()
                    .withBucketLevel(new BucketLevel())
                    .withStorageLensGroupLevel(storageLensGroupLevel);

            StorageLensConfiguration configuration = new StorageLensConfiguration()
                    .withId(configurationId)
                    .withAccountLevel(accountLevel)
                    .withIsEnabled(true);

            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();
```

```
            s3ControlClient.putStorageLensConfiguration(new
 PutStorageLensConfigurationRequest()
                    .withAccountId(sourceAccountId)
                    .withConfigId(configurationId)
                    .withStorageLensConfiguration(configuration)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Example — Lampirkan semua grup Lensa Penyimpanan dengan pengecualian

Contoh SDK for Java berikut ini melampirkan semua grup Storage Lens ke *ExampleDashboardConfigurationId* dasbor, tidak termasuk dua yang ditentukan *StorageLensGroupName1* (dan): *StorageLensGroupName2*

```
package aws.example.s3control;


import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.AccountLevel;
import com.amazonaws.services.s3control.model.BucketLevel;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensGroupLevel;
import com.amazonaws.services.s3control.model.StorageLensGroupLevelSelectionCriteria;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class CreateDashboardWith2StorageLensGroupsExcluded {
```

```
    public static void main(String[] args) {
        String configurationId = "ExampleDashboardConfigurationId";
        String storageLensGroupName1 = "StorageLensGroupName1";
        String storageLensGroupName2 = "StorageLensGroupName2";
        String sourceAccountId = "111122223333";

        try {
            StorageLensGroupLevelSelectionCriteria selectionCriteria = new
StorageLensGroupLevelSelectionCriteria()
                    .withInclude(
                            "arn:aws:s3:" + US_WEST_2.getName() + ":" + sourceAccountId
+ ":storage-lens-group/" + storageLensGroupName1,
                            "arn:aws:s3:" + US_WEST_2.getName() + ":" + sourceAccountId
+ ":storage-lens-group/" + storageLensGroupName2);

            System.out.println(selectionCriteria);
            StorageLensGroupLevel storageLensGroupLevel = new StorageLensGroupLevel()
                    .withSelectionCriteria(selectionCriteria);

            AccountLevel accountLevel = new AccountLevel()
                    .withBucketLevel(new BucketLevel())
                    .withStorageLensGroupLevel(storageLensGroupLevel);

            StorageLensConfiguration configuration = new StorageLensConfiguration()
                    .withId(configurationId)
                    .withAccountLevel(accountLevel)
                    .withIsEnabled(true);

            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.putStorageLensConfiguration(new
PutStorageLensConfigurationRequest()
                    .withAccountId(sourceAccountId)
                    .withConfigId(configurationId)
                    .withStorageLensConfiguration(configuration)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
```

```
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Memvisualisasikan data grup Lensa Penyimpanan

Anda dapat memvisualisasikan data grup Lensa Penyimpanan dengan melampirkan grup ke dasbor Lensa Penyimpanan Amazon S3 Anda. Setelah menyertakan grup Lensa Penyimpanan di agregasi grup Lensa Penyimpanan di dalam konfigurasi dasbor, data grup Lensa Penyimpanan dapat ditampilkan hingga 48 jam di dasbor.

Setelah konfigurasi dasbor berhasil diperbarui, grup Lensa Penyimpanan yang baru dilampirkan akan muncul di daftar sumber daya yang tersedia di bawah tab grup Lensa Penyimpanan. Anda juga dapat menganalisis lebih lanjut penggunaan penyimpanan di tab Ikhtisar dengan membagi data berdasarkan dimensi lain. Misalnya, Anda dapat memilih salah satu item yang tercantum di bawah 3 kategori teratas dan memilih Analisis dengan untuk membagi data berdasarkan dimensi lain. Anda tidak dapat menerapkan dimensi yang sama dengan filter itu sendiri.

> **Note**
>
> Anda tidak dapat menerapkan filter grup Lensa Penyimpanan bersama dengan filter awalan, atau sebaliknya. Anda juga tidak dapat menganalisis grup Lensa Penyimpanan lebih lanjut dengan menggunakan filter awalan.

Anda dapat menggunakan tab Grup Lensa Penyimpanan di dasbor Lensa Penyimpanan Amazon S3 untuk menyesuaikan visualisasi data untuk grup Lensa Penyimpanan yang dilampirkan ke dasbor. Anda dapat memvisualisasikan data untuk beberapa grup Lensa Penyimpanan yang dilampirkan ke dasbor, atau semua grup.

Saat memvisualisasikan data grup Lensa Penyimpanan di dasbor Lensa Penyimpanan S3, perhatikan hal-hal berikut:

- Lensa Penyimpanan S3 menggabungkan metrik penggunaan untuk objek di bawah semua grup Lensa Penyimpanan yang cocok. Oleh karena itu, jika suatu objek cocok dengan syarat filter untuk dua atau lebih grup Lensa Penyimpanan, Anda akan melihat penghitungan berulang untuk objek yang sama di seluruh penggunaan penyimpanan.

- Objek harus cocok dengan filter yang disertakan di dalam grup Lensa Penyimpanan. Jika tidak ada objek yang cocok dengan filter yang disertakan di dalam grup Lensa Penyimpanan, tidak ada metrik yang akan dihasilkan. Untuk menentukan apakah ada objek yang belum ditetapkan, periksa jumlah total objek Anda di dasbor pada tingkat akun dan tingkat bucket.

Memperbarui grup Lensa Penyimpanan

Contoh berikut ini menunjukkan cara memperbarui grup Lensa Penyimpanan Amazon S3. Anda dapat memperbarui grup Lensa Penyimpanan menggunakan konsol Amazon S3, AWS Command Line Interface (AWS CLI), dan. AWS SDK untuk Java

Menggunakan konsol S3

Untuk memperbarui grup Lensa Penyimpanan

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, pilih Grup Lensa Penyimpanan.
3. Di bawah Grup Lensa Penyimpanan, pilih grup Lensa Penyimpanan yang ingin diperbarui.
4. Di bawah Cakupan, pilih Edit.
5. Di laman Cakupan, pilih filter yang ingin dierapkan ke grup Lensa Penyimpanan. Untuk menerapkan beberapa filter, pilih filter Anda, dan pilih operator logis AND atau OR.
   - Untuk filter Awalan, pilih Awalan, dan masukkan string awalan. Untuk menambahkan beberapa awalan, pilih Tambah awalan. Untuk menghapus awalan, pilih Hapus di bagian samping awalan yang ingin dihapus.
   - Untuk filter Tag objek, masukkan pasangan nilai-kunci untuk objek Anda. Kemudian, pilih Tambah tag. Untuk menghapus tag, pilih Hapus yang ada di bagian samping tag yang ingin dihapus.
   - Untuk filter Akhiran, pilih Akhiran, dan masukkan string akhiran. Untuk menambahkan beberapa akhiran, pilih Tambah akhiran. Untuk menghapus akhiran, pilih Hapus di bagian samping akhiran yang ingin dihapus.
   - Untuk filter Usia, tentukan rentang usia objek dalam hitungan hari. Pilih Tentukan usia objek minimum, dan masukkan usia objek minimum. Untuk Tentukan usia objek maksimum, masukkan usia objek maksimum.

- Untuk filter Ukuran, tentukan rentang ukuran objek dan unit pengukurannya. Pilih Tentukan ukuran objek minimum, dan masukkan ukuran objek minimum. Untuk Tentukan ukuran objek maksimum, masukkan ukuran objek maksimum.

6. Pilih Simpan perubahan. Laman rincian untuk grup Lensa Penyimpanan akan muncul.
7. (Opsional) Jika Anda ingin menambahkan tag AWS sumber daya baru, gulir ke bagian tag AWS sumber daya, lalu pilih Tambahkan tag. Laman Tambah tag akan muncul.

   Tambahkan pasangan nilai-kunci baru, lalu pilih Simpan perubahan. Laman rincian untuk grup Lensa Penyimpanan akan muncul.
8. (Opsional) Jika Anda ingin menghapus tag AWS sumber daya yang ada, gulir ke bagian tag AWS sumber daya, dan pilih tag sumber daya. Lalu, pilih Hapus. Kotak dialog Hapus tag AWS akan muncul.

   Pilih Hapus lagi untuk menghapus tag sumber daya AWS secara permanen.

   > **Note**
   >
   > Setelah Anda menghapus tag AWS sumber daya secara permanen, tag tersebut tidak dapat dipulihkan.

Menggunakan AWS CLI

AWS CLI Contoh perintah berikut mengembalikan rincian konfigurasi untuk grup Storage Lens bernama*marketing-department*. Untuk menggunakan perintah contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3control get-storage-lens-group --account-id 111122223333 \
--region us-east-1 --name marketing-department
```

AWS CLI Contoh berikut memperbarui grup Storage Lens. Untuk menggunakan perintah contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3control update-storage-lens-group --account-id 111122223333 \
--region us-east-1 --storage-lens-group=file://./marketing-department.json
```

Untuk contoh konfigurasi JSON, lihat Konfigurasi grup Lensa Penyimpanan.

Menggunakan AWS SDK for Java

AWS SDK untuk Java Contoh berikut mengembalikan detail konfigurasi untuk grup *Marketing-Department* Storage Lens di akun*111122223333*. Untuk menggunakan contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.GetStorageLensGroupRequest;
import software.amazon.awssdk.services.s3control.model.GetStorageLensGroupResponse;

public class GetStorageLensGroup {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            GetStorageLensGroupRequest getRequest =
 GetStorageLensGroupRequest.builder()
                    .name(storageLensGroupName)
                    .accountId(accountId).build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            GetStorageLensGroupResponse response =
 s3ControlClient.getStorageLensGroup(getRequest);
            System.out.println(response);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
```

```
}
```

Contoh berikut memperbarui grup Lensa Penyimpanan yang dinamai *Marketing-Department* di akun *111122223333*. Contoh ini memperbarui cakupan dasbor untuk menyertakan objek yang cocok dengan salah satu akhiran berikut: *.png*, *.gif*, *.jpg*, atau *.jpeg*. Untuk menggunakan contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.StorageLensGroup;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupFilter;
import software.amazon.awssdk.services.s3control.model.UpdateStorageLensGroupRequest;

public class UpdateStorageLensGroup {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            // Create updated filter.
            StorageLensGroupFilter suffixFilter = StorageLensGroupFilter.builder()
                    .matchAnySuffix(".png", ".gif", ".jpg", ".jpeg")
                    .build();

            StorageLensGroup storageLensGroup = StorageLensGroup.builder()
                    .name(storageLensGroupName)
                    .filter(suffixFilter)
                    .build();

            UpdateStorageLensGroupRequest updateStorageLensGroupRequest =
 UpdateStorageLensGroupRequest.builder()
                    .name(storageLensGroupName)
                    .storageLensGroup(storageLensGroup)
                    .accountId(accountId)
                    .build();

            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
```

```
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            s3ControlClient.updateStorageLensGroup(updateStorageLensGroupRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Untuk contoh konfigurasi JSON, lihat Konfigurasi grup Lensa Penyimpanan.

Mengelola tag AWS sumber daya dengan grup Lensa Penyimpanan

Setiap grup Lensa Penyimpanan Amazon S3 dihitung sebagai AWS sumber daya dengan Nama Sumber Daya Amazon (ARN) miliknya sendiri. Oleh karena itu, Anda mengonfigurasi grup Lensa Penyimpanan, Anda dapat menambahkan tag sumber daya AWS ke grup secara opsional. Anda dapat menambahkan hingga 50 tag per grup Lensa Penyimpanan. Untuk membuat grup Lensa Penyimpanan dengan tag, Anda harus memiliki izin `s3:CreateStorageLensGroup` dan `s3:TagResource`.

Anda dapat menggunakan tag AWS sumber daya untuk mengkategorikan sumber daya menurut departemen, lini bisnis, atau proyek. Langkah ini berguna ketika Anda memiliki banyak sumber daya dengan jenis yang sama. Dengan menerapkan tag, Anda dapat dengan cepat mengidentifikasi grup Lensa Penyimpanan tertentu berdasarkan tag yang telah ditetapkan. Anda juga dapat menggunakan tag untuk melacak dan mengalokasikan biaya.

Selain itu, saat menambahkan tag AWS sumber daya ke grup Lensa Penyimpanan, Anda mengaktifkan kontrol akses berbasis atribut (ABAC). ABAC adalah strategi otorisasi yang mendefinisikan izin berdasarkan atribut, dalam hal ini tag. Anda juga dapat menggunakan kondisi yang menentukan tag sumber daya dalam kebijakan IAM Anda untuk mengontrol akses ke AWS sumber daya.

Anda dapat mengedit kunci dan nilai tanda, dan Anda dapat menghapus tanda dari sumber daya kapan pun. Selain itu, perhatikan batasan berikut:

- Kunci tag dan nilai tag peka huruf besar-kecil.
- Jika Anda menambahkan tanda yang memiliki kunci yang sama dengan tanda yang sudah ada di sumber daya tersebut, nilai baru akan menimpa nilai lama.
- Jika sumber daya dihapus, semua tanda untuk sumber daya tersebut juga akan dihapus.
- Jangan sertakan data pribadi atau sensitif dalam tag AWS sumber daya Anda.
- Tag sistem (dengan kunci tag yang dimulai dengan `aws:`) tidak didukung.
- Panjang setiap kunci tag tidak boleh melebihi 128 karakter. Panjang setiap nilai tag tidak boleh melebihi 256 karakter.

Contoh berikut menunjukkan cara menggunakan tag AWS sumber daya dengan grup Storage Lens.

Topik

- Menambahkan tag AWS sumber daya ke grup Lensa Penyimpanan
- Memperbarui nilai tag grup Lensa Penyimpanan
- Menghapus tag AWS sumber daya dari grup Lensa Penyimpanan
- Mencantumkan tag grup Lensa Penyimpanan

Menambahkan tag AWS sumber daya ke grup Lensa Penyimpanan

Contoh berikut menunjukkan cara menambahkan tag AWS sumber daya ke grup Lensa Penyimpanan Amazon S3. Anda dapat menambahkan tag sumber daya dengan menggunakan konsol Amazon S3, AWS Command Line Interface (AWS CLI), dan. AWS SDK untuk Java

Menggunakan konsol S3

Untuk menambahkan tag AWS sumber daya ke grup Lensa Penyimpanan

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, pilih Grup Lensa Penyimpanan.
3. Di bawah Grup Lensa Penyimpanan, pilih grup Lensa Penyimpanan yang ingin diperbarui.
4. Di bawah tag sumber daya AWS , pilih Tambahkan tag.
5. Di laman Tambahkan tag, tambahkan pasangan nilai-kunci yang baru.

> **Note**
>
> Menambahkan tag baru yang memiliki kunci sama dengan tag yang sudah ada akan menimpa nilai tag sebelumnya.

6. (Opsional) Untuk menambahkan lebih dari satu tag baru, pilih Tambah tag lagi untuk melanjutkan penambahan entri baru. Anda dapat menambahkan hingga 50 tag AWS sumber daya ke grup Lensa Penyimpanan Anda.
7. (Opsional) Jika Anda ingin menghapus entri yang baru ditambahkan, pilih Hapus di bagian samping tag yang ingin dihapus.
8. Pilih Simpan perubahan.

Menggunakan AWS CLI

AWS CLI Perintah contoh berikut menambahkan dua tag sumber daya ke grup Storage Lens yang ada bernama*marketing-department*. Untuk menggunakan perintah contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3control tag-resource --account-id 111122223333 \
--resource-arn arn:aws:s3:us-east-1:111122223333:storage-lens-group/marketing-
department \
--region us-east-1 --tags Key=k1,Value=v1 Key=k2,Value=v2
```

Menggunakan AWS SDK for Java

AWS SDK untuk Java Contoh berikut menambahkan dua tag AWS sumber daya ke grup Lensa Penyimpanan yang ada. Untuk menggunakan contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.Tag;
import software.amazon.awssdk.services.s3control.model.TagResourceRequest;
```

```
public class TagResource {
    public static void main(String[] args) {
        String resourceARN = "Resource_ARN";
        String accountId = "111122223333";

        try {
            Tag resourceTag1 = Tag.builder()
                .key("resource-tag-key-1")
                .value("resource-tag-value-1")
                .build();
            Tag resourceTag2 = Tag.builder()
                    .key("resource-tag-key-2")
                    .value("resource-tag-value-2")
                    .build();
            TagResourceRequest tagResourceRequest = TagResourceRequest.builder()
                    .resourceArn(resourceARN)
                    .tags(resourceTag1, resourceTag2)
                    .accountId(accountId)
                    .build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            s3ControlClient.tagResource(tagResourceRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Memperbarui nilai tag grup Lensa Penyimpanan

Contoh berikut menunjukkan cara memperbarui nilai tag grup Storage Lens dengan menggunakan konsol Amazon S3, AWS Command Line Interface (AWS CLI), dan. AWS SDK untuk Java

Menggunakan konsol S3

Untuk memperbarui tag AWS sumber daya untuk grup Lensa Penyimpanan

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, pilih Grup Lensa Penyimpanan.
3. Di bawah Grup Lensa Penyimpanan, pilih grup Lensa Penyimpanan yang ingin diperbarui.
4. Di bawah tag sumber daya AWS , pilih tag yang ingin diperbarui.
5. Tambahkan nilai tag baru dengan menggunakan kunci yang sama dengan pasangan nilai-kunci yang ingin diperbarui. Pilih ikon tanda centang untuk memperbarui nilai tag.

   > **Note**
   >
   > Menambahkan tag baru yang memiliki kunci sama dengan tag yang sudah ada akan menimpa nilai tag sebelumnya.

6. (Opsional) Jika ingin menambahkan tag baru, pilih Tambahkan tag untuk menambahkan entri baru. Laman Tambah tag akan muncul.

   Anda dapat menambahkan hingga 50 tag AWS sumber daya untuk grup Lensa Penyimpanan Anda. Setelah selesai menambahkan tag baru, pilih Simpan Perubahan.
7. (Opsional) Jika Anda ingin menghapus entri yang baru ditambahkan, pilih Hapus di bagian samping tag yang ingin dihapus. Setelah selesai menghapus tag, pilih Simpan perubahan.

Menggunakan AWS CLI

Contoh AWS CLI perintah berikut memperbarui dua nilai tag untuk grup Storage Lens bernama*marketing-department*. Untuk menggunakan perintah contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3control tag-resource --account-id 111122223333 \
--resource-arn arn:aws:s3:us-east-1:111122223333:storage-lens-group/marketing-
department \
--region us-east-1 --tags Key=k1,Value=v3 Key=k2,Value=v4
```

Menggunakan AWS SDK for Java

AWS SDK untuk Java Contoh berikut memperbarui dua nilai tag grup Storage Lens. Untuk menggunakan contoh ini, ganti *user input placeholders*dengan informasi Anda sendiri.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.Tag;
import software.amazon.awssdk.services.s3control.model.TagResourceRequest;

public class UpdateTagsForResource {
    public static void main(String[] args) {
        String resourceARN = "Resource_ARN";
        String accountId = "111122223333";

        try {
            Tag updatedResourceTag1 = Tag.builder()
                .key("resource-tag-key-1")
                .value("resource-tag-updated-value-1")
                .build();
            Tag updatedResourceTag2 = Tag.builder()
                    .key("resource-tag-key-2")
                    .value("resource-tag-updated-value-2")
                    .build();
            TagResourceRequest tagResourceRequest = TagResourceRequest.builder()
                    .resourceArn(resourceARN)
                    .tags(updatedResourceTag1, updatedResourceTag2)
                    .accountId(accountId)
                    .build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            s3ControlClient.tagResource(tagResourceRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
```

```
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Menghapus tag AWS sumber daya dari grup Lensa Penyimpanan

Contoh berikut menunjukkan cara menghapus tag AWS sumber daya dari grup Lensa Penyimpanan. Anda dapat menghapus tag dengan menggunakan konsol Amazon S3, AWS Command Line Interface (AWS CLI), dan. AWS SDK untuk Java

Menggunakan konsol S3

Untuk menghapus tag AWS sumber daya dari grup Lensa Penyimpanan

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, pilih Grup Lensa Penyimpanan.
3. Di bawah Grup Lensa Penyimpanan, pilih grup Lensa Penyimpanan yang ingin diperbarui.
4. Pada tag sumber daya AWS , pilih pasangan kunci-nilai yang ingin dihapus.
5. Pilih Hapus. Kotak dialog Hapus tag sumber daya AWS akan muncul.

> **Note**
>
> Jika tag digunakan untuk mengontrol akses, melanjutkan tindakan ini dapat memengaruhi sumber daya terkait. Setelah Anda menghapus tag secara permanen, tag tersebut tidak dapat dipulihkan.

6. Pilih Hapus untuk menghapus pasangan nilai-kunci secara permanen.

Menggunakan AWS CLI

AWS CLI Perintah berikut menghapus dua tag AWS sumber daya dari grup Lensa Penyimpanan yang ada: Untuk menggunakan perintah contoh ini, ganti *`user input placeholders`* dengan informasi Anda sendiri.

```
aws s3control untag-resource --account-id 111122223333 \
```

```
--resource-arn arn:aws:s3:us-east-1:111122223333:storage-lens-group/Marketing-
Department \
--region us-east-1 --tag-keys k1 k2
```

Menggunakan AWS SDK for Java

AWS SDK untuk Java Contoh berikut menghapus dua tag AWS sumber daya dari grup Lensa Penyimpanan Amazon Resource Name (ARN) yang Anda tentukan di akun. *111122223333* Untuk menggunakan contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.UntagResourceRequest;

public class UntagResource {
    public static void main(String[] args) {
        String resourceARN = "Resource_ARN";
        String accountId = "111122223333";

        try {
            String tagKey1 = "resource-tag-key-1";
            String tagKey2 = "resource-tag-key-2";
            UntagResourceRequest untagResourceRequest = UntagResourceRequest.builder()
                    .resourceArn(resourceARN)
                    .tagKeys(tagKey1, tagKey2)
                    .accountId(accountId)
                    .build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            s3ControlClient.untagResource(untagResourceRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
```

```
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Mencantumkan tag grup Lensa Penyimpanan

Contoh berikut menunjukkan cara membuat daftar tag AWS sumber daya yang terkait dengan grup Lensa Penyimpanan. Anda dapat mencantumkan tag dengan menggunakan konsol Amazon S3, AWS Command Line Interface (AWS CLI), dan. AWS SDK untuk Java

Menggunakan konsol S3

Untuk meninjau daftar tag dan nilai tag untuk grup Lensa Penyimpanan

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, pilih Grup Lensa Penyimpanan.
3. Di bawah Grup Lensa Penyimpanan, pilih grup Lensa Penyimpanan yang Anda minati.
4. Gulir ke bawah ke bagian tag sumber daya AWS . Semua tag AWS sumber daya yang ditentukan pengguna yang ditambahkan ke grup Lensa Penyimpanan Anda terdaftar bersama dengan nilai tag mereka.

Menggunakan AWS CLI

AWS CLI Contoh perintah berikut mencantumkan semua nilai tag grup Storage Lens untuk grup Storage Lens bernama*marketing-department*. Untuk menggunakan perintah contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3control list-tags-for-resource --account-id 111122223333 \
--resource-arn arn:aws:s3:us-east-1:111122223333:storage-lens-group/marketing-
department \
--region us-east-1
```

Menggunakan AWS SDK for Java

AWS SDK untuk Java Contoh berikut mencantumkan nilai tag grup Lensa Penyimpanan untuk grup Lensa Penyimpanan Amazon Resource Name (ARN) yang Anda tentukan. Untuk menggunakan contoh ini, ganti *user input placeholders*dengan informasi Anda sendiri.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.ListTagsForResourceRequest;
import software.amazon.awssdk.services.s3control.model.ListTagsForResourceResponse;

public class ListTagsForResource {
    public static void main(String[] args) {
        String resourceARN = "Resource_ARN";
        String accountId = "111122223333";

        try {
            ListTagsForResourceRequest listTagsForResourceRequest =
 ListTagsForResourceRequest.builder()
                    .resourceArn(resourceARN)
                    .accountId(accountId)
                    .build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            ListTagsForResourceResponse response =
 s3ControlClient.listTagsForResource(listTagsForResourceRequest);
            System.out.println(response);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Mencantumkan semua grup Lensa Penyimpanan

Contoh berikut menunjukkan cara mencantumkan semua grup Lensa Penyimpanan Amazon S3 di Akun AWS dan Wilayah asal. Contoh-contoh ini menunjukkan caranya daftar semua grup Storage Lens dengan menggunakan konsol Amazon S3, AWS Command Line Interface (AWS CLI), dan. AWS SDK untuk Java

Menggunakan konsol S3

Untuk mencantumkan semua grup Lensa Penyimpanan di akun dan Wilayah asal

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, pilih Grup Lensa Penyimpanan.
3. Di bawah Grup Lensa Penyimpanan, daftar grup Lensa Penyimpanan di akun Anda akan ditampilkan.

Menggunakan AWS CLI

AWS CLI Contoh berikut mencantumkan semua grup Lensa Penyimpanan untuk akun Anda. Untuk menggunakan perintah contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3control list-storage-lens-groups --account-id 111122223333 \
--region us-east-1
```

Menggunakan AWS SDK for Java

AWS SDK untuk Java Contoh berikut mencantumkan grup Storage Lens untuk akun*111122223333*. Untuk menggunakan contoh ini, ganti *user input placeholders*dengan informasi Anda sendiri.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.ListStorageLensGroupsRequest;
```

```
import software.amazon.awssdk.services.s3control.model.ListStorageLensGroupsResponse;

public class ListStorageLensGroups {
    public static void main(String[] args) {
        String accountId = "111122223333";

        try {
            ListStorageLensGroupsRequest listStorageLensGroupsRequest =
 ListStorageLensGroupsRequest.builder()
                    .accountId(accountId)
                    .build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            ListStorageLensGroupsResponse response =
 s3ControlClient.listStorageLensGroups(listStorageLensGroupsRequest);
            System.out.println(response);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Melihat detail grup Lensa Penyimpanan

Contoh berikut ini menunjukkan cara melihat detail konfigurasi grup Lensa Penyimpanan Amazon S3. Anda dapat melihat detail ini dengan menggunakan konsol Amazon S3, AWS Command Line Interface (AWS CLI), dan. AWS SDK untuk Java

Menggunakan konsol S3

Untuk melihat detail konfigurasi grup Lensa Penyimpanan

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

2. Di panel navigasi bagian kiri, pilih Grup Lensa Penyimpanan.
3. Di bawah Grup Lensa Penyimpanan, pilih tombol opsi di bagian samping grup Lensa Penyimpanan yang Anda minati.
4. Pilih Lihat detail. Anda sekarang dapat melihat detail grup Lensa Penyimpanan Anda.

Menggunakan AWS CLI

AWS CLI Contoh berikut mengembalikan detail konfigurasi untuk grup Storage Lens. Untuk menggunakan perintah contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3control get-storage-lens-group --account-id 111122223333 \
--region us-east-1 --name marketing-department
```

Menggunakan AWS SDK for Java

AWS SDK untuk Java Contoh berikut mengembalikan detail konfigurasi untuk grup Storage Lens bernama *Marketing-Department* dalam akun*111122223333*. Untuk menggunakan contoh ini, ganti *user input placeholders*dengan informasi Anda sendiri.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.GetStorageLensGroupRequest;
import software.amazon.awssdk.services.s3control.model.GetStorageLensGroupResponse;

public class GetStorageLensGroup {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            GetStorageLensGroupRequest getRequest =
 GetStorageLensGroupRequest.builder()
                    .name(storageLensGroupName)
                    .accountId(accountId).build();
```

```
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            GetStorageLensGroupResponse response =
 s3ControlClient.getStorageLensGroup(getRequest);
            System.out.println(response);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Menghapus grup Lensa Penyimpanan

Contoh berikut menunjukkan cara menghapus grup Lensa Penyimpanan Amazon S3 dengan menggunakan konsol Amazon S3 AWS Command Line Interface ,AWS CLI(), dan. AWS SDK untuk Java

Menggunakan konsol S3

Untuk menghapus grup Lensa Penyimpanan

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi bagian kiri, pilih Grup Lensa Penyimpanan.
3. Di bawah Grup Lensa Penyimpanan, pilih tombol opsi di bagian samping grup Lensa Penyimpanan yang ingin dihapus.
4. Pilih Hapus. Kotak dialog Hapus grup Lensa Penyimpanan akan ditampilkan.
5. Pilih Hapus lagi untuk menghapus grup Lensa Penyimpanan secara permanen.

> **Note**
>
> Setelah grup Lensa Penyimpanan dihapus, grup tersebut tidak dapat dipulihkan.

Menggunakan AWS CLI

AWS CLI Contoh berikut menghapus grup Storage Lens bernama*marketing-department*. Untuk menggunakan perintah contoh ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws s3control delete-storage-lens-group --account-id 111122223333 \
--region us-east-1 --name marketing-department
```

Menggunakan AWS SDK for Java

AWS SDK untuk Java Contoh berikut menghapus grup Storage Lens bernama *Marketing-Department* dalam akun*111122223333*. Untuk menggunakan contoh ini, ganti *user input placeholders*dengan informasi Anda sendiri.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.DeleteStorageLensGroupRequest;

public class DeleteStorageLensGroup {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            DeleteStorageLensGroupRequest deleteStorageLensGroupRequest =
 DeleteStorageLensGroupRequest.builder()
                    .name(storageLensGroupName)
                    .accountId(accountId).build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            s3ControlClient.deleteStorageLensGroup(deleteStorageLensGroupRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
```

```
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Membuat katalog dan menganalisis data Anda dengan S3 Inventory

Gunakan Inventaris Amazon S3 untuk membantu mengelola penyimpanan. Misalnya, Anda dapat menggunakannya untuk mengaudit dan melaporkan replikasi dan status enkripsi objek untuk kebutuhan bisnis, kepatuhan, dan regulasi. Anda juga dapat menyederhanakan dan mempercepat alur kerja bisnis dan pekerjaan big data dengan menggunakan Inventaris Amazon S3, yang menyediakan alternatif terjadwal untuk operasi API `List` asinkron Amazon S3. Inventaris Amazon S3 tidak menggunakan operasi `List` API untuk mengaudit objek dan tidak memengaruhi tingkat permintaan bucket.

Inventaris Amazon S3 menyediakan nilai yang dipisahkan koma (CSV), Apache baris kolom yang dioptimalkan (ORC) atau file keluaran Apache Parquet yang mencantumkan objek dan metadata terkaitnya setiap hari atau setiap minggu untuk bucket S3 atau objek dengan awalan bersama (yaitu, objek yang memiliki nama yang dimulai dengan string yang sama). Jika Anda memulai inventaris mingguan, laporan akan dibuat setiap hari Minggu (zona waktu UTC) setelah laporan awal. Untuk informasi tentang ketentuan harga inventaris Amazon S3, lihat Harga Amazon S3.

Anda dapat mengonfigurasi beberapa daftar inventaris untuk satu bucket. Saat mengonfigurasi daftar inventaris, Anda dapat menentukan hal-hal berikut:

- Metadata objek yang akan disertakan di inventaris
- Mencantumkan semua versi objek atau hanya versi saat ini
- Tempat penyimpanan output file daftar inventaris
- Membuat inventaris harian atau mingguan
- Mengenkripsi file daftar inventaris atau tidak

Anda dapat melakukan kueri Inventaris Amazon S3 dengan kueri SQL standar dengan menggunakan Amazon Athena, Amazon Redshift Spectrum, dan alat lainnya, seperti Presto, Apache Hive, dan

Apache Spark. Untuk informasi selengkapnya tentang penggunaan Athena untuk mengkueri file inventaris, lihat the section called “Membuat kueri inventaris dengan Athena”.

> **Note**
>
> Amazon S3 akan membutuhkan waktu hingga 48 jam untuk mengirimkan laporan inventaris pertama.

Bucket sumber dan tujuan

Bucket yang daftar objeknya dibuat oleh inventaris disebut bucket sumber. Bucket tempat menyimpan file daftar inventaris disebut bucket tujuan.

Bucket sumber

Inventaris akan mencantumkan objek yang disimpan di dalam bucket sumber. Anda bisa mendapatkan daftar inventaris untuk seluruh bucket, atau memfilter daftar tersebut berdasarkan awalan nama kunci objek.

Bucket sumber:

- Berisi objek yang tercantum di dalam inventaris
- Berisi konfigurasi untuk inventaris

Bucket tujuan

File daftar inventaris Amazon S3 akan ditulis ke bucket tujuan. Untuk mengelompokkan semua file daftar inventaris di lokasi yang sama di dalam bucket tujuan, Anda dapat menentukan awalan tujuan di konfigurasi inventaris.

Bucket tujuan:

- Berisi daftar file inventaris.
- Berisi file manifes yang mencantumkan semua file daftar inventaris yang disimpan di dalam bucket tujuan. Untuk informasi selengkapnya, lihat Manifes inventaris.
- Harus memiliki kebijakan bucket agar dapat memberikan izin ke Amazon S3 untuk memverifikasi kepemilikan bucket dan izin untuk menulis file ke bucket.

- Harus Wilayah AWS sama dengan ember sumber.
- Bisa sama dengan bucket sumber.
- Dapat dimiliki oleh akun yang berbeda Akun AWS dari akun yang memiliki bucket sumber.

Daftar Inventaris Amazon S3

File daftar inventaris berisi daftar objek di dalam bucket sumber dan metadata untuk setiap objek. File daftar inventaris disimpan di dalam bucket tujuan dengan salah satu format berikut:

- Sebagai file CSV yang dikompresi dengan GZIP
- Sebagai file kolom baris (ORC) yang dioptimalkan Apache dan dikompresi dengan ZLIB
- Sebagai file Apache Parquet yang dikompresi dengan Snappy

> **Note**
>
> Objek di laporan Inventaris Amazon S3 tidak dijamin akan diurutkan dengan urutan tertentu.

File daftar inventaris berisi daftar objek di dalam bucket sumber dan metadata untuk setiap objek yang terdaftar:

- Nama bucket–Nama dari bucket yang digunakan untuk menyimpan inventaris.
- Nama kunci–Nama kunci objek (atau kunci) yang mengidentifikasi objek secara unik di dalam bucket. Saat Anda menggunakan format file CSV, nama kuncinya dikodekan dengan URL dan harus didekodekan sebelum dapat digunakan.
- ID Versi–ID versi objek. Saat Anda mengaktifkan Penentuan Versi di bucket, Amazon S3 akan menetapkan nomor versi untuk objek yang ditambahkan ke dalam bucket. Untuk informasi selengkapnya, lihat Mempertahankan beberapa versi objek dengan S3 Versioning. (Bidang ini tidak disertakan jika daftar tersebut dikonfigurasi hanya untuk versi objek saat ini.)
- IsLatest – Diatur ke `True` jika objek adalah versi objek saat ini. (Bidang ini tidak disertakan jika daftar tersebut dikonfigurasi hanya untuk versi objek saat ini.)
- Delete marker–Diatur ke `True`, jika objeknya adalah delete marker. Untuk informasi selengkapnya, lihat Mempertahankan beberapa versi objek dengan S3 Versioning. (Bidang ini secara otomatis akan ditambahkan ke laporan jika Anda mengonfigurasi laporan untuk menyertakan semua versi objek).

- Ukuran–Ukuran objek dalam byte, tidak termasuk ukuran unggahan multibagian yang tidak lengkap, metadata objek, dan penanda hapus.
- Tanggal terakhir diubah–Tanggal pembuatan objek atau tanggal modifikasi terakhir, mana pun yang terbaru.
- ETagTag entitas (ETag) adalah hash dari objek. ETag Merefleksikan perubahan hanya pada isi objek, bukan metadatanya. Itu ETag bisa menjadi MD5 intisari dari data objek. Ketentuannya tergantung pada bagaimana objek dibuat dan bagaimana objek dienkripsi. Untuk informasi selengkapnya, lihat Object dalam Referensi API Amazon Simple Storage Service.
- Kelas penyimpanan–Kelas penyimpanan yang digunakan untuk menyimpan objek. Atur ke `STANDARD`, `REDUCED_REDUNDANCY`, `STANDARD_IA`, `ONEZONE_IA`, `INTELLIGENT_TIERING`, `GLACIER`, `DEEP_ARCHIVE`, `OUTPOSTS`, `GLACIER_IR`, or `SNOW`. Untuk informasi selengkapnya, lihat Memahami dan mengelola kelas penyimpanan Amazon S3.

> **Note**
>
> S3 Inventory tidak mendukung S3 Express One Zone.

- Flag unggahan multibagian–Atur ke `True` jika objek diunggah sebagai unggahan multibagian. Untuk informasi selengkapnya, lihat Mengunggah dan menyalin objek menggunakan unggahan multipart di Amazon S3.
- Status replikasi–Atur ke `PENDING`, `COMPLETED`, `FAILED`, atau `REPLICA`. Untuk informasi selengkapnya, lihat Mendapatkan informasi status replikasi.
- Status enkripsi — Status enkripsi sisi server, tergantung pada jenis kunci enkripsi yang digunakan — enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3), enkripsi sisi server dengan kunci () (SSE-KMS), enkripsi sisi server dua lapis dengan kunci AWS KMS(DSSE-KMS), atau enkripsi sisi server AWS Key Management Service dengan kunci yang disediakan pelanggan (SSE-C). AWS KMS Atur ke `SSE-S3`, `SSE-KMS`, `DSSE-KMS`, `SSE-C`, atau `NOT-SSE`. Status `NOT-SSE` mengindikasikan bahwa objek tersebut tidak dienkripsi dengan enkripsi di sisi server. Untuk informasi selengkapnya, lihat Melindungi data dengan enkripsi.
- Kunci Objek S3 bertahan hingga tanggal–Tanggal sampai objek yang terkunci tidak dapat dihapus. Untuk informasi selengkapnya, lihat Mengunci objek dengan Object Lock.
- Mode retensi Kunci Objek S3–Atur ke `Governance` atau `Compliance` untuk objek yang terkunci. Untuk informasi selengkapnya, lihat Mengunci objek dengan Object Lock.

- Status penahanan hukum Kunci Objek S3 –Atur ke `On` jika penahanan hukum sudah diterapkan pada suatu objek. Jika belum, atur statusnya ke `Off`. Untuk informasi selengkapnya, lihat Mengunci objek dengan Object Lock.
- Tingkat akses S3 Intelligent-Tiering–Tingkat akses (sering atau jarang) dari suatu objek jika disimpan di kelas penyimpanan S3 Intelligent-Tiering. Atur ke `FREQUENT`, `INFREQUENT`, `ARCHIVE_INSTANT_ACCESS`, `ARCHIVE`, atau `DEEP_ARCHIVE`. Untuk informasi selengkapnya, lihat Kelas penyimpanan untuk mengoptimalkan data secara otomatis dengan pola akses yang berubah atau tidak diketahui.
- Status Kunci Bucket S3–Atur ke `ENABLED` atau `DISABLED`. Menunjukkan apakah suatu objek menggunakan Kunci Bucket S3 untuk SSE-KMS. Untuk informasi selengkapnya, lihat Menggunakan Kunci Bucket Amazon S3.
- Algoritma checksum–Menunjukkan algoritma yang digunakan untuk membuat checksum untuk objek. Untuk informasi selengkapnya, lihat Menggunakan algoritma checksum yang didukung.
- Daftar kontrol akses objek — Daftar kontrol akses (ACL) untuk setiap objek yang mendefinisikan Akun AWS atau kelompok mana yang diberikan akses ke objek ini dan jenis akses yang diberikan. Bidang ACL Objek ditentukan dalam format JSON. Laporan Inventaris S3 menyertakan ACLs yang terkait dengan objek di bucket sumber Anda, bahkan saat bucket ACLs dinonaktifkan. Untuk informasi selengkapnya, lihat Menggunakan bidang ACL Objek dan Gambaran umum daftar kontrol akses (ACL).

> **Note**
>
> Bidang ACL Objek ditentukan dalam format JSON. Laporan inventaris akan menampilkan nilai untuk bidang Object ACL sebagai string yang dikodekan base64.
> Misalnya, katakanlah Anda memiliki ACL Objek berikut dalam format JSON:

```
{
        "version": "2022-11-10",
        "status": "AVAILABLE",
        "grants": [{
            "canonicalId": "example-canonical-user-ID",
            "type": "CanonicalUser",
            "permission": "READ"
        }]
}
```

Bidang ACL Objek dikodekan dan ditampilkan sebagai string yang dikodekan dengan base64 berikut ini:

```
eyJ2ZXJzaW9uIjoiMjAyMi0xMS0xMCIsInN0YXR1cyI6IkFWQUlMQUJMRSIsImdyYW50cyI6W3siY2Fub25pY2Fs
```

Untuk mendapatkan nilai yang terdekripsi dalam JSON untuk bidang ACL Objek, Anda dapat melakukan kueri pada bidang ini di Amazon Athena. Untuk contoh kueri, lihat Membuat kueri inventaris Amazon S3 dengan Amazon Athena.

- Pemilik objek — ID pengguna kanonik dari pemilik objek. Untuk informasi selengkapnya, lihat Menemukan ID pengguna kanonik untuk AWS akun Anda di Panduan Referensi Manajemen AWS Akun.

> **Note**
>
> Saat objek mencapai akhir masa pakai berdasarkan konfigurasi siklus hidupnya, Amazon S3 akan mengantre objek untuk dihapus dan menghapusnya tidak secara bersamaan. Oleh karena itu, penundaan antara tanggal kedaluwarsa dan tanggal saat Amazon S3 menghapus objek mungkin ditemukan. Laporan inventaris mencakup semua objek yang kedaluwarsa tetapi belum dihapus. Untuk informasi selengkapnya tentang tindakan kedaluwarsa di Siklus Hidup S3, lihat Mengakhiri objek.

Berikut ini adalah contoh laporan inventaris dengan bidang metadata tambahan yang terdiri dari empat catatan.

```
amzn-s3-demo-bucket1    example-object-1    EXAMPLEDC8l.XJCENlF7LePaNIIvs001    TRUE
     1500    2024-08-15T15:28:26.0004    EXAMPLE21e1518b92f3d92773570f600    STANDARD
  FALSE    COMPLETED    SSE-KMS    2025-01-25T15:28:26.000Z    COMPLIANCE    Off
 ENABLED
 eyJ2ZXJzaW9uIjoiMjAyMi0xMS0xMCIsInN0YXR1cyI6IkFWQUlMQUJMRSIsImdyYW50cyI6W3sicGVybWlzc2lvbiI6Ik
    EXAMPLE766e8f6b115d93d41df2eac420aa4a465d177c1398bc6a088c76b7000
amzn-s3-demo-bucket1    example-object-2    EXAMPLEDC8l.XJCENlF7LePaNIIvs002
 TRUE        200    2024-08-21T15:28:26.000Z    EXAMPLE21e1518b92f3d92773570f601
  INTELLIGENT_TIERING    FALSE    COMPLETED    SSE-KMS    2025-01-25T15:28:26.000Z
    COMPLIANCE    Off    INFREQUENT    ENABLED    SHA-256
 eyJ2ZXJzaW9uIjoiMjAyMi0xMS0xMCIsInN0YXR1cyI6IkFWQUlMQUJMRSIsImdyYW50cyI6W3sicGVybWlzc2lvbiI6Ik
    EXAMPLE766e8f6b115d93d41df2eac420aa4a465d177c1398bc6a088c76b7001
```

```
amzn-s3-demo-bucket1    example-object-3    EXAMPLEDC8l.XJCENlF7LePaNIIvs003    TRUE
      12500    2023-01-15T15:28:30.000Z    EXAMPLE21e1518b92f3d92773570f602    STANDARD
    FALSE    REPLICA    SSE-KMS    2025-01-25T15:28:26.000Z    GOVERNANCE    On
 ENABLED
 eyJ2ZXJzaW9uIjoiMjAyMi0xMS0xMCIsInN0YXR1cyI6IkFWQUlMQUJMRSIsImdyYW50cyI6W3sicGVybWlzc2lvbiI6Ik
    EXAMPLE766e8f6b115d93d41df2eac420aa4a465d177c1398bc6a088c76b7002
amzn-s3-demo-bucket1    example-object-4    EXAMPLEDC8l.XJCENlF7LePaNIIvs004    TRUE
      100    2021-02-15T15:28:27.000Z    EXAMPLE21e1518b92f3d92773570f603    STANDARD
  FALSE    COMPLETED    SSE-KMS    2025-01-25T15:28:26.000Z    COMPLIANCE    Off
 ENABLED
 eyJ2ZXJzaW9uIjoiMjAyMi0xMS0xMCIsInN0YXR1cyI6IkFWQUlMQUJMRSIsImdyYW50cyI6W3sicGVybWlzc2lvbiI6Ik
    EXAMPLE766e8f6b115d93d41df2eac420aa4a465d177c1398bc6a088c76b7003
```

Anda sebaiknya membuat kebijakan siklus hidup yang menghapus daftar inventaris lama. Untuk informasi selengkapnya, lihat Mengelola siklus hidup objek.

Izin `s3:PutInventoryConfiguration` memungkinkan pengguna untuk memilih semua bidang metadata yang tercantum sebelumnya untuk setiap objek saat mengonfigurasi daftar inventaris dan menentukan bucket tujuan untuk menyimpan inventaris. Pengguna dengan akses baca ke objek di bucket tujuan dapat mengakses semua bidang metadata objek yang tersedia di daftar inventaris. Untuk membatasi akses ke laporan inventaris, lihat Berikan izin untuk Inventaris S3 dan analitik S3.

Konsistensi inventaris

Semua objek mungkin tidak akan muncul di setiap daftar inventaris. Daftar inventaris akan memberikan konsistensi akhir untuk permintaan (objek baru dan penimpaan) PUT dan permintaan `DELETE`. Setiap daftar inventaris untuk bucket berisi cuplikan item bucket. Daftar inventari ini pada akhirnya konsisten (maksudnya, daftar mungkin tidak akan menyertakan objek yang baru ditambahkan atau dihapus).

Untuk memvalidasi status objek sebelum Anda mengambil tindakan untuk objek tertentu, sebaiknya lakukan permintaan API REST `HeadObject` untuk mengambil metadata untuk objek tersebut, atau memeriksa properti objeknya di konsol Amazon S3. Anda juga dapat memeriksa metadata objek dengan AWS CLI atau SDKS. AWS Untuk informasi lebih lanjut, lihat `HeadObject` di dalam Referensi API Layanan Penyimpanan Sederhana Amazon.

Untuk informasi selengkapnya tentang bekerja dengan Inventaris Amazon S3, lihat topik-topik berikut.

Topik

- Mengonfigurasi Inventaris Amazon S3

- Menemukan daftar inventaris
- Menyiapkan Notifikasi Peristiwa Amazon S3 untuk penyelesaian inventaris
- Membuat kueri inventaris Amazon S3 dengan Amazon Athena
- Mengonversi string ID versi kosong di laporan Inventaris Amazon S3 menjadi string null
- Menggunakan bidang ACL Objek

Mengonfigurasi Inventaris Amazon S3

Inventaris Amazon S3 menyediakan daftar file datar dari objek dan metadata, sesuai jadwal yang Anda tentukan. Anda dapat menggunakan Inventaris S3 sebagai alternatif terjadwal untuk operasi API `List` sinkron Amazon S3. Inventaris S3 menyediakan nilai yang dipisahkan koma (CSV), kolom baris yang dioptimalkan Apache (ORC), atau file output Apache Parquet (Parquet) yang mencantumkan objek Anda dan metadata yang sesuai.

Anda dapat mengonfigurasi Inventaris S3 untuk membuat daftar inventaris setiap hari atau setiap minggu untuk bucket S3 atau untuk objek yang memiliki prefiks yang sama (objek yang memiliki nama yang diawali dengan string yang sama). Untuk informasi selengkapnya, lihat Membuat katalog dan menganalisis data Anda dengan S3 Inventory.

Bagian ini menjelaskan cara mengonfigurasi inventaris, termasuk detail tentang sumber inventaris dan bucket tujuan.

Topik

- Gambaran Umum
- Membuat kebijakan bucket tujuan
- Memberikan izin kepada Amazon S3 untuk menggunakan CMK untuk enkripsi
- Mengonfigurasi inventaris dengan menggunakan konsol S3
- Penggunaan API REST untuk bekerja dengan Inventaris S3

Gambaran Umum

Inventaris Amazon S3 membantu Anda mengelola penyimpanan dengan membuat daftar objek dalam bucket S3 sesuai jadwal yang telah ditentukan. Anda dapat mengonfigurasi beberapa daftar inventaris untuk satu bucket. Daftar inventaris diterbitkan dalam format file CSV, ORC, atau file Parquet di bucket tujuan.

Cara termudah untuk menyiapkan inventaris adalah dengan menggunakan konsol Amazon S3, tetapi Anda juga dapat menggunakan Amazon S3 REST API AWS Command Line Interface ,AWS CLI(), atau. AWS SDKs Konsol menjalankan langkah pertama dari prosedur berikut ini untuk Anda: menambahkan kebijakan bucket ke bucket tujuan.

Untuk menyiapkan Inventaris Amazon S3 untuk bucket S3

1. Tambahkan kebijakan bucket untuk bucket tujuan.

   Anda harus membuat kebijakan bucket di bucket tujuan yang memberikan izin ke Amazon S3 untuk menulis objek ke bucket di lokasi yang ditentukan. Untuk contoh kebijakan, lihat Berikan izin untuk Inventaris S3 dan analitik S3.

2. Konfigurasikan inventaris untuk mencantumkan objek dalam bucket sumber dan terbitkan daftar tersebut ke bucket tujuan.

   Ketika mengkonfigurasi daftar inventaris untuk bucket sumber, Anda perlu menentukan bucket tujuan untuk menyimpan daftar tersebut, dan apakah Anda ingin membuat daftar tersebut secara harian atau mingguan. Anda juga dapat mengonfigurasi apakah akan mencantumkan semua versi objek atau hanya versi saat ini dan metadata objek apa yang akan disertakan.

   Beberapa bidang metadata objek dalam konfigurasi laporan Inventaris S3 bersifat opsional, artinya tersedia secara default tetapi dapat dibatasi saat Anda memberikan izin kepada pengguna. `s3:PutInventoryConfiguration` Anda dapat mengontrol apakah pengguna dapat menyertakan bidang metadata opsional ini dalam laporan mereka dengan menggunakan kunci `s3:InventoryAccessibleOptionalFields` kondisi.

   Untuk informasi selengkapnya tentang bidang metadata opsional yang tersedia di Inventaris S3, lihat di Referensi OptionalFieldsAPI Amazon Simple Storage Service. Untuk informasi selengkapnya tentang membatasi akses ke bidang metadata opsional tertentu dalam konfigurasi inventaris, lihat. Kontrol pembuatan konfigurasi laporan Inventaris S3

   Anda dapat menentukan bahwa file daftar inventaris dienkripsi menggunakan enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3) atau () kunci terkelola pelanggan (SSE-KMS). AWS Key Management Service AWS KMS

> **Note**
>
> The Kunci yang dikelola AWS (`aws/s3`) tidak didukung untuk enkripsi SSE-KMS dengan S3 Inventory.

Untuk informasi selengkapnya tentang SSE-S3 dan SSE-KMS, lihat Melindungi data dengan enkripsi di sisi klien. Jika Anda berencana menggunakan enkripsi SSE-KMS, lihat Langkah 3.

- Untuk informasi tentang cara menggunakan konsol untuk mengonfigurasi daftar inventaris, lihat Mengonfigurasi inventaris dengan menggunakan konsol S3.
- Untuk menggunakan Amazon S3 API untuk mengonfigurasi daftar inventaris, gunakan operasi PutBucketInventoryConfigurationREST API atau yang setara dari atau. AWS CLI AWS SDKs

3. Untuk mengenkripsi file daftar inventaris dengan SSE-KMS, berikan izin kepada Amazon S3 untuk menggunakan AWS KMS key.

   Anda dapat mengonfigurasi enkripsi untuk file daftar inventaris dengan menggunakan konsol Amazon S3, Amazon S3 REST API AWS CLI, atau. AWS SDKs Apa pun cara yang dipilih, Anda harus memberikan izin kepada Amazon S3 untuk menggunakan CMK agar dapat mengenkripsi file inventaris. Berikan izin kepada Amazon S3 dengan memodifikasi kebijakan kunci untuk CMK yang ingin Anda gunakan untuk mengenkripsi file inventaris. Untuk informasi selengkapnya, lihat Memberikan izin kepada Amazon S3 untuk menggunakan CMK untuk enkripsi.

   Bucket tujuan yang menyimpan file daftar inventaris dapat dimiliki oleh Akun AWS yang berbeda dari akun yang memiliki bucket sumber. Jika Anda menggunakan enkripsi SSE-KMS untuk operasi lintas akun Inventaris Amazon S3, sebaiknya gunakan ARN kunci KMS yang memenuhi syarat sepenuhnya saat mengonfigurasi inventaris S3. Untuk informasi selengkapnya, lihat Menggunakan enkripsi SSE-KMS untuk operasi lintas akun dan ServerSideEncryptionByDefault di Referensi API Amazon Simple Storage Service.

Membuat kebijakan bucket tujuan

Jika membuat konfigurasi inventaris melalui konsol Amazon S3, Amazon S3 secara otomatis membuat kebijakan bucket pada bucket tujuan yang memberikan izin tulis ke bucket tersebut kepada Amazon S3. Namun, jika Anda membuat konfigurasi inventaris melalui AWS CLI, AWS SDKs, atau Amazon S3 REST API, Anda harus menambahkan kebijakan bucket secara manual di bucket tujuan.

Kebijakan bucket tujuan Inventaris S3 memungkinkan Amazon S3 menulis data untuk laporan inventaris ke bucket.

Berikut ini adalah contoh kebijakan bucket.

```
{
      "Version": "2012-10-17",
      "Statement": [
        {
            "Sid": "InventoryExamplePolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-DESTINATION-BUCKET/*"
            ],
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET"
                },
                "StringEquals": {
                    "aws:SourceAccount": "source-account-id",
                    "s3:x-amz-acl": "bucket-owner-full-control"
                }
            }
        }
    ]
}
```

Untuk informasi selengkapnya, lihat Berikan izin untuk Inventaris S3 dan analitik S3.

Jika terjadi kesalahan saat mencoba membuat kebijakan bucket, Anda akan diberi petunjuk tentang cara memperbaikinya. Misalnya, jika Anda memilih bucket tujuan di bucket lain Akun AWS dan tidak memiliki izin untuk membaca dan menulis ke kebijakan bucket, Anda akan melihat pesan galat.

Dalam hal ini, pemilik bucket tujuan harus menambahkan kebijakan bucket ke bucket tujuan. Jika kebijakan tidak ditambahkan ke bucket tujuan, Anda tidak akan mendapatkan laporan inventaris karena Amazon S3 tidak memiliki izin untuk menulis ke bucket tujuan. Jika bucket sumber dimiliki oleh akun yang berbeda dengan akun pengguna saat ini, ID akun yang benar dari pemilik bucket sumber harus diganti dalam kebijakan.

> **Note**
>
> Pastikan tidak ada pernyataan Deny yang ditambahkan ke kebijakan bucket tujuan yang akan mencegah pengiriman laporan inventaris ke dalam bucket ini. Untuk informasi selengkapnya, lihat Mengapa saya tidak dapat membuat Laporan Inventaris Amazon S3? .

Memberikan izin kepada Amazon S3 untuk menggunakan CMK untuk enkripsi

Untuk memberikan izin Amazon S3 untuk menggunakan AWS Key Management Service (AWS KMS) kunci terkelola pelanggan untuk enkripsi sisi server, Anda harus menggunakan kebijakan kunci. Untuk memperbarui kebijakan kunci agar Anda dapat menggunakan CMK, gunakan prosedur berikut ini.

Untuk memberikan izin Amazon S3 untuk mengenkripsi dengan menggunakan kunci terkelola pelanggan

1. Menggunakan Akun AWS yang memiliki kunci yang dikelola pelanggan, masuk ke. AWS Management Console
2. Buka AWS KMS konsol di https://console.aws.amazon.com/kms.
3. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
4. Di panel navigasi kiri, pilih CMK.
5. Di bawah CMK, pilih kunci yang dikelola pelanggan yang ingin Anda gunakan untuk mengenkripsi file inventaris.
6. Di bagian Kebijakan Kunci, pilih Beralih ke tampilan kebijakan.
7. Untuk memperbarui kebijakan kunci, pilih Edit.
8. Pada halaman Edit kebijakan kunci, tambahkan baris berikut ke kebijakan kunci yang ada. Untuk *source-account-id* dan *amzn-s3-demo-source-bucket*, berikan nilai yang sesuai untuk kasus penggunaan Anda.

```
{
    "Sid": "Allow Amazon S3 use of the customer managed key",
    "Effect": "Allow",
    "Principal": {
        "Service": "s3.amazonaws.com"
    },
    "Action": [
```

```
            "kms:GenerateDataKey"
        ],
        "Resource": "*",
        "Condition":{
          "StringEquals":{
             "aws:SourceAccount":"source-account-id"
         },
          "ArnLike":{
            "aws:SourceARN": "arn:aws:s3:::amzn-s3-demo-source-bucket"
         }
       }
}
```

9. Pilih Simpan perubahan.

Untuk informasi selengkapnya tentang membuat CMK dan menggunakan kebijakan kunci, lihat tautan berikut di Panduan Developer AWS Key Management Service :

- Mengelola kunci
- Kebijakan utama di AWS KMS

> **Note**
>
> Pastikan tidak ada pernyataan Deny yang ditambahkan ke kebijakan bucket tujuan yang akan mencegah pengiriman laporan inventaris ke dalam bucket ini. Untuk informasi selengkapnya, lihat Mengapa saya tidak dapat membuat Laporan Inventaris Amazon S3? .

Mengonfigurasi inventaris dengan menggunakan konsol S3

Gunakan petunjuk ini untuk mengonfigurasi inventaris dengan menggunakan konsol S3.

> **Note**
>
> Amazon S3 akan membutuhkan waktu hingga 48 jam untuk mengirimkan laporan inventaris pertama.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Dalam daftar bucket, pilih nama bucket yang ingin Anda konfigurasi Inventaris Amazon S3.
4. Pilih tab Manajemen.
5. Di bawah Konfigurasi inventaris, pilih Buat konfigurasi inventaris.
6. Untuk Nama konfigurasi inventaris, masukkan nama.
7. Untuk Cakupan inventaris, lakukan hal berikut:
   - Masukkan prefiks opsional.
   - Pilih versi objek yang akan disertakan, baik Versi saat ini saja atau Sertakan semua versi.
8. Di bawah Detail laporan, pilih lokasi Akun AWS tempat Anda ingin menyimpan laporan: Akun ini atau Akun lain.
9. Di bawah Tujuan, pilih bucket tujuan tempat Anda ingin menyimpan laporan inventaris.

   Bucket tujuan harus Wilayah AWS sama dengan ember tempat Anda menyiapkan inventaris. Bucket tujuan dapat berada di Akun AWS yang berbeda. Saat menentukan bucket tujuan, Anda juga dapat menyertakan prefiks opsional untuk mengelompokkan laporan inventaris Anda.

   Di bawah bidang bucket Tujuan, terdapat pernyataan Izin bucket tujuan yang ditambahkan ke kebijakan bucket tujuan agar Amazon S3 dapat menempatkan data di bucket tersebut. Untuk informasi selengkapnya, lihat Membuat kebijakan bucket tujuan.
10. Di bawah Frekuensi, pilih seberapa sering laporan akan dibuat: Harian atau Mingguan.
11. Untuk Format output, pilih salah satu format berikut untuk laporan:
    - CSV–Jika Anda berencana untuk menggunakan laporan inventaris ini dengan Operasi Batch S3 atau jika Anda ingin menganalisis laporan ini di alat lain, seperti Microsoft Excel, pilih CSV.
    - Apache ORC
    - Apache Parquet
12. Di bawah Status, pilih Aktifkan atau Nonaktifkan.
13. Untuk mengonfigurasi enkripsi di sisi server, Enkripsi laporan inventaris, ikuti langkah-langkah berikut:
    a. Di bawah Enkripsi sisi server, pilih Jangan tentukan kunci enkripsi atau Tentukan kunci enkripsi untuk mengenkripsi data.

- Agar pengaturan bucket tetap menggunakan enkripsi di sisi server default untuk objek saat menyimpannya di Amazon S3, pilih Jangan tentukan kunci enkripsi. Selama tujuan bucket telah mengaktifkan Kunci Bucket S3, operasi penyalinan akan menerapkan Kunci Bucket S3 di bucket tujuan.

  > **Note**
  >
  > Jika kebijakan bucket untuk tujuan yang ditentukan mengharuskan objek dienkripsi sebelum menyimpannya di Amazon S3, Anda harus memilih Tentukan kunci enkripsi. Jika tidak, penyalinan objek ke tujuan akan gagal.

- Untuk mengenkripsi objek sebelum menyimpannya di Amazon S3, pilih Tentukan kunci enkripsi.

b. Jika Anda memilih Tentukan kunci enkripsi, di bawah Jenis enkripsi, Anda harus memilih kunci terkelola Amazon S3 (SSE-S3) atau AWS Key Management Service kunci (SSE-KMS).

   SSE-S3 menggunakan salah satu penyandian blok terkuat—Advanced Encryption Standard 256-bit (AES-256) untuk mengenkripsi setiap objek. SSE-KMS memberikan kontrol yang lebih besar terhadap kunci Anda. Untuk informasi selengkapnya tentang SSE-S3, lihat. Menggunakan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3) Untuk informasi lebih lanjut tentang SSE-KMS, lihat Menggunakan enkripsi sisi server dengan AWS KMS kunci (SSE-KMS).

   > **Note**
   >
   > Untuk mengenkripsi file daftar inventaris dengan SSE-KMS, Anda harus memberikan izin kepada Amazon S3 untuk menggunakan CMK. Untuk petunjuk, lihat Memberikan Izin kepada Amazon S3 untuk Melakukan Enkripsi Menggunakan Kunci KMS Anda.

c. Jika Anda memilih AWS Key Management Service kunci (SSE-KMS), di bawah AWS KMS key, Anda dapat menentukan AWS KMS kunci Anda melalui salah satu opsi berikut.

> **Note**
>
> Jika bucket tujuan yang menyimpan file daftar inventaris dimiliki oleh yang berbeda Akun AWS, pastikan Anda menggunakan ARN kunci KMS yang memenuhi syarat untuk menentukan kunci KMS Anda.

- Untuk memilih dari daftar kunci KMS yang tersedia, pilih Pilih dari AWS KMS kunci Anda, dan pilih kunci KMS enkripsi simetris dari daftar kunci yang tersedia. Pastikan kunci KMS berada di Region yang sama dengan bucket Anda.

  > **Note**
  >
  > Kunci Kunci yang dikelola AWS (`aws/s3`) dan kunci terkelola pelanggan Anda muncul dalam daftar. Namun, Kunci yang dikelola AWS (`aws/s3`) tidak didukung untuk enkripsi SSE-KMS dengan S3 Inventory.

- Untuk memasukkan ARN kunci KMS, pilih Enter AWS KMS key ARN, dan masukkan ARN kunci KMS Anda di bidang yang muncul.
- Untuk membuat kunci terkelola pelanggan baru di AWS KMS konsol, pilih Buat kunci KMS.

14. Untuk Bidang metadata tambahan, pilih satu atau beberapa hal berikut untuk ditambahkan ke laporan inventaris:

    - Ukuran–Ukuran objek dalam byte, tidak termasuk ukuran unggahan multibagian yang tidak lengkap, metadata objek, dan penanda hapus.
    - Tanggal terakhir diubah–Tanggal pembuatan objek atau tanggal modifikasi terakhir, mana pun yang terbaru.
    - Unggahan multibagian–Menentukan bahwa objek diunggah sebagai unggahan multibagian. Untuk informasi selengkapnya, lihat Mengunggah dan menyalin objek menggunakan unggahan multipart di Amazon S3.
    - Status replikasi–Status replikasi objek. Untuk informasi selengkapnya, lihat Mendapatkan informasi status replikasi.
    - Status enkripsi–Jenis enkripsi di sisi server yang digunakan untuk mengenkripsi objek. Untuk informasi selengkapnya, lihat Melindungi data dengan enkripsi di sisi klien.

- Status Kunci Bucket - Menunjukkan apakah kunci tingkat ember yang dihasilkan oleh AWS KMS berlaku untuk objek. Untuk informasi selengkapnya, lihat Mengurangi biaya SSE-KMS dengan Kunci Bucket Amazon S3.
- Daftar kontrol akses objek — Daftar kontrol akses (ACL) untuk setiap objek yang mendefinisikan Akun AWS atau kelompok mana yang diberikan akses ke objek ini dan jenis akses yang diberikan. Untuk informasi selengkapnya tentang bidang ini, lihat Menggunakan bidang ACL Objek . Untuk informasi lebih lanjut tentang ACLs, lihatGambaran umum daftar kontrol akses (ACL).
- Pemilik objek–Pemilik objek.
- Kelas penyimpanan–Kelas penyimpanan yang digunakan untuk menyimpan objek.
- Intelligent-Tiering: Tingkat Akses–Menunjukkan tingkat akses (sering atau jarang) dari objek jika disimpan di kelas penyimpanan Intelligent-Tiering S3. Untuk informasi selengkapnya, lihat Kelas penyimpanan untuk mengoptimalkan data secara otomatis dengan pola akses yang berubah atau tidak diketahui.
- ETagTag entitas (ETag) adalah hash dari objek. ETag Merefleksikan perubahan hanya pada isi objek, bukan metadatanya. ETag Mungkin atau mungkin bukan MD5 intisari data objek. Ketentuannya tergantung pada bagaimana objek dibuat dan bagaimana objek dienkripsi. Untuk informasi selengkapnya, lihat Object dalam Referensi API Amazon Simple Storage Service.
- Algoritme checksum–Menunjukkan algoritme yang digunakan untuk membuat checksum objek. Untuk informasi selengkapnya, lihat Menggunakan algoritma checksum yang didukung.
- Semua konfigurasi Kunci Objek–Status Kunci Objek, termasuk pengaturan berikut:
  - Kunci Objek: Mode retensi–Tingkat perlindungan yang diterapkan pada objek, baik Tata Kelola atau Kepatuhan.
  - Kunci Objek: Pertahankan hingga tanggal–Tanggal hingga objek yang terkunci tidak dapat dihapus.
  - Kunci Objek: Status penyimpanan yang sah–Status penyimpanan yang sah objek yang terkunci.

  Untuk informasi selngkapnya tentang Kunci Objek S3, lihat Cara kerja Kunci Objek S3.

Untuk informasi selengkapnya tentang konten laporan inventaris, lihat Daftar Inventaris Amazon S3.

Untuk informasi selengkapnya tentang membatasi akses ke bidang metadata opsional tertentu dalam konfigurasi inventaris, lihat. Kontrol pembuatan konfigurasi laporan Inventaris S3

15. Pilih Buat.

Penggunaan API REST untuk bekerja dengan Inventaris S3

Berikut ini adalah operasi REST yang dapat Anda gunakan untuk bekerja dengan Amazon S3 Inventory.

- DeleteBucketInventoryConfiguration
- GetBucketInventoryConfiguration
- ListBucketInventoryConfigurations
- PutBucketInventoryConfiguration

Menemukan daftar inventaris

Saat daftar inventaris diterbitkan, file manifes akan diterbitkan ke lokasi berikut di dalam bucket tujuan.

```
destination-prefix/amzn-s3-demo-source-bucket/config-ID/YYYY-MM-DDTHH-MMZ/manifest.json
 destination-prefix/amzn-s3-demo-source-bucket/config-ID/YYYY-MM-DDTHH-MMZ/
manifest.checksum
 destination-prefix/amzn-s3-demo-source-bucket/config-ID/hive/dt=YYYY-MM-DD-HH-MM/
symlink.txt
```

- *destination-prefix* adalah awalan nama kunci objek yang ditentukan secara opsional didalam konfigurasi inventaris. Gunakan awalan ini untuk mengelompokkan semua file daftar inventaris di lokasi yang sama di dalam bucket tujuan.
- *amzn-s3-demo-source-bucket* adalah bucket sumber yang digunakan untuk daftar inventaris. Nama bucket sumber ditambahkan untuk mencegah benturan saat beberapa laporan inventaris dari bucket sumber yang berbeda dikirimkan ke bucket tujuan yang sama.
- *config-ID* ditambahkan untuk mencegah benturan dengan beberapa laporan inventaris dari bucket sumber yang sama yang dikirim ke bucket tujuan yang sama. *config-ID* berasal dari konfigurasi laporan inventaris, dan merupakan nama untuk laporan yang ditentukan selama pengaturan.

- *YYYY-MM-DDTHH-MMZ* adalah stempel waktu yang berisi waktu mulai dan tanggal saat proses pembuatan laporan inventaris memulai pemindaian bucket; misalnya, `2016-11-06T21-32Z`.
- `manifest.json` adalah file manifes.
- `manifest.checksum`adalah MD5 hash dari isi `manifest.json` file.
- `symlink.txt` adalah file manifes yang kompatibel dengan Apache Hive.

Daftar inventaris diterbitkan setiap hari atau setiap minggu ke lokasi berikut di dalam bucket tujuan.

```
destination-prefix/amzn-s3-demo-source-bucket/config-ID/data/example-file-name.csv.gz
...
destination-prefix/amzn-s3-demo-source-bucket/config-ID/data/example-file-name-1.csv.gz
```

- *destination-prefix* adalah awalan nama kunci objek yang ditentukan secara opsional di dalam konfigurasi inventaris. Gunakan awalan ini untuk mengelompokkan semua file daftar inventaris di lokasi yang sama di dalam bucket tujuan.
- *amzn-s3-demo-source-bucket* adalah bucket sumber yang digunakan untuk daftar inventaris. Nama bucket sumber ditambahkan untuk mencegah benturan saat beberapa laporan inventaris dari bucket sumber yang berbeda dikirimkan ke bucket tujuan yang sama.
- *example-file-name*`.csv.gz` adalah salah satu file inventaris CSV. Nama inventaris ORC diakhiri dengan ekstensi nama file `.orc`, dan nama inventaris Parquet diakhiri dengan ekstensi nama file `.parquet`.

Manifes inventaris

File manifes `manifest.json` dan `symlink.txt` menjelaskan lokasi tempat file inventaris berada. Setiap kali ada daftar inventaris baru yang dikirimkan, daftar tersebut akan disertai dengan serangkaian file manifes baru. File-file ini mungkin akan saling menimpa. Di bucket dengan Penentuan Versi yang diaktifkan, Amazon S3 akan membuat versi baru dari file manifes tersebut.

Setiap manifes yang terdapat di dalam file `manifest.json` menyediakan metadata dan informasi dasar lainnya tentang inventaris. Informasi ini mencakup hal-hal berikut:

- Nama bucket sumber
- Nama bucket tujuan
- Versi inventaris

- Pembuatan stempel waktu dengan format tanggal epoch yang terdiri dari waktu mulai dan tanggal saat proses pembuatan laporan inventaris memulai pemindaian bucket
- Format dan skema file inventaris
- Daftar file inventaris yang berada di dalam bucket tujuan

Setiap kali `manifest.json` file ditulis, itu disertai dengan `manifest.checksum` file yang merupakan MD5 hash dari konten `manifest.json` file.

Example Manifes inventaris di dalam file **`manifest.json`**

Contoh berikut menunjukkan manifes inventaris di dalam file `manifest.json` untuk CSV, ORC, dan inventaris yang diformat Parquet.

CSV

Berikut ini adalah contoh manifes di dalam file `manifest.json` untuk inventaris yang diformat CSV.

```
{
    "sourceBucket": "amzn-s3-demo-source-bucket",
    "destinationBucket": "arn:aws:s3:::example-inventory-destination-bucket",
    "version": "2016-11-30",
    "creationTimestamp" : "1514944800000",
    "fileFormat": "CSV",
    "fileSchema": "Bucket, Key, VersionId, IsLatest, IsDeleteMarker,
 Size, LastModifiedDate, ETag, StorageClass, IsMultipartUploaded,
 ReplicationStatus, EncryptionStatus, ObjectLockRetainUntilDate, ObjectLockMode,
 ObjectLockLegalHoldStatus, IntelligentTieringAccessTier, BucketKeyStatus,
 ChecksumAlgorithm, ObjectAccessControlList, ObjectOwner",
    "files": [
        {
            "key": "Inventory/amzn-s3-demo-source-bucket/2016-11-06T21-32Z/
files/939c6d46-85a9-4ba8-87bd-9db705a579ce.csv.gz",
            "size": 2147483647,
            "MD5checksum": "f11166069f1990abeb9c97ace9cdfabc"
        }
    ]
}
```

ORC

Berikut ini adalah contoh manifes di dalam file `manifest.json` untuk inventaris yang diformat ORC.

```
{
    "sourceBucket": "amzn-s3-demo-source-bucket",
    "destinationBucket": "arn:aws:s3:::example-destination-bucket",
    "version": "2016-11-30",
    "creationTimestamp" : "1514944800000",
    "fileFormat": "ORC",
    "fileSchema":
 "struct<bucket:string,key:string,version_id:string,is_latest:boolean,is_delete_marker:boole
    "files": [
        {
            "key": "inventory/amzn-s3-demo-source-bucket/data/
d794c570-95bb-4271-9128-26023c8b4900.orc",
            "size": 56291,
            "MD5checksum": "5925f4e78e1695c2d020b9f6eexample"
        }
    ]
}
```

Parquet

Berikut ini adalah contoh manifes di dalam file `manifest.json` untuk inventaris yang diformat Parquet.

```
{
    "sourceBucket": "amzn-s3-demo-source-bucket",
    "destinationBucket": "arn:aws:s3:::example-destination-bucket",
    "version": "2016-11-30",
    "creationTimestamp" : "1514944800000",
    "fileFormat": "Parquet",
    "fileSchema": "message s3.inventory { required binary bucket (UTF8);
 required binary key (UTF8); optional binary version_id (UTF8); optional boolean
 is_latest; optional boolean is_delete_marker; optional int64 size; optional
 int64 last_modified_date (TIMESTAMP_MILLIS); optional binary e_tag (UTF8);
 optional binary storage_class (UTF8); optional boolean is_multipart_uploaded;
 optional binary replication_status (UTF8); optional binary encryption_status
 (UTF8); optional int64 object_lock_retain_until_date (TIMESTAMP_MILLIS); optional
 binary object_lock_mode (UTF8); optional binary object_lock_legal_hold_status
 (UTF8); optional binary intelligent_tiering_access_tier (UTF8); optional binary
```

```
 bucket_key_status (UTF8); optional binary checksum_algorithm (UTF8); optional
 binary object_access_control_list (UTF8); optional binary object_owner (UTF8);}",
    "files": [
        {
           "key": "inventory/amzn-s3-demo-source-bucket/data/
d754c470-85bb-4255-9218-47023c8b4910.parquet",
            "size": 56291,
            "MD5checksum": "5825f2e18e1695c2d030b9f6eexample"
        }
    ]
}
```

File `symlink.txt` adalah file manifes yang kompatibel dengan Apache Hive dan memungkinkan Hive untuk menemukan file inventaris dan file data terkait secara otomatis. Manifes yang kompatibel dengan Hive dapat berfungsi untuk layanan yang kompatibel dengan Hive Athena dan Amazon Redshift Spectrum. File ini juga berfungsi di aplikasi yang kompatibel dengan Hive, termasuk Presto, Apache Hive, Apache Spark, dan banyak lainnya.

> **Important**
>
> Saat ini, file manifes `symlink.txt` yang kompatibel dengan Apache Hive tidak berfungsi dengan AWS Glue.
>
> Pembacaan file `symlink.txt` dengan Apache Hive dan Apache Spark tidak didukung untuk ORC dan file inventaris yang diformat dengan Parquet.

Menyiapkan Notifikasi Peristiwa Amazon S3 untuk penyelesaian inventaris

Anda dapat mengatur notifikasi peristiwa Amazon S3 untuk menerima pemberitahuan saat file checksum manifes dibuat, yang menandakan bahwa daftar inventaris sudah ditambahkan ke bucket tujuan. Manifes adalah up-to-date daftar semua daftar inventaris di lokasi tujuan.

Amazon S3 dapat menerbitkan peristiwa ke topik Amazon Simple Notification Service (Amazon SNS), antrean Amazon Simple Queue Service (Amazon SQS), atau fungsi AWS Lambda . Untuk informasi selengkapnya, lihat Notifikasi Peristiwa Amazon S3.

Konfigurasi notifikasi berikut menerangkan bahwa semua file `manifest.checksum` yang baru ditambahkan ke bucket tujuan akan diproses oleh AWS Lambda `cloud-function-list-write`.

```
<NotificationConfiguration>
  <QueueConfiguration>
      <Id>1</Id>
      <Filter>
          <S3Key>
              <FilterRule>
                  <Name>prefix</Name>
                  <Value>destination-prefix/source-bucket</Value>
              </FilterRule>
              <FilterRule>
                  <Name>suffix</Name>
                  <Value>checksum</Value>
              </FilterRule>
          </S3Key>
      </Filter>
      <CloudFunction>arn:aws:lambda:us-west-2:222233334444:cloud-function-list-write</
CloudFunction>
      <Event>s3:ObjectCreated:*</Event>
  </QueueConfiguration>
  </NotificationConfiguration>
```

Untuk informasi selengkapnya, lihat Menggunakan AWS Lambda Amazon S3 di Panduan AWS Lambda Pengembang.

Membuat kueri inventaris Amazon S3 dengan Amazon Athena

Anda dapat membuat kueri file Inventaris Amazon S3 dengan kueri SQL standar menggunakan Amazon Athena di semua Wilayah yang menyediakan Athena. Untuk memeriksa Wilayah AWS ketersediaan, lihat Wilayah AWS Tabel.

Athena dapat membuat kueri file Inventaris Amazon S3 di format kolom baris yang dioptimalkan (ORC) Apache, Apache Parquet, atau nilai yang dipisahkan koma (CSV). Saat Anda menggunakan Athena untuk membuat kueri file inventaris, kami sarankan untuk menggunakan file inventaris yang diformat ORC atau berformat Parquet. Format ORC dan Parquet memberikan kinerja kueri yang lebih cepat dan biaya kueri yang lebih rendah. ORC dan Parquet merupakan format file berkolom yang dapat mendeskripsikan diri sendiri dan memiliki kesadaran jenis Apache Hadoop. Format berkolom memungkinkan pembaca untuk membaca, mendekompresi, dan hanya memproses kolom yang diperlukan untuk kueri saat ini. ORC dan Parquet format untuk Inventaris Amazon S3 tersedia di semua. Wilayah AWS

Menggunakan Athena untuk membuat kueri file Inventaris Amazon S3

1. Buat tabel Athena. Untuk informasi cara membuat tabel, lihat Membuat Tabel di Amazon Athena dalam Panduan Pengguna Amazon Athena.
2. Buat kueri Anda dengan menggunakan salah satu templat kueri contoh berikut ini, tergantung pada apakah Anda membuat kueri berformat ORC, berformat Parket, atau berformat CSV untuk laporan inventaris.

   - Saat Anda menggunakan Athena untuk membuat kueri laporan inventaris yang berformat ORC, gunakan kueri sampel berikut sebagai templat.

     Kueri sampel berikut mencakup seluruh kolom opsional dalam laporan inventaris yang berformat ORC.

     Untuk menggunakan kueri sampel ini, lakukan hal berikut:

     - Ganti *your_table_name* dengan nama tabel Athena yang Anda buat.
     - Hapus bidang opsional yang tidak Anda pilih untuk inventaris sehingga kueri sesuai dengan bidang yang dipilih untuk inventaris Anda.
     - Ganti nama bucket dan lokasi inventaris berikut (ID konfigurasi) yang sesuai untuk konfigurasi Anda.

       `s3://amzn-s3-demo-bucket/config-ID/hive/`

     - Ganti *2022-01-01-00-00* tanggal di bawah `projection.dt.range` dengan hari pertama rentang waktu di mana Anda mempartisi data di Athena. Untuk informasi selengkapnya, lihat Data partisi di Athena.

```
CREATE EXTERNAL TABLE your_table_name(
         bucket string,
         key string,
         version_id string,
         is_latest boolean,
         is_delete_marker boolean,
         size bigint,
         last_modified_date timestamp,
         e_tag string,
         storage_class string,
         is_multipart_uploaded boolean,
         replication_status string,
         encryption_status string,
```

```
         object_lock_retain_until_date bigint,
         object_lock_mode string,
         object_lock_legal_hold_status string,
         intelligent_tiering_access_tier string,
         bucket_key_status string,
         checksum_algorithm string,
         object_access_control_list string,
         object_owner string
) PARTITIONED BY (
        dt string
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.orc.OrcSerde'
  STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.SymlinkTextInputFormat'
  OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.IgnoreKeyTextOutputFormat'
  LOCATION 's3://source-bucket/config-ID/hive/'
  TBLPROPERTIES (
    "projection.enabled" = "true",
    "projection.dt.type" = "date",
    "projection.dt.format" = "yyyy-MM-dd-HH-mm",
    "projection.dt.range" = "2022-01-01-00-00,NOW",
    "projection.dt.interval" = "1",
    "projection.dt.interval.unit" = "HOURS"
  );
```

- Ketika Anda menggunakan Athena untuk membuat kueri laporan inventaris berformat Parquet, gunakan contoh kueri untuk laporan berformat ORC. Namun, gunakan yang berikut ini sebagai Parquet SerDe pengganti ORC SerDe dalam `ROW FORMAT SERDE` pernyataan.

  ```
  ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
  ```

- Saat Anda menggunakan Athena untuk membuat kueri laporan inventaris berformat CSV, gunakan kueri sampel berikut sebagai templat.

  Kueri sampel berikut mencakup seluruh kolom opsional dalam laporan inventaris berformat CSV.

  Untuk menggunakan kueri sampel ini, lakukan hal berikut:

  - Ganti *`your_table_name`* dengan nama tabel Athena yang Anda buat.
  - Hapus bidang opsional yang tidak Anda pilih untuk inventaris sehingga kueri sesuai dengan bidang yang dipilih untuk inventaris Anda.

- Ganti nama bucket dan lokasi inventaris berikut (ID konfigurasi) yang sesuai untuk konfigurasi Anda.

  `s3://amzn-s3-demo-bucket/config-ID/hive/`

- Ganti *2022-01-01-00-00* tanggal di bawah `projection.dt.range` dengan hari pertama rentang waktu di mana Anda mempartisi data di Athena. Untuk informasi selengkapnya, lihat, Data partisi di Athena.

```
CREATE EXTERNAL TABLE your_table_name(
         bucket string,
         key string,
         version_id string,
         is_latest boolean,
         is_delete_marker boolean,
         size string,
         last_modified_date string,
         e_tag string,
         storage_class string,
         is_multipart_uploaded boolean,
         replication_status string,
         encryption_status string,
         object_lock_retain_until_date string,
         object_lock_mode string,
         object_lock_legal_hold_status string,
         intelligent_tiering_access_tier string,
         bucket_key_status string,
         checksum_algorithm string,
         object_access_control_list string,
         object_owner string
) PARTITIONED BY (
        dt string
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
  STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.SymlinkTextInputFormat'
  OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.IgnoreKeyTextOutputFormat'
  LOCATION 's3://source-bucket/config-ID/hive/'
  TBLPROPERTIES (
    "projection.enabled" = "true",
    "projection.dt.type" = "date",
    "projection.dt.format" = "yyyy-MM-dd-HH-mm",
    "projection.dt.range" = "2022-01-01-00-00,NOW",
    "projection.dt.interval" = "1",
```

```
    "projection.dt.interval.unit" = "HOURS"
  );
```

3. Sekarang Anda dapat menjalankan berbagai kueri pada inventaris, seperti yang ditunjukkan dalam contoh-contoh berikut. Ganti masing-masing *user input placeholder* dengan informasi Anda sendiri.

```
# Get a list of the latest inventory report dates available.
SELECT DISTINCT dt FROM your_table_name ORDER BY 1 DESC limit 10;

# Get the encryption status for a provided report date.
SELECT encryption_status, count(*) FROM your_table_name WHERE dt = 'YYYY-MM-DD-HH-
MM' GROUP BY encryption_status;

# Get the encryption status for inventory report dates in the provided range.
SELECT dt, encryption_status, count(*) FROM your_table_name
WHERE dt > 'YYYY-MM-DD-HH-MM' AND dt < 'YYYY-MM-DD-HH-MM' GROUP BY dt,
 encryption_status;
```

Saat Anda mengonfigurasi Inventaris S3 untuk menambahkan bidang Daftar Kontrol Akses Objek (ACL Objek) ke laporan inventaris, laporan tersebut akan menampilkan nilai untuk bidang ACL Objek sebagai string yang dienkodekan base64. Untuk mendapatkan nilai yang didekodekan dalam JSON untuk bidang ACL Objek, Anda dapat melakukan kueri pada bidang ini dengan menggunakan Athena. Lihat contoh-contoh kueri berikut. Untuk informasi selengkapnya tentang bidang ACL Objek, lihatMenggunakan bidang ACL Objek .

```
# Get the S3 keys that have Object ACL grants with public access.
WITH grants AS (
    SELECT key,
        CAST(
            json_extract(from_utf8(from_base64(object_access_control_list)),
 '$.grants') AS ARRAY(MAP(VARCHAR, VARCHAR))
        ) AS grants_array
    FROM your_table_name
)
SELECT key,
       grants_array,
       grant
FROM grants, UNNEST(grants_array) AS t(grant)
WHERE element_at(grant, 'uri') = 'http://acs.amazonaws.com/groups/global/AllUsers'
```

```
# Get the S3 keys that have Object ACL grantees in addition to the object owner.
WITH grants AS
    (SELECT key,
    from_utf8(from_base64(object_access_control_list)) AS
 object_access_control_list,
         object_owner,
         CAST(json_extract(from_utf8(from_base64(object_access_control_list)),
         '$.grants') AS ARRAY(MAP(VARCHAR, VARCHAR))) AS grants_array
    FROM your_table_name)
SELECT key,
       grant,
       objectowner
FROM grants, UNNEST(grants_array) AS t(grant)
WHERE cardinality(grants_array) > 1 AND element_at(grant, 'canonicalId') !=
 object_owner;
```

```
# Get the S3 keys with READ permission that is granted in the Object ACL.
WITH grants AS (
    SELECT key,
        CAST(
            json_extract(from_utf8(from_base64(object_access_control_list)),
 '$.grants') AS ARRAY(MAP(VARCHAR, VARCHAR))
        ) AS grants_array
    FROM your_table_name
)
SELECT key,
       grants_array,
       grant
FROM grants, UNNEST(grants_array) AS t(grant)
WHERE element_at(grant, 'permission') = 'READ';
```

```
# Get the S3 keys that have Object ACL grants to a specific canonical user ID.
WITH grants AS (
    SELECT key,
        CAST(
            json_extract(from_utf8(from_base64(object_access_control_list)),
 '$.grants') AS ARRAY(MAP(VARCHAR, VARCHAR))
        ) AS grants_array
    FROM your_table_name
```

```
)
SELECT key,
       grants_array,
       grant
FROM grants, UNNEST(grants_array) AS t(grant)
WHERE element_at(grant, 'canonicalId') = 'user-canonical-id';
```

```
# Get the number of grantees on the Object ACL.
SELECT key,
       object_access_control_list,
       json_array_length(json_extract(object_access_control_list,'$.grants')) AS
 grants_count
FROM your_table_name;
```

Untuk informasi selengkapnya tentang cara menggunakan Athena, lihat Panduan Pengguna Amazon Athena.

Mengonversi string ID versi kosong di laporan Inventaris Amazon S3 menjadi string null

> **Note**
>
> Prosedur berikut hanya berlaku untuk laporan Inventaris Amazon S3 yang menyertakan semua versi, dan hanya jika laporan "semua versi" digunakan sebagai manifes untuk Operasi Batch S3 pada bucket yang mengaktifkan Versi S3. Anda tidak diharuskan mengonversi string untuk laporan Inventaris S3 yang hanya menentukan versi saat ini.

Anda dapat menggunakan laporan Inventaris S3 sebagai manifes untuk Operasi Batch S3. Namun, saat Penentuan Versi S3 diaktifkan di bucket, laporan Inventaris S3 yang menyertakan semua versi menandai objek berversi nol dengan string kosong di bidang ID versi. Ketika Laporan Inventaris menyertakan semua versi objek IDs, Operasi Batch mengenali `null` string sebagai versi IDs, tetapi bukan string kosong.

Saat tugas Operasi Batch S3 menggunakan laporan Inventaris S3 "semua versi" sebagai manifes, tugas tersebut menggagalkan semua tugas pada objek yang memiliki string kosong di bidang ID

versi. Untuk mengonversi string kosong di bidang ID versi laporan Inventaris S3 menjadi string `null` untuk Operasi Batch, gunakan prosedur berikut.

Memperbarui laporan Inventaris Amazon S3 untuk digunakan dengan Operasi Batch

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Buka laporan Inventaris S3. Laporan inventaris terletak di bucket tujuan yang Anda tentukan saat mengonfigurasi laporan inventaris. Untuk informasi selengkapnya tentang cara menemukan laporan inventaris, lihat Menemukan daftar inventaris.
   a. Pilih bucket tujuan.
   b. Pilih folder. Folder ini dinamai bucket sumber asli.
   c. Pilih folder yang dinamai konfigurasi inventaris.
   d. Centang kotak di samping folder bernama hive. Di bagian atas halaman, pilih Salin URI S3 guna menyalin URI S3 untuk folder tersebut.
3. Buka konsol Amazon Athena di. https://console.aws.amazon.com/athena/
4. Di editor kueri, pilih Pengaturan, lalu pilih Kelola. Pada halaman Kelola pengaturan, untuk Lokasi hasil kueri, pilih bucket S3 untuk menyimpan hasil kueri Anda.
5. Di editor kueri, buat tabel Athena untuk menyimpan data dalam laporan inventaris menggunakan perintah berikut. Ganti *`table_name`* dengan nama pilihan Anda, dan dalam klausa `LOCATION`, masukkan URI S3 yang Anda salin sebelumnya. Kemudian pilih Jalankan untuk menjalankan kueri.

```
CREATE EXTERNAL TABLE table_name(bucket string, key string,
 version_id string) PARTITIONED BY (dt string)ROW FORMAT SERDE
 'org.apache.hadoop.hive.serde2.OpenCSVSerde'STORED AS INPUTFORMAT
 'org.apache.hadoop.hive.ql.io.SymlinkTextInputFormat' OUTPUTFORMAT
 'org.apache.hadoop.hive.ql.io.IgnoreKeyTextOutputFormat' LOCATION 'Copied S3 URI';
```

6. Untuk menghapus editor kueri, pilih Hapus. Kemudian muat laporan inventaris ke dalam tabel menggunakan perintah berikut. Ganti *`table_name`* dengan yang Anda pilih pada langkah sebelumnya. Kemudian pilih Jalankan untuk menjalankan kueri.

```
MSCK REPAIR TABLE table_name;
```

7. Untuk menghapus editor kueri, pilih Hapus. Jalankan `SELECT` kueri berikut untuk mengambil semua entri dalam laporan inventaris asli dan mengganti versi kosong IDs dengan `null` string.

Ganti *table_name* dengan yang Anda pilih sebelumnya, dan ganti *YYYY-MM-DD-HH-MM* dalam klausa WHERE dengan tanggal laporan inventaris yang Anda ingin alat ini dijalankan. Kemudian pilih Jalankan untuk menjalankan kueri.

```
SELECT bucket as Bucket, key as Key, CASE WHEN version_id = '' THEN 'null' ELSE
 version_id END as VersionId FROM table_name WHERE dt = 'YYYY-MM-DD-HH-MM';
```

8. Kembali ke konsol Amazon S3 (https://console.aws.amazon.com/s3/), dan navigasikan ke bucket S3 yang Anda pilih untuk Lokasi hasil kueri sebelumnya. Di dalam, akan ada serangkaian folder yang diakhiri dengan tanggal.

   Misalnya, Anda akan melihat sesuatu seperti s3:///***amzn-s3-demo-bucket***/Unsaved/2021/10/07/ ***query-result-location***. Anda akan melihat file `.csv` yang berisi hasil kueri SELECT yang Anda jalankan.

   Pilih file CSV dengan tanggal modifikasi terbaru. Unduh file ini ke mesin lokal Anda untuk langkah selanjutnya.

9. File CSV yang dihasilkan berisi baris header. Untuk menggunakan file CSV ini sebagai input untuk tugas Operasi Batch S3, Anda harus menghapus baris header, karena Operasi Batch tidak mendukung baris header pada manifes CSV.

   Untuk menghapus baris header, Anda dapat menjalankan salah satu perintah berikut pada file. Ganti *file.csv* dengan nama file CSV Anda.

   Untuk mesin macOS dan Linux, jalankan perintah `tail` di jendela Terminal.

   ```
   tail -n +2 file.csv > tmp.csv && mv tmp.csv file.csv
   ```

   Untuk mesin Windows, jalankan skrip berikut di PowerShell jendela Windows. Ganti *File-location* dengan jalur ke file Anda, dan *file.csv* dengan nama file.

   ```
   $ins = New-Object System.IO.StreamReader File-location\file.csv
   $outs = New-Object System.IO.StreamWriter File-location\temp.csv
   try {
       $skip = 0
       while ( !$ins.EndOfStream ) {
           $line = $ins.ReadLine();
           if ( $skip -ne 0 ) {
               $outs.WriteLine($line);
           } else {
   ```

```
            $skip = 1
        }
    }
} finally {
    $outs.Close();
    $ins.Close();
}
Move-Item File-location\temp.csv File-location\file.csv -Force
```

10. Setelah menghapus baris header dari file CSV, Anda siap menggunakannya sebagai manifes dalam pekerjaan Operasi Batch S3. Unggah file CSV ke bucket S3 atau lokasi yang Anda pilih, lalu buat pekerjaan Operasi Batch menggunakan file CSV sebagai manifes.

    Untuk informasi selengkapnya tentang cara membuat tugas Operasi Batch, lihat Membuat pekerjaan Operasi Batch S3.

Menggunakan bidang ACL Objek

Laporan Inventaris Amazon S3 berisi daftar objek dalam bucket sumber S3 dan metadata untuk setiap objek. Bidang daftar kontrol akses (ACL) Objek adalah bidang metadata yang tersedia di Inventaris Amazon S3. Secara khusus, ACL Objek berisi daftar kontrol akses (ACL) untuk setiap objek. ACL untuk objek mendefinisikan Akun AWS atau kelompok mana yang diberikan akses ke objek ini dan jenis akses yang diberikan. Untuk informasi selengkapnya, lihat Gambaran umum daftar kontrol akses (ACL) dan Daftar Inventaris Amazon S3.

Bidang ACL Objek dalam laporan Inventaris Amazon S3 ditentukan dalam format JSON. Data JSON terdiri dari bidang berikut:

- `version`–Versi format bidang ACL Objek dalam laporan inventaris. Ini menggunakan format tanggal `yyyy-mm-dd`.
- `status`–Nilai yang memungkinkan adalah `AVAILABLE` atau `UNAVAILABLE` untuk menunjukkan apakah ACL Objek tersedia untuk suatu objek. Ketika status untuk ACL Objek adalah `UNAVAILABLE`, nilai bidang Pemilik Objek dalam laporan inventaris juga `UNAVAILABLE`.
- `grants`–Pasangan izin-penerima yang mencantumkan status izin dari setiap penerima izin yang diberikan oleh ACL Objek. Nilai yang tersedia untuk penerima izin adalah `CanonicalUser` dan `Group`. Untuk informasi selengkapnya tentang penerima izin, lihat Penerima izin dalam daftar kontrol akses.

  Untuk penerima izin dengan tipe `Group`, pasangan izin-penerima mencakup atribut berikut:

- `uri`–Grup Amazon S3 yang telah ditentukan sebelumnya.
- `permission`–Izin ACL yang diberikan pada objek. Untuk informasi selengkapnya, lihat izin ACL pada objek.
- `type`–Jenis `Group`, yang menunjukkan bahwa penerima izin adalah kelompok.

Untuk penerima izin dengan jenis `CanonicalUser`, pasangan izin-penerima mencakup atribut berikut:

- `canonicalId`–Bentuk ID Akun AWS yang disamarkan. ID pengguna kanonik untuk akun khusus untuk akun Akun AWS itu. Anda dapat mengambil ID pengguna kanonik. Untuk informasi selengkapnya, lihat Menemukan ID pengguna kanonik untuk Anda Akun AWS di Panduan Referensi Manajemen AWS Akun.

  > **Note**
  >
  > Jika penerima hibah dalam ACL adalah alamat email dari Akun AWS, S3 Inventory menggunakan dari itu Akun AWS dan `CanonicalUser` jenisnya untuk `canonicalId` menentukan penerima hibah ini. Untuk informasi selengkapnya, lihat Penerima izin dalam daftar kontrol akses.

- `permission`–Izin ACL yang diberikan pada objek. Untuk informasi selengkapnya, lihat izin ACL pada objek.
- `type`— Jenis`CanonicalUser`, yang menunjukkan bahwa penerima hibah adalah. Akun AWS

Contoh berikut menunjukkan nilai yang memungkinkan untuk bidang ACL Objek dalam format JSON:

```
{
    "version": "2022-11-10",
    "status": "AVAILABLE",
    "grants": [{
        "uri": "http://acs.amazonaws.com/groups/global/AllUsers",
        "permission": "READ",
        "type": "Group"
    }, {
        "canonicalId": "example-canonical-id",
        "permission": "FULL_CONTROL",
        "type": "CanonicalUser"
    }]
}
```

Note

Bidang ACL Objek ditentukan dalam format JSON. Laporan inventaris akan menampilkan nilai untuk bidang Object ACL sebagai string yang dikodekan base64.
Misalnya, katakanlah Anda memiliki ACL Objek berikut dalam format JSON:

```
{
        "version": "2022-11-10",
        "status": "AVAILABLE",
        "grants": [{
            "canonicalId": "example-canonical-user-ID",
            "type": "CanonicalUser",
            "permission": "READ"
        }]
}
```

Bidang ACL Objek dikodekan dan ditampilkan sebagai string yang dikodekan dengan base64 berikut ini:

```
eyJ2ZXJzaW9uIjoiMjAyMi0xMS0xMCIsInN0YXR1cyI6IkFWQUlMQUJMRSIsImdyYW50cyI6W3siY2Fub25pY2FsSW
```

Untuk mendapatkan nilai yang terdekripsi dalam JSON untuk bidang ACL Objek, Anda dapat melakukan kueri pada bidang ini di Amazon Athena. Untuk contoh kueri, lihat Membuat kueri inventaris Amazon S3 dengan Amazon Athena.

Pola desain praktik terbaik: mengoptimalkan performa Amazon S3

Aplikasi Anda dapat dengan mudah mencapai ribuan transaksi per detik dalam performa permintaan saat mengunggah dan mengambil penyimpanan dari Amazon S3. Amazon S3 secara otomatis menyesuaikan ke tingkat permintaan yang tinggi. Misalnya, aplikasi Anda dapat mencapai setidaknya 3.500 PUT/COPY/POST/DELETE or 5,500 GET/HEAD permintaan per detik per awalan Amazon S3 yang dipartisi. Tidak ada batas jumlah prefiks dalam bucket. Anda dapat meningkatkan performa membaca atau menulis dengan menggunakan paralelisasi. Misalnya, jika Anda membuat 10 prefiks dalam bucket Amazon S3 untuk paralelisasi bacaan, Anda dapat menskalakan performa baca Anda menjadi 55.000 permintaan baca per detik. Demikian pula, Anda dapat menskalakan operasi tulis dengan menulis ke beberapa prefiks. Penskalaan, dalam kasus operasi baca dan tulis, terjadi secara bertahap dan tidak instan, dan kinerja aktual akan bervariasi berdasarkan karakteristik beban kerja spesifik Anda, pola penggunaan, dan konfigurasi sistem. Sementara Amazon S3 menskalakan ke tingkat permintaan baru Anda yang lebih tinggi, Anda mungkin melihat beberapa kesalahan 503 (Perlambat). Kesalahan ini akan hilang saat penskalaan selesai. Untuk informasi selengkapnya tentang cara membuat dan menggunakan prefiks, lihat Organisasi objek menggunakan prefiks.

Beberapa aplikasi danau data di Amazon S3 memindai jutaan atau miliar objek untuk kueri yang berjalan di atas petabyte data. Aplikasi data lake ini mencapai kecepatan transfer instans tunggal yang memaksimalkan penggunaan antarmuka jaringan untuk EC2 instans Amazon mereka, yang dapat mencapai hingga 100 Gb/s pada satu instance. Aplikasi ini kemudian menggabungkan throughput di beberapa instans untuk mendapatkan beberapa terabit per detik.

Aplikasi lain sensitif terhadap latensi, seperti aplikasi perpesanan media sosial. Aplikasi ini dapat mencapai latensi objek kecil yang konsisten (dan first-byte-out latensi untuk objek yang lebih besar) sekitar 100-200 milidetik.

AWS Layanan lain juga dapat membantu mempercepat kinerja untuk arsitektur aplikasi yang berbeda. Misalnya, jika Anda menginginkan laju transfer yang lebih tinggi pada koneksi HTTP tunggal atau latensi milidetik digit, gunakan Amazon CloudFront atau Amazon ElastiCache untuk menyimpan cache dengan Amazon S3.

Selain itu, jika Anda ingin transportasi data cepat melalui jarak jauh antara klien dan bucket S3, gunakan Mengonfigurasi transfer file yang cepat dan aman menggunakan Amazon S3 Transfer Acceleration. Transfer Acceleration menggunakan lokasi tepi yang didistribusikan secara global

CloudFront untuk mempercepat transportasi data melalui jarak geografis. Jika beban kerja Amazon S3 Anda menggunakan enkripsi sisi server AWS KMS, lihat AWS KMS Batas di Panduan AWS Key Management Service Pengembang untuk informasi tentang tarif permintaan yang didukung untuk kasus penggunaan Anda.

Topik berikut menjelaskan panduan praktik terbaik dan pola desain untuk mengoptimalkan performa aplikasi yang menggunakan Amazon S3. Lihat Pedoman kinerja untuk Amazon S3 dan Pola desain kinerja untuk Amazon S3 untuk informasi terkini tentang optimalisasi performa untuk Amazon S3.

> **Note**
>
> Untuk informasi selengkapnya tentang menggunakan kelas penyimpanan Amazon S3 Express One Zone dengan bucket direktori, lihat S3 Express One Zone dan Bekerja dengan bucket direktori.

Topik

- Pedoman kinerja untuk Amazon S3
- Pola desain kinerja untuk Amazon S3

Pedoman kinerja untuk Amazon S3

Ketika membangun aplikasi yang mengunggah dan mengambil objek dari Amazon S3, ikuti panduan praktik terbaik kami untuk mengoptimalkan performa. Kami juga menawarkan Pola desain kinerja untuk Amazon S3 yang lebih terperinci.

Untuk mendapatkan performa terbaik untuk aplikasi Anda di Amazon S3, kami merekomendasikan panduan berikut ini.

Topik

- Ukur performa
- Skalakan koneksi penyimpanan secara horizontal
- Gunakan pengambilan rentang byte
- Permintaan coba lagi untuk aplikasi yang sensitif latensi

- Gabungkan Amazon S3 (Penyimpanan) dan Amazon EC2 (komputasi) dalam hal yang sama Wilayah AWS
- Gunakan Amazon S3 Transfer Acceleration untuk meminimalkan latensi yang disebabkan oleh jarak
- Gunakan versi terbaru dari AWS SDKs

Ukur performa

Ketika mengoptimalkan performa, lihat kebutuhan throughput jaringan, CPU, dan DRAM. Bergantung pada campuran permintaan untuk sumber daya yang berbeda ini, mungkin ada baiknya mengevaluasi jenis EC2 instans Amazon yang berbeda. Untuk informasi selengkapnya tentang jenis instans, lihat Jenis Instance di Panduan EC2 Pengguna Amazon.

Ini juga membantu untuk melihat waktu pencarian DNS, latensi, dan kecepatan transfer data menggunakan alat analisis HTTP saat mengukur performa.

Untuk memahami persyaratan performa dan mengoptimalkan performa aplikasi Anda, Anda juga dapat memantau respons kesalahan 503 yang Anda terima. Memantau metrik performa tertentu dapat menimbulkan biaya tambahan. Untuk informasi selengkapnya, lihat Harga Amazon S3.

Pantau jumlah respons kesalahan status 503 (Perlambat)

Untuk memantau jumlah respons kesalahan status 503 yang Anda dapatkan, Anda dapat menggunakan salah satu opsi berikut:

- Gunakan metrik CloudWatch permintaan Amazon untuk Amazon S3. Metrik CloudWatch permintaan menyertakan metrik untuk respons status 5xx. Untuk informasi selengkapnya tentang metrik CloudWatch permintaan, lihatMemantau metrik dengan Amazon CloudWatch.
- Gunakan jumlah kesalahan 503 (Layanan Tidak Tersedia) yang tersedia di bagian metrik lanjutan Lensa Penyimpanan Amazon S3. Untuk informasi selengkapnya, lihat Menggunakan metrik Lensa Penyimpanan S3 untuk meningkatkan kinerja.
- Gunakan pencatatan log akses server Amazon S3. Dengan pencatatan akses server, Anda dapat memfilter dan meninjau semua permintaan yang menerima respons 503 (Kesalahan Internal). Anda juga dapat menggunakan Amazon Athena untuk mengurai log. Untuk informasi selengkapnya tentang pencatatan log akses server, lihat Pencatatan permintaan dengan pencatatan akses server.

Dengan memantau jumlah kode kesalahan status HTTP 503, Anda sering kali dapat memperoleh wawasan berharga tentang prefiks, kunci, atau bucket mana yang mendapatkan permintaan throttling paling banyak.

Skalakan koneksi penyimpanan secara horizontal

Menyebarkan permintaan di berbagai koneksi adalah pola desain umum untuk menskalakan performa secara horizontal. Ketika Anda membangun aplikasi berperforma tinggi, bayangkan Amazon S3 sebagai sistem terdistribusi yang sangat besar, bukan sebagai titik akhir jaringan tunggal seperti server penyimpanan tradisional. Anda dapat mencapai performa terbaik dengan mengeluarkan beberapa permintaan sekaligus ke Amazon S3. Sebarkan permintaan ini melalui koneksi terpisah untuk memaksimalkan bandwidth yang dapat diakses dari Amazon S3. Amazon S3 tidak memiliki batas untuk jumlah koneksi yang dilakukan ke bucket Anda.

Gunakan pengambilan rentang byte

Menggunakan Header HTTP `Range` di permintaan GET Object, Anda dapat mengambil rentang byte dari objek, dengan hanya mentransfer bagian tertentu. Anda dapat menggunakan koneksi bersamaan ke Amazon S3 untuk mengambil byte yang berbeda dari dalam objek yang sama. Ini membantu Anda mencapai throughput agregat yang lebih tinggi dibandingkan permintaan objek utuh tunggal. Pengambilan rentang objek besar yang lebih kecil juga memungkinkan aplikasi Anda untuk meningkatkan waktu coba kembali ketika permintaan terganggu. Untuk informasi selengkapnya, lihat Mengunggah objek.

Ukuran umum untuk permintaan rentang byte adalah 8 MB atau 16 MB. Jika objek adalah PUT yang menggunakan pengunggahan multibagian, praktik yang baik adalah GET dalam ukuran bagian yang sama (atau setidaknya sejajar dengan batas bagian) untuk performa terbaik. Permintaan GET dapat secara langsung menangani bagian individual; misalnya, `GET ?partNumber=N.`

Permintaan coba lagi untuk aplikasi yang sensitif latensi

Batas waktu dan percobaan ulang yang agresif membantu mendorong latensi yang konsisten. Mengingat skala besar Amazon S3, jika permintaan pertama berjalan lambat, permintaan yang dicoba kembali kemungkinan besar mengambil jejak yang berbeda dan dengan cepat berhasil. AWS SDKs Memiliki batas waktu yang dapat dikonfigurasi dan nilai coba lagi yang dapat Anda sesuaikan dengan toleransi aplikasi spesifik Anda.

Gabungkan Amazon S3 (Penyimpanan) dan Amazon EC2 (komputasi) dalam hal yang sama Wilayah AWS

Meskipun nama bucket S3 bersifat unik secara global, setiap bucket disimpan di Wilayah yang Anda pilih saat membuat bucket. Untuk mempelajari lebih lanjut tentang pedoman penamaan bucket, lihat Ikhtisar bucket dan aturan penamaan Bucket. Untuk mengoptimalkan kinerja, kami menyarankan Anda mengakses bucket dari EC2 instans Amazon dalam hal yang sama Wilayah AWS jika memungkinkan. Ini membantu mengurangi latensi jaringan dan biaya transfer data.

Untuk informasi lebih lanjut tentang biaya transfer data, lihat Harga Amazon S3.

Gunakan Amazon S3 Transfer Acceleration untuk meminimalkan latensi yang disebabkan oleh jarak

Mengonfigurasi transfer file yang cepat dan aman menggunakan Amazon S3 Transfer Acceleration mengelola transfer file yang cepat, mudah, dan aman dalam jarak geografis yang panjang antara klien dan bucket S3. Transfer Acceleration memanfaatkan lokasi edge yang didistribusikan secara global di Amazon CloudFront. Saat data tiba di lokasi edge, data diarahkan ke Amazon S3 melalui jalur jaringan yang dioptimalkan. Transfer Acceleration ideal untuk mentransfer gigabyte ke terabyte data secara teratur melintasi benua. Ini juga berguna bagi klien yang mengunggah ke bucket terpusat dari seluruh dunia.

Anda dapat menggunakan alat perbandingan Amazon S3 Transfer Acceleration Speed untuk membandingkan kecepatan unggah yang dipercepat dan tidak dipercepat di seluruh Wilayah Amazon S3. Alat Speed Comparison menggunakan unggahan multibagian untuk mentransfer file dari browser Anda ke berbagai Wilayah Amazon S3 dengan dan tanpa menggunakan Amazon S3 Transfer Acceleration.

Gunakan versi terbaru dari AWS SDKs

Ini AWS SDKs menyediakan dukungan bawaan untuk banyak pedoman yang direkomendasikan untuk mengoptimalkan kinerja Amazon S3. SDKs Menyediakan API yang lebih sederhana untuk memanfaatkan Amazon S3 dari dalam aplikasi dan diperbarui secara berkala untuk mengikuti praktik terbaik terbaru. Misalnya, logika SDKs sertakan untuk mencoba ulang permintaan secara otomatis pada kesalahan HTTP 503 dan berinvestasi dalam kode untuk merespons dan beradaptasi dengan koneksi yang lambat.

Ini SDKs juga menyediakan Manajer Transfer, yang mengotomatiskan koneksi penskalaan horizontal untuk mencapai ribuan permintaan per detik, menggunakan permintaan rentang byte jika sesuai.

Sangat penting untuk menggunakan versi terbaru dari AWS SDKs untuk mendapatkan fitur optimasi kinerja terbaru.

Anda juga dapat mengoptimalkan performa saat Anda menggunakan permintaan API REST HTTP. Saat menggunakan REST API, Anda harus mengikuti praktik terbaik yang sama yang merupakan bagian dari SDKs. Izinkan batas waktu dan percobaan ulang pada permintaan yang lambat, dan beberapa koneksi untuk memungkinkan pengambilan data objek secara paralel. Untuk informasi selengkapnya tentang API REST, lihat Referensi API Amazon Simple Storage Service.

Pola desain kinerja untuk Amazon S3

Saat merancang aplikasi untuk mengunggah dan mengambil objek dari Amazon S3, gunakan pola desain praktik terbaik kami untuk mencapai performa terbaik untuk aplikasi Anda. Kami juga menawarkan Pedoman kinerja untuk Amazon S3 untuk Anda pertimbangkan saat sedang merencanakan arsitektur aplikasi Anda.

Untuk mengoptimalkan performa, Anda dapat menggunakan pola desain berikut.

Topik

- Menggunakan caching untuk konten yang sering diakses
- Batas waktu dan percobaan ulang untuk aplikasi yang sensitif terhadap latensi
- Penskalaan horizontal dan paralelisasi permintaan untuk throughput tinggi
- Menggunakan Amazon S3 Transfer Acceleration untuk mempercepat transfer data yang berbeda secara geografis

Menggunakan caching untuk konten yang sering diakses

Banyak aplikasi yang menyimpan data di Amazon S3 melayani “set kerja” data yang berulang kali diminta oleh pengguna. Jika beban kerja mengirimkan permintaan GET berulang untuk sekumpulan objek umum, Anda dapat menggunakan cache seperti Amazon CloudFront, Amazon ElastiCache, atau AWS Elemental MediaStoreuntuk mengoptimalkan kinerja. Penerapan cache yang berhasil dapat menghasilkan latensi rendah dan laju transfer data tinggi. Aplikasi yang menggunakan caching juga mengirimkan lebih sedikit permintaan langsung ke Amazon S3, yang dapat membantu mengurangi biaya permintaan.

Amazon CloudFront adalah jaringan pengiriman konten cepat (CDN) yang secara transparan menyimpan data dari Amazon S3 dalam satu set besar titik kehadiran yang didistribusikan secara

geografis (). PoPs Ketika objek dapat diakses dari beberapa Wilayah, atau melalui internet, CloudFront memungkinkan data untuk di-cache dekat dengan pengguna yang mengakses objek. Ini dapat menghasilkan pengiriman konten Amazon S3 populer dengan performa tinggi. Untuk selengkapnya CloudFront, lihat Panduan CloudFront Pengembang Amazon.

Amazon ElastiCache adalah cache dalam memori yang dikelola. Dengan ElastiCache, Anda dapat menyediakan EC2 instans Amazon yang menyimpan objek cache dalam memori. Hasil caching ini menghasilkan urutan pengurangan besar-besaran dalam latensi GET dan peningkatan substansial dalam throughput unduhan. Untuk menggunakannya ElastiCache, Anda memodifikasi logika aplikasi untuk mengisi cache dengan objek panas dan memeriksa cache untuk objek panas sebelum memintanya dari Amazon S3. Untuk contoh penggunaan ElastiCache untuk meningkatkan kinerja Amazon S3 GET, lihat posting blog Turbocharge Amazon S3 dengan Amazon untuk Redis. ElastiCache

AWS Elemental MediaStore adalah sistem caching dan distribusi konten yang khusus dibuat untuk alur kerja video dan pengiriman media dari Amazon S3. MediaStore menyediakan end-to-end penyimpanan APIs khusus untuk video, dan direkomendasikan untuk beban kerja video yang sensitif terhadap kinerja. Untuk selengkapnya MediaStore, lihat Panduan AWS Elemental MediaStore Pengguna.

Batas waktu dan percobaan ulang untuk aplikasi yang sensitif terhadap latensi

Ada situasi tertentu ketika aplikasi menerima respons dari Amazon S3 yang menunjukkan bahwa percobaan ulang diperlukan. Bucket peta Amazon S3 dan nama objek ke data objek yang terkait dengannya. Jika aplikasi menghasilkan tingkat permintaan yang tinggi (biasanya tingkat permintaan lebih dari 5.000 permintaan per detik ke sejumlah kecil objek), aplikasi mungkin akan menerima respons perlambatan HTTP 503. Jika kesalahan ini terjadi, setiap SDK AWS menerapkan logika percobaan ulang otomatis menggunakan mundur eksponensial. Jika Anda tidak menggunakan SDK AWS , Anda harus menerapkan logika percobaan ulang saat menerima kesalahan HTTP 503. Untuk informasi tentang teknik back-off, lihat Coba lagi perilaku di Panduan Referensi Alat AWS SDKs dan Alat.

Amazon S3 secara otomatis menskalakan dalam menanggapi tingkat permintaan baru yang berkelanjutan, secara dinamis mengoptimalkan performa. Meskipun Amazon S3 secara internal mengoptimalkan untuk tingkat permintaan baru, Anda akan menerima respons permintaan HTTP 503 secara sementara hingga optimalisasi selesai. Setelah Amazon S3 secara internal mengoptimalkan

performa untuk tingkat permintaan baru, semua permintaan umumnya dilayani tanpa percobaan ulang.

Untuk aplikasi sensitif latensi, Amazon S3 menyarankan pelacakan dan mencoba kembali operasi yang lebih lambat secara agresif. Saat Anda mencoba lagi permintaan, kami menyarankan menggunakan koneksi baru ke Amazon S3 dan melakukan pencarian DNS baru.

Saat Anda membuat permintaan berukuran besar (misalnya, lebih dari 128 MB), kami menyarankan untuk melacak throughput yang dicapai, dan mencoba kembali 5 persen paling lambat dari permintaan tersebut. Saat Anda membuat permintaan yang lebih kecil (misalnya, kurang dari 512 KB), yaitu latensi median sering kali berada dalam rentang milidetik, panduan yang baik adalah mencoba kembali operasi GET atau PUT setelah 2 detik. Jika percobaan ulang tambahan diperlukan, praktik terbaik adalah back off. Misalnya, kami sarankan untuk mengeluarkan satu percobaan ulang setelah 2 detik dan percobaan ulang kedua setelah 4 detik tambahan.

Jika aplikasi Anda membuat permintaan ukuran tetap ke Amazon S3, Anda akan mengharapkan waktu respons yang lebih konsisten untuk setiap permintaan ini. Dalam kasus ini, strategi sederhana adalah mengidentifikasi 1 persen permintaan paling lambat dan mencobanya lagi. Bahkan, satu kali coba ulang sering kali efektif dalam mengurangi latensi.

Jika Anda menggunakan AWS Key Management Service (AWS KMS) untuk enkripsi sisi server, lihat Kuota di Panduan AWS Key Management Service Pengembang untuk informasi tentang tarif permintaan yang didukung untuk kasus penggunaan Anda.

Penskalaan horizontal dan paralelisasi permintaan untuk throughput tinggi

Amazon S3 adalah sistem terdistribusi yang sangat besar. Untuk membantu Anda memanfaatkan skalanya, kami mendorong Anda untuk menskalakan permintaan paralel secara horizontal ke titik akhir layanan Amazon S3. Selain mendistribusikan permintaan di dalam Amazon S3, jenis pendekatan penskalaan ini membantu mendistribusikan muatan melalui beberapa jejak melalui jaringan.

Untuk transfer dengan throughput tinggi, Amazon S3 menyarankan penggunaan aplikasi yang menggunakan beberapa koneksi untuk GET atau PUT data secara paralel. Misalnya, ini didukung oleh Amazon S3 Transfer Manager di AWS Java SDK, dan sebagian besar lainnya AWS SDKs menyediakan konstruksi serupa. Untuk beberapa aplikasi, Anda dapat mencapai koneksi paralel dengan meluncurkan beberapa permintaan secara bersamaan dalam utas aplikasi yang berbeda, atau dalam instans aplikasi yang berbeda. Pendekatan terbaik untuk diambil tergantung pada aplikasi Anda dan struktur objek yang Anda akses.

Anda dapat menggunakan AWS SDKs untuk mengeluarkan permintaan GET dan PUT secara langsung daripada menggunakan manajemen transfer di AWS SDK. Dengan pendekatan ini, Anda dapat mengatur beban kerja secara lebih langsung, sembari tetap memanfaatkan dukungan SDK untuk percobaan ulang dan penanganannya terhadap setiap respons HTTP 503 yang mungkin terjadi. Sebagai aturan umum, saat Anda mengunduh objek besar dalam Wilayah dari Amazon S3 ke Amazon EC2, kami sarankan untuk membuat permintaan bersamaan untuk rentang byte objek dengan perincian 8-16 MB. Buat satu permintaan bersamaan untuk setiap 85—90 throughput jaringan MB/s yang diinginkan. Untuk menjenuhkan 10 kartu antarmuka Gb/s jaringan (NIC), Anda dapat menggunakan sekitar 15 permintaan bersamaan melalui koneksi terpisah. Anda dapat meningkatkan permintaan bersamaan melalui lebih banyak koneksi untuk jenuh lebih cepat NICs, seperti 25 Gb/s atau 100. Gb/s NICs

Mengukur performa sangat penting saat Anda menyesuaikan jumlah permintaan untuk masalah secara bersamaan. Kami menyarankan untuk memulai dengan permintaan tunggal pada satu waktu. Mengukur bandwith jaringan yang dicapai dan penggunaan sumber daya lain yang digunakan aplikasi Anda dalam memproses data. Anda kemudian dapat mengidentifikasi sumber daya bottleneck (yaitu, sumber daya dengan penggunaan tertinggi), dan karenanya jumlah permintaan yang mungkin berguna. Misalnya, jika memproses satu permintaan pada satu waktu menghasilkan penggunaan CPU sebesar 25 persen, hal ini menunjukkan bahwa hingga empat permintaan bersamaan dapat diakomodasi. Pengukuran sangat penting, dan perlu mengonfirmasi penggunaan sumber daya karena tingkat permintaan meningkat.

Jika aplikasi Anda menerbitkan permintaan secara langsung ke Amazon S3 menggunakan API REST, kami menyarankan menggunakan kumpulan koneksi HTTP dan menggunakan kembali setiap koneksi untuk serangkaian permintaan. Menghindari penyiapan koneksi per permintaan menghilangkan kebutuhan untuk menjalankan TCP slow-start dan handshake Secure Sockets Layer (SSL) pada setiap permintaan. Untuk informasi selengkapnya tentang API REST, lihat Referensi API Amazon Simple Storage Service.

Terakhir, penting untuk memperhatikan DNS dan pemeriksaan ulang bahwa permintaan tersebar di sejumlah besar alamat IP Amazon S3. Kueri DNS untuk siklus Amazon S3 melalui daftar besar titik akhir IP. Namun pemecah cache atau kode aplikasi yang menggunakan kembali satu alamat IP tidak mendapatkan keuntungan dari keragaman alamat dan load balancing yang mengikutinya. Alat utilitas jaringan seperti alat baris perintah `netstat` dapat menunjukkan alamat IP yang digunakan untuk komunikasi dengan Amazon S3, dan kami memberikan panduan untuk konfigurasi DNS untuk digunakan. Untuk informasi selengkapnya tentang pedoman ini, lihat Membuat permintaan di Referensi API Amazon S3.

Menggunakan Amazon S3 Transfer Acceleration untuk mempercepat transfer data yang berbeda secara geografis

Mengonfigurasi transfer file yang cepat dan aman menggunakan Amazon S3 Transfer Acceleration terbukti efektif dalam meminimalkan atau menghilangkan latensi yang disebabkan oleh jarak geografis antara klien yang tersebar secara global dan aplikasi regional menggunakan Amazon S3. Transfer Acceleration menggunakan lokasi edge yang didistribusikan secara global CloudFront untuk transportasi data. Jaringan AWS edge memiliki titik kehadiran di lebih dari 50 lokasi. Saat ini, digunakan untuk mendistribusikan konten melalui CloudFront dan untuk memberikan respons cepat terhadap kueri DNS yang dibuat ke Amazon Route 53.

Jaringan edge juga membantu mempercepat transfer data masuk dan keluar dari Amazon S3. Sangat ideal untuk aplikasi yang mentransfer data antar benua, memiliki koneksi internet yang cepat, menggunakan objek besar, atau memiliki banyak konten untuk diunggah. Saat datanya tiba di lokasi edge, data diarahkan ke Amazon S3 melalui jalur jaringan yang dioptimalkan. Secara umum, semakin jauh Anda berada di Wilayah Amazon S3, semakin tinggi peningkatan kecepatan yang dapat Anda harapkan dari penggunaan Transfer Acceleration.

Anda dapat mengatur Transfer Acceleration pada bucket baru atau yang sudah ada. Anda dapat menggunakan endpoint Amazon S3 Transfer Acceleration terpisah untuk menggunakan lokasi AWS tepi. Cara terbaik untuk menguji apakah Transfer Acceleration membantu klien meminta performa adalah menggunakan Alat Amazon S3 Transfer Acceleration Speed Comparison. Konfigurasi dan kondisi jaringan bervariasi dari waktu ke waktu dan dari lokasi ke lokasi. Jadi, Anda hanya dikenakan biaya untuk transfer ketika Amazon S3 Transfer Acceleration dapat meningkatkan performa pengunggahan Anda. Untuk informasi tentang penggunaan Akselerasi Transfer dengan yang berbeda AWS SDKs, lihatMengaktifkan dan menggunakan S3 Transfer Acceleration.

Hosting situs web statis menggunakan Amazon S3

Anda dapat menggunakan Amazon S3 untuk host situs web statis. Pada situs web statis, halaman web individu termasuk konten statis. Mungkin juga berisi skrip sisi klien.

> **Note**
>
> Kami menyarankan Anda menggunakan AWS Amplify Hosting untuk meng-host konten situs web statis yang disimpan di S3. Amplify Hosting adalah layanan yang dikelola sepenuhnya yang memudahkan penyebaran situs web Anda di jaringan pengiriman konten (CDN) yang tersedia secara global yang didukung oleh Amazon CloudFront, memungkinkan hosting situs web statis yang aman.
> Dengan AWS Amplify Hosting, Anda dapat memilih lokasi objek Anda dalam bucket tujuan umum Anda, menyebarkan konten Anda ke CDN terkelola, dan membuat URL HTTPS publik agar situs web Anda dapat diakses di mana saja. Untuk informasi selengkapnya tentang Amplify Hosting, lihat Menerapkan situs web statis ke AWS Amplify Hosting dari bucket tujuan umum S3 dan Menerapkan situs web statis dari S3 menggunakan konsol Amplify di Panduan Pengguna Konsol.AWS Amplify

Untuk informasi selengkapnya tentang hosting situs web statis di Amazon S3, termasuk petunjuk dan step-by-step penelusuran, lihat topik berikut.

Topik

- Titik akhir situs web
- Mengaktifkan hosting situs web
- Mengonfigurasi dokumen indeks
- Mengonfigurasi dokumen kesalahan khusus
- Mengatur izin untuk akses situs web
- (Opsional) Mencatat lalu lintas web
- (Opsional) Mengonfigurasi pengalihan halaman web
- Berbagi sumber daya lintas asal (CORS)
- Tutorial situs web statis

Titik akhir situs web

Saat Anda mengonfigurasi bucket sebagai situs web statis, situs web ini tersedia di titik akhir situs web khusus Wilayah AWS. Titik akhir situs web berbeda dari titik akhir saat Anda mengirimkan permintaan API REST. Untuk informasi selengkapnya tentang perbedaan antara titik akhir, lihat Perbedaan utama antara titik akhir situs web dan titik akhir API REST.

Bergantung pada Wilayah Anda, titik akhir situs web Amazon S3 Anda mengikuti salah satu dari dua format ini.

- s3-situs web (-) Wilayah - `http://bucket-name.s3-website-Region.amazonaws.com`
- s3-situs web dot (.) Wilayah-`http://bucket-name.s3-website.Region.amazonaws.com`

Ini URLs mengembalikan dokumen indeks default yang Anda konfigurasikan untuk situs web. Untuk daftar lengkap titik akhir situs web Amazon S3, lihat Titik Akhir Situs Web Amazon S3.

> **Note**
>
> Untuk meningkatkan keamanan situs web statis Amazon S3 Anda, domain titik akhir situs web Amazon S3 (misalnya, s3 website-us-east - -1.amazonaws.com atau s3-website.ap-south-1.amazonaws.com) terdaftar di Daftar Akhiran Publik (PSL). Untuk keamanan lebih lanjut, kami menyarankan Anda menggunakan cookie dengan prefiks `__Host-` jika Anda perlu mengatur cookie sensitif di nama domain untuk situs web statis Amazon S3 Anda. Praktik ini akan membantu mempertahankan domain Anda dari upaya pemalsuan permintaan lintas situs (CSRF). Untuk informasi selengkapnya, lihat halaman Set-Cookie di Jaringan Pengembang Mozilla.

Jika Anda ingin situs web Anda menjadi publik, Anda harus membuat semua konten Anda dapat dibaca publik agar pelanggan Anda dapat mengaksesnya di titik akhir situs web. Untuk informasi selengkapnya, lihat Mengatur izin untuk akses situs web.

> **Important**
>
> - Titik akhir situs web Amazon S3 tidak mendukung HTTPS atau titik akses. Jika Anda ingin menggunakan HTTPS, Anda dapat melakukan salah satu hal berikut:

- (Disarankan) Gunakan AWS Amplify Hosting untuk meng-host konten situs web statis yang disimpan di S3. Amplify Hosting adalah layanan yang dikelola sepenuhnya yang memudahkan penyebaran situs web Anda di jaringan pengiriman konten (CDN) yang tersedia secara global yang didukung oleh Amazon CloudFront, memungkinkan hosting situs web statis yang aman.

  Dengan AWS Amplify Hosting, Anda dapat memilih lokasi objek Anda dalam bucket tujuan umum Anda, menyebarkan konten Anda ke CDN terkelola, dan membuat URL HTTPS publik agar situs web Anda dapat diakses di mana saja. Untuk informasi selengkapnya tentang Amplify Hosting, lihat Menerapkan situs web statis ke AWS Amplify Hosting dari bucket tujuan umum S3 dan Menerapkan situs web statis dari S3 menggunakan konsol Amplify di Panduan Pengguna Konsol.AWS Amplify
- Gunakan Amazon CloudFront untuk melayani situs web statis yang dihosting di Amazon S3. Untuk informasi selengkapnya, lihat Bagaimana cara menggunakan CloudFront untuk melayani permintaan HTTPS untuk bucket Amazon S3 saya? Untuk menggunakan HTTPS dengan domain khusus, lihat Mengonfigurasi situs web statis menggunakan domain khusus yang terdaftar di Route 53.

- Bucket Peminta Membayar tidak mengizinkan akses melalui titik akhir situs web. Setiap permintaan ke bucket tersebut akan menerima respons 403 Akses Ditolak. Untuk informasi selengkapnya, lihat Menggunakan Requester Pays ember tujuan umum untuk transfer dan penggunaan penyimpanan.

Topik

- Contoh titik akhir situs web
- Menambahkan CNAME DNS
- Menggunakan domain khusus dengan Route 53
- Perbedaan utama antara titik akhir situs web dan titik akhir API REST

Contoh titik akhir situs web

Contoh berikut menunjukkan bagaimana Anda dapat mengakses bucket Amazon S3 yang dikonfigurasi sebagai situs web statis.

Example — Meminta objek di tingkat root

Untuk meminta objek tertentu yang disimpan di tingkat root dalam bucket, gunakan struktur URL berikut.

```
http://bucket-name.s3-website.Region.amazonaws.com/object-name
```

Misalnya, URL berikut meminta objek `photo.jpg` yang disimpan di tingkat root dalam bucket.

```
http://example-bucket.s3-website.us-west-2.amazonaws.com/photo.jpg
```

Example — Meminta satu objek dengan prefiks

Untuk meminta objek yang disimpan dalam folder di bucket Anda, gunakan struktur URL ini.

```
http://bucket-name.s3-website.Region.amazonaws.com/folder-name/object-name
```

URL berikut meminta objek `docs/doc1.html` tersebut dalam bucket Anda.

```
http://example-bucket.s3-website.us-west-2.amazonaws.com/docs/doc1.html
```

Menambahkan CNAME DNS

Jika Anda memiliki domain terdaftar, Anda dapat menambahkan entri CNAME DNS ke titik akhir situs web Amazon S3. Misalnya, jika Anda mendaftarkan domain `www.example-bucket.com`, Anda dapat membuat bucket `www.example-bucket.com`, dan menambahkan data CNAME DNS yang mengarah ke `www.example-bucket.com.s3-website.Region.amazonaws.com`. Semua permintaan untuk `http://www.example-bucket.com` dirutekan ke `www.example-bucket.com.s3-website.Region.amazonaws.com`.

Untuk informasi selengkapnya, lihat Menyesuaikan Amazon URLs S3 dengan catatan CNAME.

Menggunakan domain khusus dengan Route 53

Daripada mengakses situs web menggunakan titik akhir situs web Amazon S3, Anda dapat menggunakan domain Anda sendiri yang terdaftar di Amazon Route 53 untuk menyajikan konten Anda—misalnya, `example.com`. Anda dapat menggunakan Amazon S3 dengan Route 53 untuk menghosting situs web di domain root. Misalnya, jika Anda memiliki domain root `example.com` dan Anda menghosting situs web Anda di Amazon S3, pengunjung situs web Anda dapat mengakses

situs tersebut dari browser mereka dengan memasukkan salah satu `http://www.example.com` atau `http://example.com`.

Untuk panduan contoh, lihat Tutorial: Mengonfigurasi situs web statis menggunakan domain kustom yang terdaftar di Route 53.

Perbedaan utama antara titik akhir situs web dan titik akhir API REST

Titik akhir situs web Amazon S3 dioptimalkan untuk akses dari browser web. Tabel berikut merangkum perbedaan utama antara titik akhir API REST dan titik akhir situs web.

Perbedaan utama	Titik akhir API REST	Titik akhir situs web
Kontrol akses	Mendukung konten publik dan privat	Mendukung hanya konten yang dapat dibaca oleh publik
Penanganan pesan kesalahan	Mengembalikan respons kesalahan yang diformat XML	Mengembalikan dokumen HTML
Dukungan pengalihan	Tidak berlaku	Mendukung pengalihan tingkat objek dan tingkat bucket
Permintaan yang didukung	Mendukung semua operasi bucket dan objek	Hanya mendukung `GET` dan `HEAD` permintaan pada objek
Tanggapan `GET` dan `HEAD` permintaan di akar ember	Mengembalikan daftar kunci objek di dalam bucket	Mengembalikan dokumen indeks yang ditentukan dalam konfigurasi situs web
Dukungan Secure Sockets Layer (SSL)	Mendukung koneksi SSL	Tidak mendukung koneksi SSL

Untuk daftar lengkap titik akhir dan Wilayah Amazon S3, lihat Titik akhir dan kuota Amazon S3 di Referensi Umum AWS.

Mengaktifkan hosting situs web

Saat mengonfigurasi bucket sebagai situs web statis, Anda harus mengaktifkan hosting situs web statis, mengonfigurasi dokumen indeks, dan mengatur izin.

Anda dapat mengaktifkan hosting situs web statis menggunakan konsol Amazon S3, REST API, AWS SDKs, atau. AWS CLI AWS CloudFormation

Untuk mengonfigurasi situs web Anda dengan domain khusus, lihat Tutorial: Mengonfigurasi situs web statis menggunakan domain kustom yang terdaftar di Route 53.

Menggunakan konsol S3

Untuk mengaktifkan hosting situs web statis

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Dalam daftar bucket, pilih nama bucket yang ingin Anda aktifkan hosting situs web statis.
4. Pilih Properti.
5. Di bagian bawah Hosting situs web statis, pilih Edit.
6. Pilih Gunakan bucket ini untuk menghosting situs web.
7. Di bagian bawah Hosting situs web statis, pilih Aktifkan.
8. Di Dokumen indeks, masukkan nama file dokumen indeks, biasanya `index.html`.

   Nama dokumen indeks peka huruf besar/kecil, dan harus sama persis dengan nama file dokumen indeks HTML yang ingin Anda unggah ke bucket S3 Anda. Saat Anda mengonfigurasi bucket untuk hosting situs web, Anda harus menentukan dokumen indeks. Amazon S3 mengembalikan dokumen indeks ini ketika permintaan dibuat ke domain root atau subfolder mana pun. Untuk informasi selengkapnya, lihat Mengonfigurasi dokumen indeks.
9. Agar dapat menyediakan dokumen kesalahan kustom Anda sendiri untuk kesalahan kelas 4XX, di Dokumen kesalahan, masukkan nama file dokumen kesalahan kustom.

   Nama dokumen kesalahan peka huruf besar/kecil, dan harus sama persis dengan nama file dokumen indeks HTML yang ingin Anda unggah ke bucket S3 Anda. Jika Anda tidak menentukan dokumen kesalahan khusus dan terjadi kesalahan, Amazon S3 mengembalikan

dokumen kesalahan HTML default. Untuk informasi selengkapnya, lihat Mengonfigurasi dokumen kesalahan khusus.

10. (Opsional) Jika Anda ingin menentukan aturan pengalihan lanjutan, dalam Aturan pengalihan, masukkan JSON untuk menjelaskan aturannya.

    Misalnya, Anda dapat merutekan permintaan secara kondisional dengan nama kunci atau prefiks objek tertentu dalam permintaan tersebut. Untuk informasi selengkapnya, lihat Konfigurasikan aturan pengalihan untuk menggunakan pengalihan bersyarat lanjutan.

11. Pilih Simpan perubahan.

    Amazon S3 memungkinkan hosting situs web statis untuk bucket Anda. Di bagian bawah halaman, di bawah Hosting situs web statis, Anda melihat titik akhir situs web untuk bucket Anda.

12. Di bagian bawah Hosting situs web statis, perhatikan Titik Akhir.

    Titik Akhir adalah titik akhir situs web Amazon S3 untuk bucket Anda. Setelah Anda menyelesaikan konfigurasi bucket Anda sebagai situs web statis, Anda dapat menggunakan titik akhir ini untuk menguji situs web Anda.

Penggunaan API REST

Untuk informasi selengkapnya tentang mengirim permintaan REST secara langsung untuk mengaktifkan hosting situs web statis, lihat bagian berikut di Referensi API Amazon Simple Storage Service:

- PUT Bucket Situs web
- GET Bucket Situs web
- DELETE Bucket situs web

Menggunakan AWS SDKs

Untuk meng-host situs web statis di Amazon S3, Anda mengonfigurasi bucket Amazon S3 untuk menghosting situs web lalu mengunggah konten situs web Anda ke bucket. Anda juga dapat menggunakan AWS SDKs untuk membuat, memperbarui, dan menghapus konfigurasi situs web secara terprogram. SDKs Menyediakan kelas pembungkus di sekitar Amazon S3 REST API. Jika aplikasi Anda memerlukannya, Anda dapat mengirim permintaan API REST langsung dari aplikasi Anda.

.NET

Contoh berikut menunjukkan cara menggunakan konfigurasi AWS SDK untuk .NET untuk mengelola situs web untuk bucket. Untuk menambahkan konfigurasi situs web ke bucket, berikan nama bucket dan konfigurasi situs web. Konfigurasi situs web harus menyertakan dokumen indeks dan dapat berisi dokumen kesalahan opsional. Dokumen ini harus disimpan dalam bucket. Untuk informasi selengkapnya, lihat Situs web PUT Bucket. Untuk informasi selengkapnya tentang fitur situs web Amazon S3, lihat Hosting situs web statis menggunakan Amazon S3.

Contoh kode C# berikut menambahkan konfigurasi situs web ke bucket tertentu. Konfigurasi menentukan dokumen indeks dan nama dokumen kesalahan. Untuk informasi tentang menyiapkan dan menjalankan contoh kode, lihat Memulai SDK for .NET di AWSAWS SDK for .NET Developer Guide.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class WebsiteConfigTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string indexDocumentSuffix = "*** index object key ***"; //
 For example, index.html.
        private const string errorDocument = "*** error object key ***"; // For
 example, error.html.
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 client;
        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            AddWebsiteConfigurationAsync(bucketName, indexDocumentSuffix,
 errorDocument).Wait();
        }

        static async Task AddWebsiteConfigurationAsync(string bucketName,
                                            string indexDocumentSuffix,
```

```
                                            string errorDocument)
        {
            try
            {
                // 1. Put the website configuration.
                PutBucketWebsiteRequest putRequest = new PutBucketWebsiteRequest()
                {
                    BucketName = bucketName,
                    WebsiteConfiguration = new WebsiteConfiguration()
                    {
                        IndexDocumentSuffix = indexDocumentSuffix,
                        ErrorDocument = errorDocument
                    }
                };
                PutBucketWebsiteResponse response = await
 client.PutBucketWebsiteAsync(putRequest);

                // 2. Get the website configuration.
                GetBucketWebsiteRequest getRequest = new GetBucketWebsiteRequest()
                {
                    BucketName = bucketName
                };
                GetBucketWebsiteResponse getResponse = await
 client.GetBucketWebsiteAsync(getRequest);
                Console.WriteLine("Index document: {0}",
 getResponse.WebsiteConfiguration.IndexDocumentSuffix);
                Console.WriteLine("Error document: {0}",
 getResponse.WebsiteConfiguration.ErrorDocument);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
        }
    }
}
```

PHP

Contoh PHP berikut ini menambahkan konfigurasi situs web ke bucket yang ditentukan. Metode `create_website_config` secara eksplisit menyediakan nama dokumen indeks dan dokumen kesalahan. Contoh ini juga mengambil konfigurasi situs web dan mencetak respons. Untuk informasi selengkapnya tentang fitur situs web Amazon S3, lihat Hosting situs web statis menggunakan Amazon S3.

Untuk informasi selengkapnya tentang AWS SDK for Ruby API, AWS buka SDK for Ruby - Versi 2.

```
 require 'vendor/autoload.php';

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);


// Add the website configuration.
$s3->putBucketWebsite([
    'Bucket'                => $bucket,
    'WebsiteConfiguration'  => [
        'IndexDocument' => ['Suffix' => 'index.html'],
        'ErrorDocument' => ['Key' => 'error.html']
    ]
]);

// Retrieve the website configuration.
$result = $s3->getBucketWebsite([
    'Bucket' => $bucket
]);
echo $result->getPath('IndexDocument/Suffix');

// Delete the website configuration.
$s3->deleteBucketWebsite([
    'Bucket' => $bucket
]);
```

Menggunakan AWS CLI

Untuk informasi selengkapnya tentang penggunaan AWS CLI untuk mengonfigurasi bucket S3 sebagai situs web statis, lihat situs web di Referensi AWS CLI Perintah.

Selanjutnya, Anda harus mengonfigurasi dokumen indeks dan mengatur izin. Untuk informasi selengkapnya, lihat Mengonfigurasi dokumen indeks dan Mengatur izin untuk akses situs web.

Anda juga dapat secara opsional mengonfigurasi dokumen kesalahan, catatan lalu lintas web, atau arahkan ulang.

Mengonfigurasi dokumen indeks

Saat Anda mengaktifkan hosting situs web, Anda juga harus mengonfigurasi dan mengunggah dokumen indeks. Dokumen indeks adalah halaman web yang dikembalikan Amazon S3 ketika permintaan dibuat ke root situs web atau setiap subfolder. Misalnya, jika pengguna memasukkan `http://www.example.com` di browser, pengguna tidak meminta halaman tertentu. Dalam hal ini, Amazon S3 menyajikan dokumen indeks, yang kadang-kadang disebut sebagai halaman default.

Saat mengaktifkan hosting situs web statis untuk bucket Anda, Anda memasukkan nama dokumen indeks (misalnya, `index.html`). Setelah Anda mengaktifkan hosting situs web statis untuk bucket, Anda mengunggah file HTML dengan nama dokumen indeks ke bucket Anda.

Garis miring di akhir URL tingkat root bersifat opsional. Misalnya, jika Anda mengonfigurasi situs web Anda `index.html` sebagai dokumen indeks, salah satu dari URLs pengembalian berikut`index.html`.

```
http://example-bucket.s3-website.Region.amazonaws.com/
http://example-bucket.s3-website.Region.amazonaws.com
```

Untuk informasi selengkapnya tentang titik akhir situs web Amazon S3, lihat Titik akhir situs web.

Indeks dokumen dan folder

Di Amazon S3, bucket adalah kontainer datar objek. Bucket tidak menyediakan penyusunan hierarkis seperti halnya sistem file di komputer Anda. Namun, Anda dapat membuat hierarki logis dengan menggunakan nama kunci objek yang menyiratkan struktur folder.

Sebagai contoh, anggaplah bucket dengan tiga objek yang memiliki nama utama berikut. Meskipun ketiga objek ini disimpan tanpa organisasi hierarki fisik, Anda dapat menyimpulkan struktur folder logis berikut dari nama kunci:

- `sample1.jpg`—Objek berada di root bucket.
- `photos/2006/Jan/sample2.jpg`—Objek berada di subfolder `photos/2006/Jan`.
- `photos/2006/Feb/sample3.jpg`—Objek berada di subfolder `photos/2006/Feb`.

Di konsol Amazon S3, Anda juga dapat membuat folder di bucket. Misalnya, Anda dapat membuat folder dengan nama `photos`. Anda dapat mengunggah objek ke bucket atau ke folder `photos` di dalam bucket. Jika Anda menambahkan objek `sample.jpg` ke bucket, nama kunci adalah `sample.jpg`. Jika Anda meng-upload objek ke folder `photos`, nama kunci objek adalah `photos/sample.jpg`.

Jika Anda membuat struktur folder dalam bucket, Anda harus memiliki dokumen indeks di setiap tingkat. Dalam setiap folder, dokumen indeks harus mempunyai nama yang sama, sebagai contoh, `index.html`. Saat pengguna menentukan URL yang menyerupai pencarian folder, ada atau tidak adanya garis miring menentukan perilaku situs web. Misalnya, URL berikut, dengan garis miring, mengembalikan dokumen indeks `photos/index.html`.

```
http://bucket-name.s3-website.Region.amazonaws.com/photos/
```

Namun, jika Anda mengecualikan garis miring dari URL sebelumnya, Amazon S3 akan terlebih dahulu mencari objek `photos` dalam bucket. Jika objek `photos` tidak ditemukan, maka Amazon S3 akan mencari dokumen indeks, `photos/index.html`. Jika dokumen itu ditemukan, Amazon S3 mengembalikan pesan `302 Found` dan menunjuk ke kunci `photos/`. Untuk permintaan berikutnya ke `photos/`, Amazon S3 mengembalikan `photos/index.html`. Jika dokumen indeks tidak ditemukan, Amazon S3 mengembalikan kesalahan.

Mengonfigurasi dokumen indeks

Untuk mengonfigurasi dokumen indeks menggunakan konsol S3, gunakan prosedur berikut. Anda juga dapat mengonfigurasi dokumen indeks menggunakan REST API, the AWS SDKs, the AWS CLI, or AWS CloudFormation.

> **Note**
>
> Dalam bucket berkemampuan Penentuan Versi, Anda dapat mengunggah beberapa salinan `index.html`, tetapi hanya versi terbaru yang akan diselesaikan. Untuk informasi selengkapnya tentang Penentuan Versi S3, lihat Mempertahankan beberapa versi objek dengan S3 Versioning.

Saat mengaktifkan hosting situs web statis untuk bucket Anda, Anda memasukkan nama dokumen indeks (misalnya, **`index.html`**). Setelah Anda mengaktifkan hosting situs web statis untuk bucket, Anda mengunggah file HTML dengan nama dokumen indeks ke bucket Anda.

Untuk mengonfigurasi dokumen indeks

1. Buat file `index.html`.

   Jika Anda tidak memiliki file `index.html`, Anda dapat menggunakan HTML berikut ini untuk membuatnya:

   ```
   <html xmlns="http://www.w3.org/1999/xhtml" >
   <head>
       <title>My Website Home Page</title>
   </head>
   <body>
     <h1>Welcome to my website</h1>
     <p>Now hosted on Amazon S3!</p>
   </body>
   </html>
   ```

2. Simpan file indeks secara lokal.

   Nama file dokumen indeks harus sama persis dengan nama dokumen indeks yang Anda masukkan ke dalam kotak dialog Hosting situs web statis. Nama dokumen indeks peka huruf besar/kecil. Misalnya, jika Anda memasukkan `index.html` untuk Dokumen indeks dalam kotak dialog Hosting situs web statis, nama file dokumen indeks Anda juga harus dan bukan `Index.html`.

3. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

4. Di panel navigasi kiri, pilih Bucket tujuan umum.

5. Dalam daftar bucket, pilih nama bucket yang ingin Anda gunakan untuk meng-host situs web statis.
6. Aktifkan hosting situs web statis untuk bucket Anda, lalu masukkan nama persis dokumen indeks Anda (misalnya, `index.html`). Untuk informasi selengkapnya, lihat Mengaktifkan hosting situs web.

   Setelah mengaktifkan hosting situs web statis, lanjutkan ke langkah 6.
7. Untuk mengunggah dokumen indeks ke bucket Anda, lakukan salah satu hal berikut ini:

   - Seret dan jatuhkan file indeks ke dalam daftar bucket konsol.
   - Pilih Unggah, dan ikuti petunjuk untuk memilih dan mengunggah file indeks.

   Untuk step-by-step instruksi, lihatMengunggah Objek.
8. (Opsional) Mengunggah konten situs web lain ke bucket Anda.

Selanjutnya, Anda harus mengatur izin akses situs web. Untuk informasi, lihat Mengatur izin untuk akses situs web.

Anda juga dapat secara opsional mengonfigurasi dokumen kesalahan, catatan lalu lintas web, atau arahkan ulang.

Mengonfigurasi dokumen kesalahan khusus

Setelah Anda mengonfigurasi bucket Anda sebagai situs web statis, ketika kesalahan terjadi, Amazon S3 mengembalikan dokumen kesalahan HTML. Anda dapat secara opsional mengonfigurasi bucket Anda dengan dokumen kesalahan kustom sehingga Amazon S3 mengembalikan dokumen saat kesalahan terjadi.

> **Note**
>
> Beberapa browser menampilkan pesan kesalahan mereka sendiri saat terjadi kesalahan, mengabaikan dokumen kesalahan yang dikembalikan Amazon S3. Misalnya, ketika terjadi kesalahan HTTP 404 Not Found, Google Chrome mungkin mengabaikan dokumen kesalahan yang dikembalikan Amazon S3 dan menampilkan kesalahannya sendiri.

Topik

- Kode respons HTTP Amazon S3
- Mengonfigurasi dokumen kesalahan khusus

Kode respons HTTP Amazon S3

Tabel berikut mencantumkan subset kode respons HTTP yang dikembalikan Amazon S3 saat kesalahan terjadi.

Kode kesalahan HTTP	Deskripsi
301 Dipindahkan Secara Permanen	Saat pengguna mengirimkan permintaan secara langsung ke titik akhir situs web Amazon S3 (`http://s3-website. Region.amazonaws.com/`), Amazon S3 mengembalikan 301 Dipindahkan Secara Permanen menanggapi dan mengalihkan permintaan tersebut ke `https://aws.amazon.com/s3/`.
302 Ditemukan	Ketika Amazon S3 menerima permintaan untuk kunci `x`, `http://bucket-name .s3-website. Region.amazonaws.com/x` , tanpa garis miring, pertama-tama Amazon akan mencari objek dengan nama kunci `x`. Jika objek tidak ditemukan, Amazon S3 menentukan bahwa permintaan itu untuk subfolder `x` dan mengalihkan permintaan dengan menambahkan garis miring di akhir, dan mengembalikan 302 Ditemukan.
304 Tidak Dimodifikasi	Amazon S3 menggunakan header permintaan `If-Modified-Since` , `If-Unmodified-Since` , `If-Match` dan/atau `If-None-Match` untuk menentukan apakah objek yang diminta sama dengan salinan yang disimpan di cache yang dipegang oleh klien. Jika objek sama, titik akhir situs web mengembalikan 304 Tidak Dimodifikasi tanggapan mereka.
400 Permintaan Berformat Salah	Titik akhir situs web merespons dengan 400 Permintaan Berformat Salah saat pengguna mencoba mengakses bucket melalui titik akhir regional yang salah.
403 Dilarang	

Kode kesalahan HTTP	Deskripsi
	Titik akhir situs web merespons dengan 403 Dilarang ketika permintaan pengguna menerjemahkan ke objek yang tidak dapat dibaca oleh publik. Pemilik objek harus membuat objek dapat dibaca oleh publik menggunakan kebijakan bucket atau ACL.
404 Tidak Ditemukan	Titik akhir situs web merespons dengan 404 Tidak Ditemukan karena alasan berikut: • Amazon S3 menentukan bahwa URL situs web mengacu pada kunci objek yang tidak ada. • Amazon S3 menyimpulkan bahwa permintaan tersebut adalah untuk dokumen indeks yang tidak ada. • Bucket yang ditentukan dalam URL tidak tersedia. • Terdapat bucket yang disebutkan dalam URL, tetapi tidak dikonfigurasi sebagai situs web. Anda dapat membuat dokumen kustom yang dikembalikan untuk 404 Tidak Ditemukan. Pastikan dokumen diunggah ke bucket yang dikonfigurasi sebagai situs web, dan bahwa konfigurasi hosting situs web diatur untuk menggunakan dokumen. Untuk informasi tentang cara Amazon S3 menginterpretasikan URL sebagai permintaan objek atau dokumen indeks, lihat Mengonfigurasi dokumen indeks.
500 Kesalahan Layanan	Titik akhir situs web merespons dengan 500 Kesalahan Layanan saat terjadi kesalahan server internal.

Kode kesalahan HTTP	Deskripsi
503 Layanan Tidak Tersedia	Titik akhir situs web merespons dengan 503 Layanan Tidak Tersedia ketika Amazon S3 menentukan bahwa Anda perlu mengurangi tingkat permintaan Anda.

Untuk setiap kesalahan ini, Amazon S3 mengembalikan pesan HTML yang sudah ditentukan sebelumnya. Berikut ini adalah contoh pesan HTML yang dikembalikan untuk respons 403 Dilarang.

403 Forbidden

- Code: AccessDenied
- Message: Access Denied
- RequestId: 873CA367A51F7EC7
- HostId: DdQezl9vkuw5luD5HKsFaTDm9KH4PZzCPRkW3igimILbTu1DiYlvXjgyd7pVxq32

An Error Occurred While Attempting to Retrieve a Custom Error Document

- Code: AccessDenied
- Message: Access Denied

Mengonfigurasi dokumen kesalahan khusus

Saat Anda mengonfigurasi bucket sebagai situs web statis, Anda dapat memberikan dokumen kesalahan khusus yang berisi pesan kesalahan dan bantuan tambahan yang mudah digunakan. Amazon S3 mengembalikan dokumen kesalahan kustom Anda hanya untuk kode kesalahan kelas HTTP 4XX saja.

Untuk mengonfigurasi dokumen kesalahan kustom menggunakan konsol S3, ikuti langkah-langkah di bawah ini. Anda juga dapat mengonfigurasi dokumen kesalahan menggunakan REST API, the AWS SDKs, the AWS CLI, or AWS CloudFormation. Untuk informasi selengkapnya, lihat berikut ini:

- PutBucketWebsitedi Referensi API Layanan Penyimpanan Sederhana Amazon
- AWS::S3::Bucket WebsiteConfiguration di Panduan Pengguna AWS CloudFormation
- put-bucket-websitedalam Referensi AWS CLI Perintah

Saat mengaktifkan hosting situs web statis untuk bucket Anda, Anda memasukkan nama dokumen kesalahan (misalnya, **`404.html`**). Setelah Anda mengaktifkan hosting situs web statis untuk bucket, Anda mengunggah file HTML dengan nama dokumen indeks ke bucket Anda.

Untuk mengonfigurasi dokumen kesalahan

1. Membuat dokumen kesalahan, misalnya `404.html`.
2. Simpan file dokumen kesalahan secara lokal.

   Nama dokumen kesalahan peka huruf besar/kecil, dan harus sama persis dengan nama yang Anda masukkan saat Anda mengaktifkan hosting situs web statis. Misalnya, jika Anda memasukkan `404.html` sebagai nama Dokumen kesalahan di kotak dialog Hosting situs web statis, nama file dokumen kesalahan Anda juga harus bernama `404.html`.
3. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
4. Di panel navigasi kiri, pilih Bucket tujuan umum.
5. Dalam daftar bucket, pilih nama bucket yang ingin Anda gunakan untuk meng-host situs web statis.
6. Aktifkan hosting situs web statis untuk bucket Anda, lalu masukkan nama persis dokumen indeks Anda (misalnya, `404.html`). Untuk informasi selengkapnya, lihat Mengaktifkan hosting situs web dan Mengonfigurasi dokumen kesalahan khusus.

   Setelah mengaktifkan hosting situs web statis, lanjutkan ke langkah 6.
7. Untuk mengunggah dokumen kesalahan ke bucket Anda, lakukan salah satu hal berikut ini:

   - Seret dan jatuhkan file dokumen kesalahan ke dalam daftar bucket konsol.
   - Pilih Unggah, dan ikuti petunjuk untuk memilih dan mengunggah file indeks.

   Untuk step-by-step instruksi, lihatMengunggah Objek.

Mengatur izin untuk akses situs web

Saat Anda mengonfigurasikan bucket sebagai situs web statis, jika Anda ingin situs web Anda bersifat publik, Anda dapat memberikan akses baca publik. Agar bucket dapat dibaca publik, Anda harus menonaktifkan pengaturan akses publik untuk bucket dan menulis kebijakan bucket yang memberi akses baca publik. Jika bucket Anda berisi objek yang tidak dimiliki oleh pemilik bucket,

Anda mungkin juga perlu menambahkan daftar kontrol akses (ACL) objek yang memberi setiap orang akses baca.

Jika Anda tidak ingin menonaktifkan blokir pengaturan akses publik untuk bucket Anda tetapi Anda masih ingin situs web Anda menjadi publik, Anda dapat membuat CloudFront distribusi Amazon untuk melayani situs web statis Anda. Untuk informasi selengkapnya, lihat Mempercepat situs web Anda dengan Amazon CloudFront atau Menggunakan CloudFront distribusi Amazon untuk menayangkan situs web statis di Panduan Pengembang Amazon Route 53.

> **Note**
>
> Di titik akhir situs web, jika pengguna meminta objek yang tidak ada, Amazon S3 mengembalikan kode respons HTTP `404 (Not Found)`. Jika objek ada tetapi Anda belum memberikan izin baca padanya, titik akhir situs web mengembalikan kode respons HTTP `403 (Access Denied)`. Pengguna dapat menggunakan kode respons untuk menyimpulkan apakah objek tertentu ada. Jika Anda tidak menginginkan perilaku ini, Anda tidak boleh mengaktifkan dukungan situs web untuk bucket Anda.

Topik

- Langkat 1: Edit pengaturan Blokir Akses Publik S3
- Langkah 2: Menambahkan kebijakan bucket
- Daftar kontrol akses objek

Langkat 1: Edit pengaturan Blokir Akses Publik S3

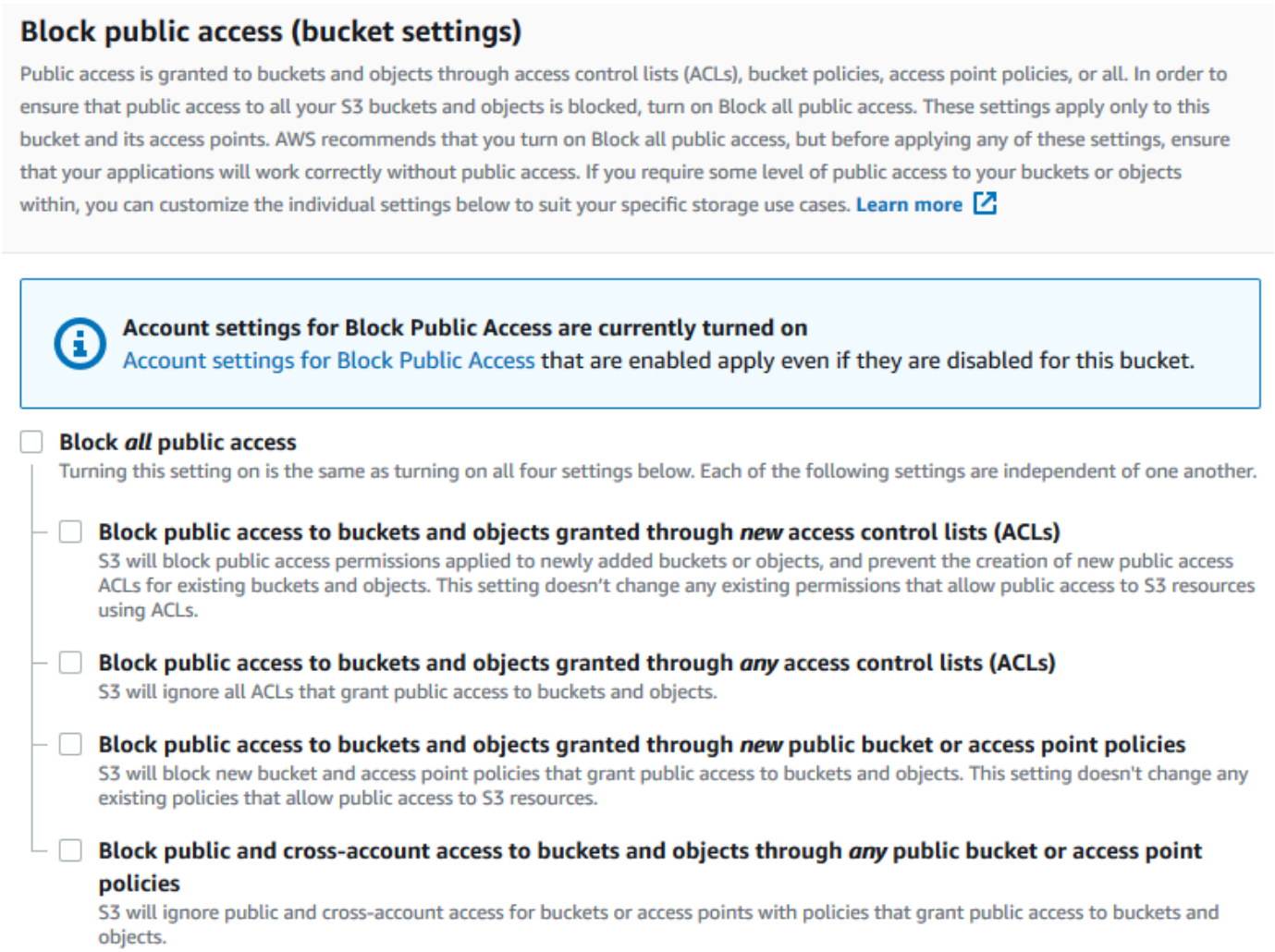
Jika Anda ingin mengonfigurasi bucket yang ada sebagai situs web statis yang memiliki akses publik, Anda harus mengedit pengaturan Blokir Akses Publik untuk bucket tersebut. Anda mungkin juga perlu mengedit pengaturan Blokir Akses Publik tingkat akun Anda. Amazon S3 menerapkan kombinasi paling ketat dari pengaturan akses publik tingkat bucket dan tingkat akun.

Misalnya, jika Anda mengizinkan akses publik untuk bucket tetapi memblokir semua akses publik di tingkat akun, Amazon S3 akan terus memblokir akses publik ke bucket. Dalam skenario ini, Anda harus mengedit pengaturan Blokir Akses Publik tingkat bucket dan tingkat akun Anda. Untuk informasi selengkapnya, lihat Melakukan blok akses publik ke penyimpanan Amazon S3 Anda.

Secara default, Amazon S3 memblokir akses publik ke akun dan bucket Anda. Jika Anda ingin menggunakan bucket untuk menghosting situs web statis, Anda dapat menggunakan langkah-langkah ini untuk mengedit pengaturan blokir akses publik Anda.

> **Warning**
>
> Sebelum Anda menyelesaikan langkah-langkah ini, tinjau Melakukan blok akses publik ke penyimpanan Amazon S3 Anda untuk memastikan bahwa Anda memahami dan menerima risiko yang terkait dengan mengizinkan akses publik. Saat Anda mematikan pengaturan blokir akses publik untuk membuat bucket Anda menjadi publik, siapa pun di internet dapat mengakses bucket Anda. Kami sarankan agar Anda memblokir semua akses publik ke bucket Anda.

1. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Pilih nama bucket yang telah Anda konfigurasikan sebagai situs web statis.
3. Pilih Izin.
4. Di bagian bawah Blokir akses publik (pengaturan bucket), pilih Edit.
5. Kosongkan Blokir semua akses publik, lalu pilih Simpan perubahan.



Amazon S3 mematikan pengaturan Blokir Akses Publik untuk bucket Anda. Untuk membuat situs web statis publik, Anda mungkin juga harus mengedit pengaturan Blokir Akses Publik untuk akun Anda sebelum menambahkan kebijakan bucket. Jika pengaturan Blokir Akses Publik untuk akun Anda saat ini diaktifkan, Anda akan melihat catatan di bawah Blokir akses publik (pengaturan bucket).

Langkah 2: Menambahkan kebijakan bucket

Untuk membuat objek dalam bucket Anda dapat dibaca publik, Anda harus menulis kebijakan bucket yang memberikan setiap orang izin `s3:GetObject`.

Setelah Anda mengedit pengaturan Blokir Akses Publik S3, Anda dapat menambahkan kebijakan bucket untuk memberikan akses baca publik ke bucket Anda. Saat Anda memberikan akses baca publik, siapa pun di internet dapat mengakses bucket Anda.

> **Important**
>
> Kebijakan berikut ini hanya merupakan contoh, dan memungkinkan akses penuh ke konten bucket Anda. Sebelum melanjutkan langkah ini, tinjau Bagaimana saya dapat mengamankan file dalam bucket Amazon S3 saya? untuk memastikan bahwa Anda telah memahami praktik terbaik untuk mengamankan file dalam bucket S3, dan risiko yang terlibat dalam pemberian akses publik.

1. Di bagian bawah Bucket, pilih nama bucket Anda.
2. Pilih Izin.
3. Di Bawah Kebijakan bucket, pilih Edit.
4. Untuk memberikan akses baca bagi publik untuk situs web Anda, salin kebijakan kelompok berikut, dan tempelkan di Editor kebijakan bucket.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::Bucket-Name/*"
            ]
        }
    ]
}
```

5. Perbarui `Resource` dengan nama bucket Anda.

   Dalam contoh kebijakan bucket sebelumnya, *Bucket-Name* adalah placeholder untuk nama bucket. Untuk menggunakan kebijakan bucket ini dengan bucket Anda sendiri, Anda harus memperbarui nama ini agar sesuai dengan nama bucket Anda.

6. Pilih Simpan perubahan.

Pesan akan muncul, yang menunjukkan bahwa kebijakan bucket telah berhasil ditambahkan.

Jika Anda melihat kesalahan yang mengatakan `Policy has invalid resource`, konfirmasikan bahwa nama bucket dalam kebijakan bucket tersebut sesuai dengan nama bucket Anda. Untuk informasi tentang menambahkan kebijakan bucket, lihat Bagaimana cara menambahkan kebijakan S3 bucket?

Jika Anda mendapatkan pesan kesalahan dan tidak dapat menyimpan kebijakan bucket, periksa pengaturan akun dan bucket Blokir Akses Publik untuk mengonfirmasi bahwa Anda mengizinkan akses publik ke bucket.

Daftar kontrol akses objek

Anda dapat menggunakan kebijakan bucket untuk memberikan izin baca publik kepada objek Anda. Namun, kebijakan bucket hanya berlaku untuk objek yang dimiliki oleh pemilik bucket. Apabila bucket Anda berisi objek yang bukan milik pemilik bucket, pemilik bucket harus menggunakan daftar kontrol akses (ACL) objek untuk memberikan izin BACA publik pada objek tersebut.

S3 Object Ownership adalah setelan tingkat ember Amazon S3 yang dapat Anda gunakan untuk mengontrol kepemilikan objek yang diunggah ke bucket dan untuk menonaktifkan atau mengaktifkan. ACLs Secara default, Kepemilikan Objek disetel ke setelan diberlakukan pemilik Bucket, dan semuanya ACLs dinonaktifkan. Ketika ACLs dinonaktifkan, pemilik bucket memiliki semua objek di bucket dan mengelola akses ke mereka secara eksklusif dengan menggunakan kebijakan manajemen akses.

Mayoritas kasus penggunaan modern di Amazon S3 tidak lagi memerlukan penggunaan. ACLs Kami menyarankan agar Anda tetap ACLs dinonaktifkan, kecuali dalam keadaan yang tidak biasa di mana Anda perlu mengontrol akses untuk setiap objek secara individual. Dengan ACLs dinonaktifkan, Anda dapat menggunakan kebijakan untuk mengontrol akses ke semua objek di bucket, terlepas dari siapa yang mengunggah objek ke bucket Anda. Untuk informasi selengkapnya, lihat Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.

> **Important**
>
> Jika bucket tujuan umum Anda menggunakan setelan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek S3, Anda harus menggunakan kebijakan untuk memberikan akses ke bucket tujuan umum dan objek di dalamnya. Dengan pengaturan yang diberlakukan pemilik Bucket diaktifkan, permintaan untuk menyetel daftar kontrol akses (ACLs) atau

pembaruan ACLs gagal dan mengembalikan kode `AccessControlListNotSupported` kesalahan. Permintaan untuk membaca ACLs masih didukung.

Untuk membuat objek yang dapat dibaca oleh publik dengan menggunakan ACL, berikan izin READ kepada kelompok `AllUsers`, sebagaimana ditunjukkan dalam elemen pemberian berikut. Tambahkan elemen pemberian ini ke objek ACL. Untuk informasi tentang mengelola ACLs, lihatGambaran umum daftar kontrol akses (ACL).

```
<Grant>
  <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xsi:type="Group">
    <URI>http://acs.amazonaws.com/groups/global/AllUsers</URI>
  </Grantee>
  <Permission>READ</Permission>
</Grant>
```

(Opsional) Mencatat lalu lintas web

Anda dapat secara opsional mengaktifkan pencatatan akses server Amazon S3 untuk bucket yang dikonfigurasi sebagai situs web statis. Pencatatan akses server menyediakan catatan terperinci untuk permintaan yang dilakukan ke bucket Anda. Untuk informasi selengkapnya, lihat Pencatatan permintaan dengan pencatatan akses server. Jika Anda berencana menggunakan Amazon CloudFront untuk mempercepat situs web Anda, Anda juga dapat menggunakan CloudFront logging. Untuk informasi selengkapnya, lihat Mengonfigurasi dan Menggunakan Log Akses di Panduan CloudFront Pengembang Amazon.

Untuk mengaktifkan log akses server untuk bucket situs web statis Anda

1. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di Wilayah yang sama tempat Anda membuat bucket yang dikonfigurasi sebagai situs web statis, buat bucket tujuan umum untuk pencatatan, misalnya`logs.example.com`.
3. Buat folder untuk file log pencatatan akses server (misalnya, `logs`).
4. (Opsional) Jika Anda ingin menggunakan CloudFront untuk meningkatkan kinerja situs web Anda, buat folder untuk file CloudFront log (misalnya,`cdn`).

   Untuk informasi selengkapnya, lihat Mempercepat situs web Anda dengan Amazon CloudFront.
5. Di daftar Bucket pilih bucket Anda.

6. Pilih Properti.
7. Di bagian bawah Pencatatan akses server, pilih Edit.
8. Pilih Aktifkan.
9. Under the Target bucket, choose the bucket and folder destination for the server access:

   - Jelajahi ke lokasi folder dan bucket:
     1. Pilih Jelajahi S3.
     2. Pilih nama bucket, lalu pilih folder log.
     3. Pilih Pilih jalur.
   - Masukkan alur bucket S3, misalnya, **`s3://logs.example.com/logs/`**.
10. Pilih Simpan perubahan.

    Dalam bucket log, sekarang dapat mengakses log Anda. Amazon S3 menulis log akses situs web ke bucket log Anda setiap 2 jam.

(Opsional) Mengonfigurasi pengalihan halaman web

Jika bucket Amazon S3 Anda dikonfigurasi untuk hosting situs web statis, Anda dapat mengonfigurasi pengalihan untuk bucket atau objek Anda di dalamnya. Anda memiliki opsi berikut untuk mengonfigurasikan pengalihan.

Topik

- Permintaan pengalihan untuk titik akhir situs web bucket Anda ke bucket atau domain lain
- Konfigurasikan aturan pengalihan untuk menggunakan pengalihan bersyarat lanjutan
- Mengalihkan permintaan untuk objek

Permintaan pengalihan untuk titik akhir situs web bucket Anda ke bucket atau domain lain

Anda dapat mengalihkan semua permintaan ke sebuah titik akhir situs web untuk sebuah bucket ke bucket atau domain yang lain. Jika Anda mengalihkan semua permintaan, permintaan apapun yang dibuat ke titik akhir situs web dialihkan ke bucket atau domain tertentu.

Misalnya, jika domain root Anda adalah `example.com`, dan Anda ingin melayani permintaan untuk keduanya `http://example.com` dan `http://www.example.com`, Anda harus membuat dua

bucket bernama `example.com` dan `www.example.com`. Kemudian, pertahankan konten di bucket `example.com`, dan konfigurasikan bucket `www.example.com` lainnya untuk mengalihkan semua permintaan ke bucket `example.com`. Untuk informasi selengkapnya, lihat Mengonfigurasi Situs Web Statis Menggunakan Nama Domain Khusus.

Untuk mengalihkan permintaan titik akhir situs web bucket

1. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di Bawah Bucket, pilih nama bucket tempat Anda ingin mengalihkan permintaan (misalnya, `www.example.com`).
3. Pilih Properti.
4. Di bagian bawah Hosting situs web statis, pilih Edit.
5. Pilih Alihkan permintaan objek.
6. Di kotak Nama host, masukkan titik akhir situs web untuk bucket atau domain khusus Anda.

   Misalnya, jika Anda mengalihkan ke alamat domain root, Anda akan memasukkan **`example.com`**.
7. Untuk Protokol, pilih protokol untuk permintaan pengalihan (tidak ada,http, atau https).

   Jika Anda tidak menentukan protokol, opsi default adalah tidak ada.
8. Pilih Simpan perubahan.

Konfigurasikan aturan pengalihan untuk menggunakan pengalihan bersyarat lanjutan

Menggunakan aturan pengalihan lanjutan, Anda dapat mengirimkan permintaan secara bersyarat sesuai dengan nama kunci, prefiks tertentu dalam permintaan, atau kode tanggapan. Misalnya, bayangkan bahwa Anda menghapus atau mengganti nama sebuah objek di dalam bucket Anda. Anda dapat menambahkan aturan perutean yang mengarahkan permintaan ke objek lain. Jika Anda ingin membuat folder menjadi tidak tersedia, Anda dapat menambahkan aturan perutean untuk mengalihkan permintaan ke halaman web lain. Anda juga dapat menambahkan aturan perutean untuk menangani kondisi kesalahan dengan mengirimkan permintaan yang mengembalikan kesalahan ke domain lain saat kesalahan diproses.

Saat mengaktifkan hosting situs web statis untuk bucket Anda, Anda dapat menentukan aturan pengalihan lanjutan secara opsional. Amazon S3 memiliki batasan 50 aturan perutean per konfigurasi

situs web. Jika Anda memerlukan lebih dari 50 aturan perutean, Anda dapat menggunakan pengalihan objek. Untuk informasi selengkapnya, lihat Menggunakan konsol S3.

Untuk informasi lebih lanjut tentang konfigurasi aturan perutean menggunakan API REST, lihat PutBucketWebsite dalam Referensi API Layanan Penyimpanan Sederhana Amazon.

> **Important**
>
> Untuk membuat aturan pengalihan di konsol Amazon S3, Anda harus menggunakan JSON. Untuk contoh JSON, lihat Contoh aturan pengalihan.

Untuk mengonfigurasi aturan pengalihan situs web statis

Untuk menambahkan aturan pengalihan ke bucket yang telah mengaktifkan hosting situs web statis, ikuti langkah-langkah ini.

1. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Dalam daftar bucket, pilih nama bucket yang telah Anda konfigurasikan sebagai situs web statis.
4. Pilih Properti.
5. Di bagian bawah Hosting situs web statis, pilih Edit.
6. Di kotak Aturan pengalihan, masukkan aturan pengalihan Anda di JSON.

   Pada konsol S3 Anda mendeskripsikan aturan menggunakan JSON. Untuk contoh JSON, lihat Contoh aturan pengalihan. Amazon S3 memiliki batasan 50 aturan perutean per konfigurasi situs web.
7. Pilih Simpan perubahan.

Elemen aturan perutean

Berikut ini adalah sintaks umum untuk menentukan aturan perutean dalam konfigurasi situs web di JSON dan XML Untuk mengonfigurasi aturan pengalihan di konsol S3 yang baru, Anda harus menggunakan JSON. Untuk contoh JSON, lihat Contoh aturan pengalihan.

JSON

```
[
```

```
    {
      "Condition": {
        "HttpErrorCodeReturnedEquals": "string",
        "KeyPrefixEquals": "string"
      },
      "Redirect": {
        "HostName": "string",
        "HttpRedirectCode": "string",
        "Protocol": "http"|"https",
        "ReplaceKeyPrefixWith": "string",
        "ReplaceKeyWith": "string"
      }
    }
  ]

Note: Redirect must each have at least one child element. You can have either
 ReplaceKeyPrefix with or ReplaceKeyWith but not both.
```

XML

```
<RoutingRules> =
    <RoutingRules>
         <RoutingRule>...</RoutingRule>
         [<RoutingRule>...</RoutingRule>
         ...]
    </RoutingRules>

<RoutingRule> =
   <RoutingRule>
      [ <Condition>...</Condition> ]
      <Redirect>...</Redirect>
   </RoutingRule>

<Condition> =
   <Condition>
      [ <KeyPrefixEquals>...</KeyPrefixEquals> ]
      [ <HttpErrorCodeReturnedEquals>...</HttpErrorCodeReturnedEquals> ]
   </Condition>
    Note: <Condition> must have at least one child element.

<Redirect> =
   <Redirect>
      [ <HostName>...</HostName> ]
```

```
      [ <Protocol>...</Protocol> ]
      [ <ReplaceKeyPrefixWith>...</ReplaceKeyPrefixWith>  ]
      [ <ReplaceKeyWith>...</ReplaceKeyWith> ]
      [ <HttpRedirectCode>...</HttpRedirectCode> ]
   </Redirect>

Note: <Redirect> must have at least one child element. You can have either
 ReplaceKeyPrefix with or ReplaceKeyWith but not both.
```

Tabel berikut menjelaskan elemen-elemen dalam aturan perutean.

Nama	Deskripsi
`RoutingRules`	Kontainer untuk pengumpulan `RoutingRule` yang berbeda.
`RoutingRule`	Aturan yang mengidentifikasi kondisi dan pengalihan yang diterapkan saat kondisi terpenuhi. Ketentuan: • Kontainer `RoutingRules` harus memuat setidaknya satu aturan perutean.
`Condition`	Kontainer untuk menjelaskan kondisi yang harus dipenuhi untuk pengalihan yang ditentukan untuk diterapkan. Jika aturan perutean tidak mencakup syarat, aturan diterapkan ke semua permintaan.
`KeyPrefixEquals`	Prefiks nama kunci objek yang menjadi tujuan pengalihan permintaan. `KeyPrefixEquals` diperlukan jika `HttpErrorCodeReturnedEquals` tidak ditentukan. Jika `KeyPrefixEquals` dan `HttpErrorCodeReturnedEquals` ditentukan, keduanya harus bernilai benar agar syarat dapat dipenuhi.

Nama	Deskripsi
`HttpErrorCodeReturnedEquals`	Kode kesalahan HTTP yang harus cocok agar pengalihan dapat diterapkan. Jika terjadi kesalahan, dan jika kode kesalahan memenuhi nilai ini, maka pengalihan yang ditentukan berlaku. `HttpErrorCodeReturnedEquals` diperlukan jika `KeyPrefixEquals` tidak ditentukan. Jika `KeyPrefixEquals` dan `HttpErrorCodeReturnedEquals` ditentukan, keduanya harus bernilai benar agar syarat dapat dipenuhi.
`Redirect`	Elemen kontainer yang memberikan instruksi untuk mengalihkan permintaan. Anda dapat mengarahkan permintaan ke halaman host lain atau halaman lainnya, atau Anda dapat menentukan protokol lain yang akan digunakan. `RoutingRule` harus memiliki elemen `Redirect`. Elemen `Redirect` harus mengandung setidaknya satu dari elemen saudara berikut: `Protocol`, `HostName`, `ReplaceKeyPrefixWith`, `ReplaceKeyWith`, atau `HttpRedirectCode`.
`Protocol`	Protokolnya, `http` atau `https`, untuk digunakan dalam `Location` yang dikembalikan di respons. Jika salah satu dari saudaranya mendapat persediaan, `Protocol` tidak diperlukan.
`HostName`	Nama host yang akan digunakan dalam header `Location` yang dikembalikan dalam respons. Jika salah satu dari saudaranya mendapat persediaan, `HostName` tidak diperlukan.

Nama	Deskripsi
`ReplaceKeyPrefixWith`	Prefiks nama kunci objek yang menggantikan nilai `KeyPrefixEquals` dalam permintaan pengalihan. Jika salah satu dari saudaranya mendapat persediaan, `ReplaceKeyPrefixWith` tidak diperlukan. Perangkat ini hanya dapat disediakan jika `ReplaceKeyWith` tidak disediakan.
`ReplaceKeyWith`	Kunci objek yang akan digunakan dalam header `Location` yang dikembalikan dalam respons. Jika salah satu dari saudaranya mendapat persediaan, `ReplaceKeyWith` tidak diperlukan. Perangkat ini hanya dapat disediakan jika `ReplaceKeyPrefixWith` tidak disediakan.
`HttpRedirectCode`	Kode pengalihan HTTP yang akan digunakan dalam header `Location` yang dikembalikan dalam respons. Jika salah satu dari saudaranya mendapat persediaan, `HttpRedirectCode` tidak diperlukan.

Contoh aturan pengalihan

Contoh berikut ini menjelaskan tugas pengalihan umum:

> **Important**
>
> Untuk membuat aturan pengalihan di konsol Amazon S3, Anda harus menggunakan JSON.

Example 1: Alihkan setelah mengganti nama sebuah prefiks kunci

Misalkan bucket Anda berisi objek berikut:

- index.html
- docs/article1.html

- docs/article2.html

Anda memutuskan untuk mengganti nama folder dari `docs/` ke `documents/`. Setelah Anda membuat perubahan ini, Anda harus mengalihkan permintaan untuk prefiks `docs/` ke `documents/`. Misalnya, permintaan untuk `docs/article1.html` akan dialihkan ke `documents/article1.html`.

Dalam hal ini, Anda menambahkan aturan perutean berikut ke konfigurasi situs web.

JSON

```
[
    {
        "Condition": {
            "KeyPrefixEquals": "docs/"
        },
        "Redirect": {
            "ReplaceKeyPrefixWith": "documents/"
        }
    }
]
```

XML

```
  <RoutingRules>
    <RoutingRule>
    <Condition>
      <KeyPrefixEquals>docs/</KeyPrefixEquals>
    </Condition>
    <Redirect>
      <ReplaceKeyPrefixWith>documents/</ReplaceKeyPrefixWith>
    </Redirect>
    </RoutingRule>
  </RoutingRules>
```

Example 2: Alihkan permintaan untuk folder yang dihapus ke sebuah halaman

Misalkan Anda menghapus folder `images/` (yaitu, Anda menghapus semua objek dengan prefiks kunci `images/`). Anda dapat menambahkan aturan perutean yang mengalihkan permintaan untuk objek apa pun dengan prefiks kunci `images/` ke halaman yang diberi nama `folderdeleted.html`.

JSON

```
[
    {
        "Condition": {
            "KeyPrefixEquals": "images/"
        },
        "Redirect": {
            "ReplaceKeyWith": "folderdeleted.html"
        }
    }
]
```

XML

```
  <RoutingRules>
    <RoutingRule>
    <Condition>
       <KeyPrefixEquals>images/</KeyPrefixEquals>
    </Condition>
    <Redirect>
      <ReplaceKeyWith>folderdeleted.html</ReplaceKeyWith>
    </Redirect>
    </RoutingRule>
  </RoutingRules>
```

Example 3: Alihkan untuk kesalahan HTTP

Misalkan ketika objek yang diminta tidak ditemukan, Anda ingin mengarahkan permintaan ke instance Amazon Elastic Compute Cloud EC2 (Amazon). Tambahkan aturan pengalihan sehingga ketika kode status HTTP 404 (Tidak Ditemukan) dikembalikan, pengunjung situs dialihkan ke EC2 instance Amazon yang menangani permintaan.

Contoh berikut juga memasukkan prefiks kunci objek `report-404/` dalam pengalihan. Misalnya, jika Anda meminta halaman `ExamplePage.html` dan menghasilkan kesalahan HTTP 404, permintaan tersebut dialihkan ke halaman `report-404/ExamplePage.html` pada instance Amazon EC2 yang ditentukan. Jika tidak ada aturan perutean dan terjadi kesalahan HTTP 404, dokumen kesalahan yang ditentukan dalam konfigurasi akan dikembalikan.

JSON

```
[
    {
        "Condition": {
            "HttpErrorCodeReturnedEquals": "404"
        },
        "Redirect": {
            "HostName": "ec2-11-22-333-44.compute-1.amazonaws.com",
            "ReplaceKeyPrefixWith": "report-404/"
        }
    }
]
```

XML

```
  <RoutingRules>
    <RoutingRule>
    <Condition>
      <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals >
    </Condition>
    <Redirect>
      <HostName>ec2-11-22-333-44.compute-1.amazonaws.com</HostName>
      <ReplaceKeyPrefixWith>report-404/</ReplaceKeyPrefixWith>
    </Redirect>
    </RoutingRule>
  </RoutingRules>
```

Mengalihkan permintaan untuk objek

Anda dapat mengalihkan permintaan untuk objek ke objek atau URL lain dengan mengatur lokasi pengalihan situs web di metadata objek tersebut. Anda mengatur pengalihan dengan menambahkan properti `x-amz-website-redirect-location` ke metadata objek tersebut. Di konsol Amazon S3, Anda menetapkan Lokasi Pengalihan Situs Web dalam metadata objek tersebut. Jika Anda menggunakan API Amazon S3, Anda mengatur `x-amz-website-redirect-location`. Halaman web kemudian mengartikan objek sebagai pengalihan 301.

Untuk mengarahkan permintaan ke objek lain, Anda menetapkan lokasi pengalihan ke kunci objek target. Untuk mengalihkan permintaan ke URL eksternal, Anda mengatur lokasi pengalihan ke URL

yang diinginkan. Untuk informasi selengkapnya tentang pemberian metadata objek, lihat Metadata objek yang ditentukan sistem.

Saat Anda mengatur suatu halaman, Anda dapat menyimpan atau menghapus konten objek sumber. Misalnya, jika Anda memiliki objek `page1.html` di bucket Anda, Anda dapat mengalihkan permintaan apa pun untuk halaman ini ke objek lain, `page2.html`. Anda memiliki dua opsi:

- Simpan konten `page1.html` dan alihkan permintaan halaman.
- Hapus konten `page1.html` dan unggah objek zero-byte dengan nama `page1.html` untuk mengganti objek yang ada dan mengarahkan kembali permintaan halaman.

Menggunakan konsol S3

1. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di daftar Bucket, pilih nama bucket yang telah Anda konfigurasikan sebagai situs web statis (misalnya, `example.com`).
3. Di Bawah Objek, pilih objek Anda.
4. Pilih Tindakan, lalu pilih Edit metadata.
5. Pilih Metadata.
6. Pilih Tambahkan Metadata.
7. Di Bawah Jenis, pilih Sistem Didefinisikan.
8. Di Key, pilih x-amz-website-redirect-location.
9. Di Nilai, masukkan nama kunci objek yang ingin Anda arahkan, misalnya, `/page2.html`.

   Untuk objek lain dalam bucket yang sama, prefiks `/` pada nilai wajib diisi. Anda juga dapat mengatur nilai ke URL eksternal, misalnya, `http://www.example.com`.
10. Pilih Edit metadata.

Penggunaan API REST

Tindakan API Amazon S3 berikut mendukung header `x-amz-website-redirect-location` dalam permintaan. Amazon S3 menyimpan nilai header di metadata objek sebagai `x-amz-website-redirect-location`.

- PUT Objek

- Mulai Unggahan Multibagian
- Objek POST
- PUT Objek-Salin

Bucket yang dikonfigurasi untuk hosting situs web memiliki titik akhir situs web dan titik akhir REST. Permintaan untuk halaman yang dikonfigurasi sebagai pengalihan 301 memiliki kemungkinan hasil berikut, tergantung pada titik akhir permintaan:

- Titik akhir situs web spesifik wilayah–Amazon S3 mengalihkan permintaan halaman sesuai dengan nilai properti `x-amz-website-redirect-location`.
- Titik akhir REST–Amazon S3 tidak mengalihkan permintaan halaman. Ini akan mengembalikan objek yang diminta.

Untuk informasi selengkapnya tentang titik akhir, lihat Perbedaan utama antara titik akhir situs web dan titik akhir API REST.

Saat mengatur pengalihan halaman, Anda dapat menyimpan atau menghapus konten objek. Misalnya, anggap bahwa Anda memiliki objek `page1.html` tersebut dalam bucket Anda.

- Untuk menyimpan konten `page1.html` dan hanya mengalihkan permintaan halaman, Anda dapat mengirimkan permintaan PUT Objek-Salin untuk membuat objek `page1.html` yang menggunakan objek `page1.html` yang sudah ada sebagai sumbernya. Dalam permintaan Anda, Anda menetapkan header `x-amz-website-redirect-location`. Ketika permintaan selesai, Anda akan memiliki halaman asli dengan konten yang tidak berubah, tetapi Amazon S3 akan mengalihkan semua permintaan halaman untuk halaman tersebut ke lokasi pengalihan yang Anda tentukan.
- Untuk menghapus konten objek `page1.html` dan mengalihkan permintaan untuk halaman, Anda dapat mengirim permintaan PUT Objek untuk mengunggah objek nol-byte yang memiliki kunci objek yang sama: `page1.html`. Dalam permintaan PUT, Anda menetapkan `x-amz-website-redirect-location` untuk `page1.html` ke objek baru. Setelah permintaan selesai, `page1.html` tidak memiliki konten, dan permintaan diarahkan kembali ke lokasi yang ditentukan oleh `x-amz-website-redirect-location`.

Saat Anda mengambil objek menggunakan tindakan GET Objek bersama dengan metadata objek lain, Amazon S3 mengembalikan header `x-amz-website-redirect-location` di dalam respons.

Berbagi sumber daya lintas asal (CORS)

Cross-origin resource sharing (CORS) menentukan cara aplikasi web klien yang dimuat di dalam satu domain untuk berinteraksi dengan sumber daya di domain yang berbeda. Dengan dukungan CORS, Anda dapat membangun aplikasi web sisi klien yang kaya dengan Amazon S3, dan secara selektif mengizinkan akses lintas asal ke sumber daya Amazon S3 Anda.

Bagian ini memberikan ikhtisar tentang CORS. Subtopik menjelaskan bagaimana Anda dapat mengaktifkan CORS menggunakan konsol Amazon S3, atau secara terprogram dengan menggunakan Amazon S3 REST API dan. AWS SDKs

Berbagi sumber daya lintas asal: Skenario kasus penggunaan

Berikut adalah contoh skenario untuk menggunakan CORS.

Skenario 1

Misalkan Anda menyelenggarakan situs web di bucket Amazon S3 dengan nama `website` sebagaimana dijelaskan dalam Hosting situs web statis menggunakan Amazon S3. Pengguna Anda memuat titik akhir situs web:

```
http://website.s3-website.us-east-1.amazonaws.com
```

Sekarang Anda ingin menggunakan JavaScript pada halaman web yang disimpan di bucket ini untuk dapat membuat permintaan GET dan PUT yang diautentikasi terhadap bucket yang sama dengan menggunakan titik akhir API Amazon S3 untuk bucket,. `website.s3.us-east-1.amazonaws.com` Browser biasanya akan memblokir JavaScript agar tidak mengizinkan permintaan tersebut, tetapi dengan CORS Anda dapat mengonfigurasi bucket Anda untuk secara eksplisit mengaktifkan permintaan lintas asal. `website.s3-website.us-east-1.amazonaws.com`

Skenario 2

Misalkan, Anda ingin menghosting font web dari bucket S3. Sekali lagi, browser memerlukan pemeriksaan CORS (juga disebut pemeriksaan preflight) untuk memuat font web. Anda harus mengonfigurasi bucket yang menghosting font web untuk memungkinkan asal dari mana pun untuk membuat permintaan ini.

Bagaimana Amazon S3 mengevaluasi konfigurasi CORS pada bucket?

Saat Amazon S3 menerima permintaan preflight dari browser, itu akan mengevaluasi konfigurasi CORS untuk bucket dan menggunakan aturan `CORSRule` pertama yang sesuai dengan permintaan browser yang masuk untuk mengaktifkan permintaan lintas asal. Agar aturan dapat dicocokkan, ketentuan berikut harus dipenuhi:

- `Origin`Header dalam permintaan CORS ke bucket Anda harus cocok dengan asal `AllowedOrigins` elemen dalam konfigurasi CORS Anda.
- Metode HTTP yang ditentukan dalam permintaan CORS ke bucket Anda harus cocok dengan metode atau metode yang tercantum dalam `AllowedMethods` elemen dalam konfigurasi CORS Anda. `Access-Control-Request-Method`
- Header yang tercantum di `Access-Control-Request-Headers` header dalam permintaan pra-penerbangan harus cocok dengan header dalam `AllowedHeaders` elemen dalam konfigurasi CORS Anda.

> **Note**
>
> Kebijakan ACLs dan kebijakan terus berlaku saat Anda mengaktifkan CORS di bucket Anda.

Bagaimana Titik Akses Lambda Objek mendukung CORS

Ketika S3 Lambda Objek menerima permintaan dari browser atau permintaannya menyertakan header `Origin`, S3 Lambda Objek selalu menambahkan bidang header `"AllowedOrigins":"*"`.

Untuk informasi lebih lanjut menggunakan CORS, lihat topik berikut.

Topik

- Elemen konfigurasi CORS
- Berbagi sumber daya lintas asal (CORS)
- Menguji CORS
- Penyelesaian masalah CORS

Elemen konfigurasi CORS

Untuk mengonfigurasi bucket Anda agar permintaan lintas asal dapat dilakukan, Anda membuat konfigurasi CORS. Konfigurasi CORS adalah dokumen dengan elemen yang mengidentifikasi asal yang akan Anda izinkan untuk mengakses bucket, operasi (metode HTTP) yang akan Anda dukung untuk setiap asal, dan informasi spesifik operasi lainnya. Anda dapat menambahkan hingga 100 aturan ke konfigurasi. Anda dapat menambahkan konfigurasi CORS sebagai sub-sumber daya `cors` ke bucket.

Jika Anda mengonfigurasi CORS di konsol S3, Anda harus menggunakan JSON untuk membuat konfigurasi CORS. Konsol S3 baru hanya mendukung konfigurasi JSON CORS.

Untuk informasi lebih lanjut tentang konfigurasi CORS dan unsur-unsur di dalamnya, lihat topik di bawah ini. Untuk petunjuk tentang cara menambahkan konfigurasi CORS, lihat Berbagi sumber daya lintas asal (CORS).

> **Important**
>
> Pada konsol S3, konfigurasi CORS harus merupakan JSON.

Topik

- Elemen AllowedMethods
- Elemen AllowedOrigins
- Elemen AllowedHeaders
- Elemen ExposeHeaders
- Elemen MaxAgeSeconds
- Contoh konfigurasi CORS

Elemen **`AllowedMethods`**

Dalam konfigurasi CORS, Anda dapat menentukan nilai berikut untuk elemen `AllowedMethods` lainnya.

- GET
- PUT

- POST
- HAPUS
- HEAD

Elemen **AllowedOrigins**

Di elemen `AllowedOrigins`, Anda menentukan asal yang ingin Anda izinkan permintaan lintas domainnya, misalnya, `http://www.example.com`. String asal hanya dapat berisi satu karakter wildcard `*`, seperti `http://*.example.com`. Anda secara opsional dapat menentukan `*` sebagai asal untuk mengaktifkan semua asal untuk mengirim permintaan lintas asal. Anda juga dapat menentukan `https` untuk hanya memungkinkan asal yang aman.

Elemen **AllowedHeaders**

Elemen `AllowedHeaders` menentukan header yang diizinkan dalam permintaan preflight melalui header `Access-Control-Request-Headers`. Setiap nama header di `Access-Control-Request-Headers` header harus cocok dengan entri yang sesuai dalam elemen. Amazon S3 hanya akan mengirimkan header yang diizinkan dalam respons yang diminta. Untuk daftar sampel header yang dapat digunakan dengan permintaan ke Amazon S3, kunjungi Header Permintaan Umum dalam panduan Referensi API Amazon Simple Storage Service.

Setiap AllowedHeaders string dalam konfigurasi Anda dapat berisi paling banyak satu karakter wildcard. Misalnya, `<AllowedHeader>x-amz-*</AllowedHeader>` akan mengaktifkan semua header kustom Amazon.

Elemen **ExposeHeaders**

Setiap `ExposeHeader` elemen mengidentifikasi header dalam respons yang Anda ingin pelanggan dapat mengakses dari aplikasi mereka (misalnya, dari JavaScript `XMLHttpRequest` objek).
Untuk daftar header respons Amazon S3 umum, kunjungi Header Respons Umum dalam panduan Referensi API Amazon Simple Storage Service.

Elemen **MaxAgeSeconds**

Elemen `MaxAgeSeconds` menentukan waktu dalam hitungan detik bahwa browser Anda dapat menyimpan respons untuk permintaan preflight sebagaimana diidentifikasi oleh sumber daya, metode HTTP, dan asal.

Contoh konfigurasi CORS

Alih-alih mengakses situs web dengan menggunakan titik akhir situs web Amazon S3, Anda dapat menggunakan domain Anda sendiri, seperti `example1.com` untuk melayani konten Anda. Untuk informasi tentang menggunakan domain Anda sendiri, lihat Tutorial: Mengonfigurasi situs web statis menggunakan domain kustom yang terdaftar di Route 53.

Contoh konfigurasi CORS berikut memiliki tiga aturan, yang ditentukan sebagai `CORSRule` elemen:

- Aturan pertama mengizinkan permintaan PUT, POST, dan DELETE lintas asal dari asal `http://www.example1.com`. Aturan ini juga memungkinkan semua header dalam permintaan OPSI preflight melalui header `Access-Control-Request-Headers`. Menanggapi permintaan OPTIONS preflight, Amazon S3 mengembalikan header yang diminta.
- Aturan kedua mengizinkan permintaan lintas asal yang sama seperti aturan pertama, tetapi aturan tersebut berlaku untuk asal lain, `http://www.example2.com`.
- Aturan ketiga memungkinkan permintaan GET lintas asal dari semua asal. Karakter wildcard * mengacu pada semua asal.

JSON

```
[
    {
        "AllowedHeaders": [
            "*"
        ],
        "AllowedMethods": [
            "PUT",
            "POST",
            "DELETE"
        ],
        "AllowedOrigins": [
            "http://www.example1.com"
        ],
        "ExposeHeaders": []
    },
    {
        "AllowedHeaders": [
            "*"
        ],
        "AllowedMethods": [
```

```
            "PUT",
            "POST",
            "DELETE"
        ],
        "AllowedOrigins": [
            "http://www.example2.com"
        ],
        "ExposeHeaders": []
    },
    {
        "AllowedHeaders": [],
        "AllowedMethods": [
            "GET"
        ],
        "AllowedOrigins": [
            "*"
        ],
        "ExposeHeaders": []
    }
]
```

XML

```
<CORSConfiguration>
 <CORSRule>
   <AllowedOrigin>http://www.example1.com</AllowedOrigin>

   <AllowedMethod>PUT</AllowedMethod>
   <AllowedMethod>POST</AllowedMethod>
   <AllowedMethod>DELETE</AllowedMethod>

   <AllowedHeader>*</AllowedHeader>
 </CORSRule>
 <CORSRule>
   <AllowedOrigin>http://www.example2.com</AllowedOrigin>

   <AllowedMethod>PUT</AllowedMethod>
   <AllowedMethod>POST</AllowedMethod>
   <AllowedMethod>DELETE</AllowedMethod>

   <AllowedHeader>*</AllowedHeader>
 </CORSRule>
 <CORSRule>
```

```
   <AllowedOrigin>*</AllowedOrigin>
   <AllowedMethod>GET</AllowedMethod>
 </CORSRule>
</CORSConfiguration>
```

Konfigurasi CORS juga memungkinkan parameter konfigurasi opsional, seperti ditunjukkan dalam konfigurasi CORS berikut. Dalam contoh ini, konfigurasi CORS memungkinkan permintaan PUT, POST, dan DELETE lintas asal dari asal `http://www.example.com`.

JSON

```
[
    {
        "AllowedHeaders": [
            "*"
        ],
        "AllowedMethods": [
            "PUT",
            "POST",
            "DELETE"
        ],
        "AllowedOrigins": [
            "http://www.example.com"
        ],
        "ExposeHeaders": [
            "x-amz-server-side-encryption",
            "x-amz-request-id",
            "x-amz-id-2"
        ],
        "MaxAgeSeconds": 3000
    }
]
```

XML

```
<CORSConfiguration>
 <CORSRule>
   <AllowedOrigin>http://www.example.com</AllowedOrigin>
   <AllowedMethod>PUT</AllowedMethod>
   <AllowedMethod>POST</AllowedMethod>
   <AllowedMethod>DELETE</AllowedMethod>
   <AllowedHeader>*</AllowedHeader>
```

```
  <MaxAgeSeconds>3000</MaxAgeSeconds>
  <ExposeHeader>x-amz-server-side-encryption</
ExposeHeader>
  <ExposeHeader>x-amz-request-id</
ExposeHeader>
  <ExposeHeader>x-amz-id-2</ExposeHeader>
 </CORSRule>
</CORSConfiguration>
```

Elemen `CORSRule` dalam konfigurasi sebelumnya mencakup elemen opsional berikut:

- `MaxAgeSeconds`—Menentukan jumlah waktu dalam detik (dalam contoh ini, 3000) saat browser menyimpan respons Amazon S3 terhadap permintaan OPTIONS preflight untuk sumber daya yang ditentukan. Dengan menyimpan respons, browser tidak perlu mengirim permintaan preflight ke Amazon S3 jika permintaan awal akan diulang.
- `ExposeHeaders`—Mengidentifikasi header respons (dalam contoh ini,, `x-amz-server-side-encryption`x-amz-request-id, dan`x-amz-id-2`) yang dapat diakses pelanggan dari aplikasi mereka (misalnya, dari objek). JavaScript `XMLHttpRequest`

Berbagi sumber daya lintas asal (CORS)

Cross-origin resource sharing (CORS) menentukan cara aplikasi web klien yang dimuat di dalam satu domain untuk berinteraksi dengan sumber daya di domain yang berbeda. Dengan dukungan CORS, Anda dapat membangun aplikasi web sisi klien yang kaya dengan Amazon S3, dan secara selektif mengizinkan akses lintas asal ke sumber daya Amazon S3 Anda.

Bagian ini menunjukkan kepada Anda cara mengaktifkan CORS menggunakan konsol Amazon S3, Amazon S3 REST API, dan. AWS SDKs Untuk mengonfigurasi bucket Anda untuk mengizinkan permintaan lintas asal, Anda menambahkan konfigurasi CORS ke bucket. Konfigurasi CORS adalah dokumen yang menetapkan aturan yang mengidentifikasi asal-usul yang akan Anda izinkan untuk mengakses bucket Anda, metode operasi (metode HTTP) yang didukung untuk setiap asal, dan informasi kustom operasi lainnya. Pada konsol S3, konfigurasi CORS harus dokumen JSON.

Misalnya konfigurasi CORS di JSON dan XL, lihat Elemen konfigurasi CORS.

Menggunakan konsol S3

Bagian ini menjelaskan cara menggunakan konsol Amazon S3 untuk menambahkan konfigurasi berbagi sumber daya lintas asal (CORS) ke bucket S3.

Saat Anda mengaktifkan CORS di bucket, daftar kontrol akses (ACLs) dan kebijakan izin akses lainnya terus berlaku.

> **Important**
>
> Pada konsol S3, konfigurasi CORS harus merupakan JSON. Untuk contoh konfigurasi CORS di JSON dan XML, lihat Elemen konfigurasi CORS.

Untuk menambahkan konfigurasi CORS ke bucket S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Dalam daftar bucket, pilih nama bucket yang ingin Anda buat kebijakan bucket.
4. Pilih Izin.
5. Di bagian Berbagi sumber daya lintas asal (CORS), pilih Edit.
6. Di kotak teks Editor konfigurasi CORS, ketik atau salin dan tempel konfigurasi CORS baru, atau edit konfigurasi yang ada.

   Konfigurasi CORS adalah file JSON. Teks yang Anda ketikkan di editor harus valid JSON. Untuk informasi selengkapnya, lihat Elemen konfigurasi CORS.
7. Pilih Simpan perubahan.

   > **Note**
   >
   > Amazon S3 menampilkan Amazon Resource Name (ARN) untuk bucket di samping judul Editor konfigurasi CORS. Untuk informasi selengkapnya ARNs, lihat Nama Sumber Daya Amazon (ARNs) dan Ruang Nama AWS Layanan di. Referensi Umum Amazon Web Services

Menggunakan AWS SDKs

Anda dapat menggunakan AWS SDK untuk mengelola berbagi sumber daya lintas asal (CORS) untuk bucket. Untuk informasi selengkapnya tentang CORS, lihat Berbagi sumber daya lintas asal (CORS).

Contoh berikut:

- Membuat konfigurasi CORS dan mengatur konfigurasi pada bucket
- Membuka kembali konfigurasi dan memodifikasinya dengan menambahkan aturan
- Menambahkan konfigurasi yang telah dimodifikasi ke bucket
- Menghapus konfigurasi

Java

Example

Example

Untuk petunjuk tentang cara membuat dan menguji sampel kerja, lihat Memulai di Panduan AWS SDK untuk Java Pengembang.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.BucketCrossOriginConfiguration;
import com.amazonaws.services.s3.model.CORSRule;

import java.io.IOException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class CORS {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";

        // Create two CORS rules.
        List<CORSRule.AllowedMethods> rule1AM = new
 ArrayList<CORSRule.AllowedMethods>();
        rule1AM.add(CORSRule.AllowedMethods.PUT);
        rule1AM.add(CORSRule.AllowedMethods.POST);
```

```
        rule1AM.add(CORSRule.AllowedMethods.DELETE);
        CORSRule rule1 = new
CORSRule().withId("CORSRule1").withAllowedMethods(rule1AM)
                .withAllowedOrigins(Arrays.asList("http://*.example.com"));

        List<CORSRule.AllowedMethods> rule2AM = new
ArrayList<CORSRule.AllowedMethods>();
        rule2AM.add(CORSRule.AllowedMethods.GET);
        CORSRule rule2 = new
CORSRule().withId("CORSRule2").withAllowedMethods(rule2AM)
                .withAllowedOrigins(Arrays.asList("*")).withMaxAgeSeconds(3000)
                .withExposedHeaders(Arrays.asList("x-amz-server-side-encryption"));

        List<CORSRule> rules = new ArrayList<CORSRule>();
        rules.add(rule1);
        rules.add(rule2);

        // Add the rules to a new CORS configuration.
        BucketCrossOriginConfiguration configuration = new
BucketCrossOriginConfiguration();
        configuration.setRules(rules);

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            // Add the configuration to the bucket.
            s3Client.setBucketCrossOriginConfiguration(bucketName, configuration);

            // Retrieve and display the configuration.
            configuration = s3Client.getBucketCrossOriginConfiguration(bucketName);
            printCORSConfiguration(configuration);

            // Add another new rule.
            List<CORSRule.AllowedMethods> rule3AM = new
ArrayList<CORSRule.AllowedMethods>();
            rule3AM.add(CORSRule.AllowedMethods.HEAD);
            CORSRule rule3 = new
CORSRule().withId("CORSRule3").withAllowedMethods(rule3AM)
                    .withAllowedOrigins(Arrays.asList("http://www.example.com"));

            rules = configuration.getRules();
```

```
            rules.add(rule3);
            configuration.setRules(rules);
            s3Client.setBucketCrossOriginConfiguration(bucketName, configuration);

            // Verify that the new rule was added by checking the number of rules in
the
            // configuration.
            configuration = s3Client.getBucketCrossOriginConfiguration(bucketName);
            System.out.println("Expected # of rules = 3, found " +
configuration.getRules().size());

            // Delete the configuration.
            s3Client.deleteBucketCrossOriginConfiguration(bucketName);
            System.out.println("Removed CORS configuration.");

            // Retrieve and display the configuration to verify that it was
            // successfully deleted.
            configuration = s3Client.getBucketCrossOriginConfiguration(bucketName);
            printCORSConfiguration(configuration);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }

    private static void printCORSConfiguration(BucketCrossOriginConfiguration
configuration) {
        if (configuration == null) {
            System.out.println("Configuration is null.");
        } else {
            System.out.println("Configuration has " +
configuration.getRules().size() + " rules\n");

            for (CORSRule rule : configuration.getRules()) {
                System.out.println("Rule ID: " + rule.getId());
                System.out.println("MaxAgeSeconds: " + rule.getMaxAgeSeconds());
                System.out.println("AllowedMethod: " + rule.getAllowedMethods());
                System.out.println("AllowedOrigins: " + rule.getAllowedOrigins());
                System.out.println("AllowedHeaders: " + rule.getAllowedHeaders());
```

```
                System.out.println("ExposeHeader: " + rule.getExposedHeaders());
                System.out.println();
            }
        }
    }
}
```

.NET

Example

Untuk informasi tentang menyiapkan dan menjalankan contoh kode, lihat Memulai SDK for .NET di AWSAWS SDK for .NET Developer Guide.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class CORSTest
    {
        private const string bucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            CORSConfigTestAsync().Wait();
        }
        private static async Task CORSConfigTestAsync()
        {
            try
            {
                // Create a new configuration request and add two rules
                CORSConfiguration configuration = new CORSConfiguration
                {
```

```
                    Rules = new System.Collections.Generic.List<CORSRule>
                        {
                          new CORSRule
                          {
                            Id = "CORSRule1",
                            AllowedMethods = new List<string> {"PUT", "POST",
 "DELETE"},
                            AllowedOrigins = new List<string> {"http://
*.example.com"}
                          },
                          new CORSRule
                          {
                            Id = "CORSRule2",
                            AllowedMethods = new List<string> {"GET"},
                            AllowedOrigins = new List<string> {"*"},
                            MaxAgeSeconds = 3000,
                            ExposeHeaders = new List<string> {"x-amz-server-side-
encryption"}
                          }
                        }
                };

                // Add the configuration to the bucket.
                await PutCORSConfigurationAsync(configuration);

                // Retrieve an existing configuration.
                configuration = await RetrieveCORSConfigurationAsync();

                // Add a new rule.
                configuration.Rules.Add(new CORSRule
                {
                    Id = "CORSRule3",
                    AllowedMethods = new List<string> { "HEAD" },
                    AllowedOrigins = new List<string> { "http://www.example.com" }
                });

                // Add the configuration to the bucket.
                await PutCORSConfigurationAsync(configuration);

                // Verify that there are now three rules.
                configuration = await RetrieveCORSConfigurationAsync();
                Console.WriteLine();
                Console.WriteLine("Expected # of rulest=3; found:{0}",
 configuration.Rules.Count);
```

```
                Console.WriteLine();
                Console.WriteLine("Pause before configuration delete. To continue,
click Enter...");
                Console.ReadKey();

                // Delete the configuration.
                await DeleteCORSConfigurationAsync();

                // Retrieve a nonexistent configuration.
                configuration = await RetrieveCORSConfigurationAsync();
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
        }

        static async Task PutCORSConfigurationAsync(CORSConfiguration configuration)
        {

            PutCORSConfigurationRequest request = new PutCORSConfigurationRequest
            {
                BucketName = bucketName,
                Configuration = configuration
            };

            var response = await s3Client.PutCORSConfigurationAsync(request);
        }

        static async Task<CORSConfiguration> RetrieveCORSConfigurationAsync()
        {
            GetCORSConfigurationRequest request = new GetCORSConfigurationRequest
            {
                BucketName = bucketName

            };
            var response = await s3Client.GetCORSConfigurationAsync(request);
            var configuration = response.Configuration;
```

```
            PrintCORSRules(configuration);
            return configuration;
        }

        static async Task DeleteCORSConfigurationAsync()
        {
            DeleteCORSConfigurationRequest request = new
 DeleteCORSConfigurationRequest
            {
                BucketName = bucketName
            };
            await s3Client.DeleteCORSConfigurationAsync(request);
        }

        static void PrintCORSRules(CORSConfiguration configuration)
        {
            Console.WriteLine();

            if (configuration == null)
            {
                Console.WriteLine("\nConfiguration is null");
                return;
            }

            Console.WriteLine("Configuration has {0} rules:",
 configuration.Rules.Count);
            foreach (CORSRule rule in configuration.Rules)
            {
                Console.WriteLine("Rule ID: {0}", rule.Id);
                Console.WriteLine("MaxAgeSeconds: {0}", rule.MaxAgeSeconds);
                Console.WriteLine("AllowedMethod: {0}", string.Join(", ",
 rule.AllowedMethods.ToArray()));
                Console.WriteLine("AllowedOrigins: {0}", string.Join(", ",
 rule.AllowedOrigins.ToArray()));
                Console.WriteLine("AllowedHeaders: {0}", string.Join(", ",
 rule.AllowedHeaders.ToArray()));
                Console.WriteLine("ExposeHeader: {0}", string.Join(", ",
 rule.ExposeHeaders.ToArray()));
            }
        }
    }
}
```

Penggunaan API REST

Untuk mengatur konfigurasi CORS pada bucket Anda, Anda dapat menggunakan AWS Management Console. Jika aplikasi Anda memerlukannya, Anda juga dapat mengirim permintaan REST secara langsung. Bagian berikut dalam Referensi API Amazon Simple Storage Service menguraikan tindakan API REST terkait konfigurasi CORS:

- PutBucketCors
- GetBucketCors
- DeleteBucketCors
- OPTIONS object

Menguji CORS

Untuk menguji konfigurasi CORS Anda, permintaan preflight CORS dapat dikirim dengan `OPTIONS` metode sehingga server dapat merespons jika dapat diterima untuk mengirim permintaan. Saat Amazon S3 menerima permintaan preflight, S3 mengevaluasi konfigurasi CORS untuk bucket dan menggunakan `CORSRule` aturan pertama yang cocok dengan permintaan masuk untuk mengaktifkan permintaan lintas asal. Agar aturan dapat dicocokkan, ketentuan berikut harus dipenuhi:

- `Origin`Header dalam permintaan CORS ke bucket Anda harus cocok dengan asal `AllowedOrigins` elemen dalam konfigurasi CORS Anda.
- Metode HTTP yang ditentukan dalam permintaan CORS ke bucket Anda harus cocok dengan metode atau metode yang tercantum dalam `AllowedMethods` elemen dalam konfigurasi CORS Anda. `Access-Control-Request-Method`
- Header yang tercantum di `Access-Control-Request-Headers` header dalam permintaan preflight harus cocok dengan header dalam `AllowedHeaders` elemen dalam konfigurasi CORS Anda.

Berikut ini adalah contoh konfigurasi CORS. Untuk membuat Konfigurasi CORS, lihat Mengkonfigurasi CORS. Untuk contoh konfigurasi CORS lainnya, lihat Elemen konfigurasi CORS.

Untuk panduan tentang mengonfigurasi dan memecahkan masalah aturan CORS, lihat Bagaimana cara mengonfigurasi CORS di Amazon S3 dan mengonfirmasi aturan CORS menggunakan cURL? di pusat AWS re:Post pengetahuan.

JSON

```
[
    {
        "AllowedHeaders": [
            "Authorization"
        ],
        "AllowedMethods": [
            "GET",
            "PUT",
            "POST",
            "DELETE"
        ],
        "AllowedOrigins": [
            "http://www.example1.com"
        ],
        "ExposeHeaders":  [
             "x-amz-meta-custom-header"
        ]

    }
]
```

Untuk menguji konfigurasi CORS, Anda dapat mengirim `OPTIONS` cek preflight dengan menggunakan perintah CURL berikut. CURL adalah alat baris perintah yang dapat digunakan untuk berinteraksi dengan S3. Untuk informasi lebih lanjut, lihat CURL.

```
curl -v -X OPTIONS \
 -H "Origin: http://www.example1.com" \
 -H "Access-Control-Request-Method: PUT" \
 -H "Access-Control-Request-Headers: Authorization" \
 -H "Access-Control-Expose-Headers: x-amz-meta-custom-header"\
    "http://bucket_name.s3.amazonaws.com/object_prefix_name"
```

Dalam contoh di atas, `curl -v -x OPTIONS` perintah digunakan untuk mengirim permintaan preflight ke S3 untuk menanyakan apakah diizinkan oleh S3 untuk mengirim `PUT` permintaan pada objek dari asal silang. `http://www.example1.com` Header `Access-Control-Request-Headers` dan `Access-Control-Expose-Headers` opsional.

- Menanggapi `Access-Control-Request-Method` header dalam `OPTIONS` permintaan preflight, Amazon S3 mengembalikan daftar metode yang diizinkan jika metode yang diminta cocok.
- Menanggapi `Access-Control-Request-Headers` header dalam `OPTIONS` permintaan preflight, Amazon S3 mengembalikan daftar header yang diizinkan jika header yang diminta cocok.
- Menanggapi `Access-Control-Expose-Headers` header dalam `OPTIONS` permintaan preflight, Amazon S3 mengembalikan daftar header yang diizinkan jika header yang diminta cocok dengan header yang diizinkan yang dapat diakses oleh skrip yang berjalan di browser.

> **Note**
>
> Saat mengirim permintaan preflight, jika salah satu header permintaan CORS tidak diizinkan, tidak ada header CORS respons yang dikembalikan.

Menanggapi `OPTIONS` permintaan preflight ini, Anda akan menerima `200 OK` tanggapan. Untuk kode kesalahan umum yang diterima saat menguji CORS dan informasi selengkapnya untuk mengatasi masalah terkait CORS, lihat Pemecahan Masalah CORS.

```
< HTTP/1.1 200 OK
< Date: Fri, 12 Jul 2024 00:23:51 GMT
< Access-Control-Allow-Origin: http://www.example1.com
< Access-Control-Allow-Methods: GET, PUT, POST, DELETE
< Access-Control-Allow-Headers: Authorization
< Access-Control-Expose-Headers: x-amz-meta-custom-header
< Access-Control-Allow-Credentials: true
< Vary: Origin, Access-Control-Request-Headers, Access-Control-Request-Method
< Server: AmazonS3
< Content-Length: 0
```

Penyelesaian masalah CORS

Topik berikut dapat membantu Anda memecahkan masalah beberapa masalah CORS umum yang terkait dengan S3.

Topik

- 403 Kesalahan terlarang: CORS tidak diaktifkan untuk ember ini
- 403 Kesalahan terlarang: Permintaan CORS ini tidak diizinkan

- Header tidak ditemukan dalam respons CORS
- Pertimbangan CORS pada integrasi proxy S3

403 Kesalahan terlarang: CORS tidak diaktifkan untuk ember ini

`403 Forbidden`Kesalahan berikut terjadi saat permintaan lintas asal dikirim ke Amazon S3 tetapi CORS tidak dikonfigurasi pada bucket S3 Anda.

Kesalahan: HTTP/1.1 403 Respons CORS Terlarang: CORS tidak diaktifkan untuk bucket ini.

Konfigurasi CORS adalah dokumen atau kebijakan dengan aturan yang mengidentifikasi asal yang Anda izinkan untuk mengakses bucket, operasi (metode HTTP) yang akan Anda dukung untuk setiap asal, dan informasi spesifik operasi lainnya. Lihat cara mengonfigurasi CORS di S3 dengan menggunakan konsol Amazon S3 AWS SDKs, dan REST API. Untuk informasi lebih lanjut tentang CORS dan contoh konfigurasi CORS, lihat Elemen CORS.

403 Kesalahan terlarang: Permintaan CORS ini tidak diizinkan

`403 Forbidden`Kesalahan berikut diterima ketika aturan CORS dalam konfigurasi CORS Anda tidak cocok dengan data dalam permintaan Anda.

Kesalahan: HTTP/1.1 403 Tanggapan CORS Terlarang: Permintaan CORS ini tidak diizinkan.

Akibatnya, `403 Forbidden` kesalahan ini dapat terjadi karena berbagai alasan:

- Asal tidak diperbolehkan.
- Metode tidak diperbolehkan.
- Header yang diminta tidak diperbolehkan.

Untuk setiap permintaan yang diterima Amazon S3, Anda harus memiliki aturan CORS dalam konfigurasi CORS yang cocok dengan data dalam permintaan Anda.

Asal tidak diperbolehkan

`Origin`Header dalam permintaan CORS ke bucket Anda harus cocok dengan asal `AllowedOrigins` elemen dalam konfigurasi CORS Anda. Karakter wildcard (`"*"`) dalam `AllowedOrigins` elemen akan cocok dengan semua metode HTTP. Untuk informasi selengkapnya tentang cara memperbarui `AllowedOrigins` elemen, lihat Mengonfigurasi berbagi sumber daya lintas asal (CORS).

Misalnya, jika hanya `http://www.example1.com` domain yang disertakan dalam `AllowedOrigins` elemen, maka permintaan CORS yang dikirim dari `http://www.example2.com` domain akan menerima `403 Forbidden` kesalahan.

Contoh berikut menunjukkan bagian dari konfigurasi CORS yang menyertakan `http://www.example1.com` domain dalam `AllowedOrigins` elemen.

```
"AllowedOrigins":[
   "http://www.example1.com"
]
```

Agar permintaan CORS yang dikirim dari `http://www.example2.com` domain berhasil, `http://www.example2.com` domain harus disertakan dalam `AllowedOrigins` elemen konfigurasi CORS.

```
"AllowedOrigins":[
   "http://www.example1.com"
   "http://www.example2.com"
]
```

Metode tidak diperbolehkan

Metode HTTP yang ditentukan dalam permintaan CORS ke bucket Anda harus cocok dengan metode atau metode yang tercantum dalam `AllowedMethods` elemen dalam konfigurasi CORS Anda. `Access-Control-Request-Method` Karakter wildcard (`"*"`) di `AllowedMethods` akan cocok dengan semua metode HTTP. Untuk informasi selengkapnya tentang cara memperbarui `AllowedOrigins` elemen, lihat Mengonfigurasi berbagi sumber daya lintas asal (CORS).

Dalam konfigurasi CORS, Anda dapat menentukan metode berikut dalam `AllowedMethods` elemen:

- `GET`
- `PUT`
- `POST`
- `DELETE`
- `HEAD`

Contoh berikut menunjukkan bagian dari konfigurasi CORS yang mencakup `GET` metode dalam `AllowedMethods` elemen. Hanya permintaan termasuk `GET` metode yang akan berhasil.

```
"AllowedMethods":[
   "GET"
]
```

Jika metode HTTP (misalnya,PUT) digunakan dalam permintaan CORS atau disertakan dalam permintaan CORS pra-penerbangan ke bucket Anda tetapi metode tersebut tidak ada dalam konfigurasi CORS Anda, permintaan tersebut akan menghasilkan kesalahan. `403 Forbidden` Untuk mengizinkan permintaan CORS atau permintaan pra-penerbangan CORS ini, `PUT` metode harus ditambahkan ke konfigurasi CORS Anda.

```
"AllowedMethods":[
   "GET"
   "PUT"
]
```

Header yang diminta tidak diperbolehkan

Header yang tercantum di `Access-Control-Request-Headers` header dalam permintaan pra-penerbangan harus cocok dengan header dalam `AllowedHeaders` elemen dalam konfigurasi CORS Anda. Untuk daftar header umum yang dapat digunakan dalam permintaan ke Amazon S3, lihat Header Permintaan Umum. Untuk informasi selengkapnya tentang cara memperbarui `AllowedHeaders` elemen, lihat Mengonfigurasi berbagi sumber daya lintas asal (CORS).

Contoh berikut menunjukkan bagian dari konfigurasi CORS yang menyertakan `Authorization` header dalam `AllowedHeaders` elemen. Hanya permintaan untuk `Authorization` header yang akan berhasil.

```
"AllowedHeaders":  [
    "Authorization"
]
```

Jika header (misalnya `Content-MD5` disertakan dalam permintaan CORS tetapi header tidak ada dalam konfigurasi CORS Anda, permintaan tersebut akan menghasilkan kesalahan. `403 Forbidden` Untuk mengizinkan permintaan CORS ini, `Content-MD5` header harus ditambahkan ke konfigurasi CORS Anda. Jika Anda ingin meneruskan keduanya `Authorization` dan `Content-MD5` header dalam permintaan CORS ke bucket Anda, konfirmasikan bahwa kedua header disertakan dalam `AllowedHeaders` elemen dalam konfigurasi CORS Anda.

```
"AllowedHeaders":  [
```

```
    "Authorization"
    "Content-MD5"
]
```

Header tidak ditemukan dalam respons CORS

`ExposeHeaders`Elemen dalam konfigurasi CORS Anda mengidentifikasi header respons mana yang ingin Anda akses ke skrip dan aplikasi yang berjalan di browser, sebagai tanggapan atas permintaan CORS.

Jika objek yang disimpan di bucket S3 memiliki metadata yang ditentukan pengguna (misalnya,`x-amz-meta-custom-header`) beserta data respons, header kustom ini dapat berisi metadata atau informasi tambahan yang ingin diakses dari kode sisi klien. JavaScript Namun, secara default, browser memblokir akses ke header khusus untuk alasan keamanan. Untuk memungkinkan sisi klien Anda JavaScript mengakses header khusus, Anda harus menyertakan header dalam konfigurasi CORS Anda.

Pada contoh di bawah ini, `x-amz-meta-custom-header1` header disertakan dalam `ExposeHeaders` elemen. `x-amz-meta-custom-header2`Tidak termasuk dalam `ExposeHeaders` elemen dan hilang dari konfigurasi CORS. Sebagai tanggapan, hanya nilai yang termasuk dalam `ExposeHeaders` elemen yang akan dikembalikan. Jika permintaan menyertakan `x-amz-meta-custom-header2` header di `Access-Control-Expose-Headers` header, respons akan tetap mengembalikan a`200 OK`. Namun, hanya header yang diizinkan, Misalnya `x-amz-meta-custom-header` akan dikembalikan dan ditampilkan dalam respons.

```
"ExposeHeaders":  [
    "x-amz-meta-custom-header1"
]
```

Untuk memastikan semua header muncul dalam respons, tambahkan semua header yang diizinkan ke `ExposeHeaders` elemen dalam konfigurasi CORS Anda seperti yang ditunjukkan di bawah ini.

```
"ExposeHeaders":  [
    "x-amz-meta-custom-header1",
    "x-amz-meta-custom-header2"
]
```

Pertimbangan CORS pada integrasi proxy S3

Jika Anda mengalami kesalahan dan telah memeriksa konfigurasi CORS di bucket S3 Anda, dan permintaan lintas asal dikirim ke proxy seperti AWS CloudFront, coba yang berikut ini:

- Konfigurasikan pengaturan untuk memungkinkan `OPTIONS` metode untuk permintaan HTTP.
- Konfigurasikan proxy untuk meneruskan header berikut:`Origin`,`Access-Control-Request-Headers`, dan`Access-Control-Request-Method`.
- Konfigurasikan pengaturan proxy untuk memasukkan header asal dalam kunci cache-nya. Ini penting karena proxy caching yang tidak menyertakan header asal dalam kunci cache mereka dapat menyajikan respons cache yang tidak menyertakan header CORS yang sesuai untuk asal yang berbeda.

Beberapa proxy menyediakan fitur yang telah ditentukan sebelumnya untuk permintaan CORS. Misalnya, di CloudFront, Anda dapat mengonfigurasi kebijakan yang menyertakan header

yang mengaktifkan permintaan berbagi sumber daya lintas asal (CORS) saat asalnya adalah bucket Amazon S3.

Kebijakan ini memiliki pengaturan berikut:

- Header yang disertakan dalam permintaan asal:

  `Origin`

  `Access-Control-Request-Headers`

  `Access-Control-Request-Method`
- Cookie yang disertakan dalam permintaan asal: Tidak ada
- String kueri yang disertakan dalam permintaan asal: Tidak ada

Untuk informasi selengkapnya, lihat Mengontrol permintaan asal dengan kebijakan dan Menggunakan kebijakan permintaan asal terkelola di Panduan CloudFront Pengembang.

Tutorial situs web statis

Tutorial atau panduan berikut menyajikan prosedur lengkap tentang cara membuat dan mengonfigurasi bucket tujuan umum Amazon S3 untuk hosting situs web statis dan hosting streaming

video sesuai permintaan. Tujuan dari tutorial ini adalah untuk memberikan panduan umum. Tutorial ini ditujukan untuk lingkungan tipe lab, dan mereka menggunakan contoh nama bucket, nama pengguna, dan sebagainya. Produk ini tidak dimaksudkan untuk penggunaan secara langsung di lingkungan produksi tanpa tinjauan dan adaptasi yang cermat untuk memenuhi kebutuhan unik di lingkungan organisasi Anda.

- Hosting video streaming sesuai permintaan dengan Amazon S3, Amazon, dan CloudFront Amazon Route 53 - Anda dapat menggunakan Amazon S3 dengan CloudFront Amazon untuk meng-host video untuk dilihat sesuai permintaan dengan cara yang aman dan terukur. Setelah video Anda dikemas ke dalam format yang tepat, Anda dapat menyimpannya di server atau di bucket tujuan umum S3, dan kemudian mengirimkannya sesuai permintaan pemirsa. CloudFront Dalam tutorial ini, Anda akan belajar cara mengonfigurasi bucket tujuan umum Anda untuk meng-host streaming video sesuai permintaan menggunakan CloudFront untuk pengiriman dan Amazon Route 53 untuk Domain Name System (DNS) dan manajemen domain khusus. CloudFront menyajikan video dari cache-nya, mengambilnya dari bucket tujuan umum Anda hanya jika belum di-cache. Fitur manajemen caching ini mempercepat pengiriman video Anda ke pemirsa secara global dengan latensi rendah, throughput tinggi, dan kecepatan transfer tinggi. Untuk informasi selengkapnya tentang manajemen CloudFront caching, lihat Mengoptimalkan caching dan ketersediaan di Panduan Pengembang Amazon CloudFront .
- Mengkonfigurasi situs web statis — Anda dapat mengonfigurasi bucket tujuan umum agar berfungsi seperti situs web. Tutorial ini memandu Anda melalui langkah-langkah hosting situs web di Amazon S3 termasuk membuat ember, mengaktifkan hosting situs web statis di konsol S3, membuat dokumen indeks, dan membuat dokumen kesalahan. Untuk informasi selengkapnya, lihat Hosting situs web statis menggunakan Amazon S3.
- Mengonfigurasi situs web statis menggunakan domain khusus yang terdaftar di Route 53 — Anda dapat membuat dan mengonfigurasi bucket tujuan umum untuk meng-host situs web statis dan membuat pengalihan di S3 untuk situs web dengan nama domain khusus yang terdaftar di Amazon Route 53. Anda menggunakan Route 53 untuk mendaftarkan domain dan menentukan di mana Anda ingin merutekan lalu lintas internet untuk domain Anda. Tutorial ini menunjukkan cara membuat catatan alias Route 53 yang merutekan lalu lintas untuk domain dan subdomain Anda ke bucket tujuan umum Anda yang berisi file HTML. Untuk informasi selengkapnya, lihat Menggunakan domain Anda untuk situs web statis di bucket Amazon S3 di Panduan Pengembang Amazon Route 53. Setelah Anda menyelesaikan tutorial ini, Anda dapat secara opsional menggunakan CloudFront untuk meningkatkan kinerja situs web Anda. Untuk informasi selengkapnya, lihat Mempercepat situs web Anda dengan Amazon CloudFront.

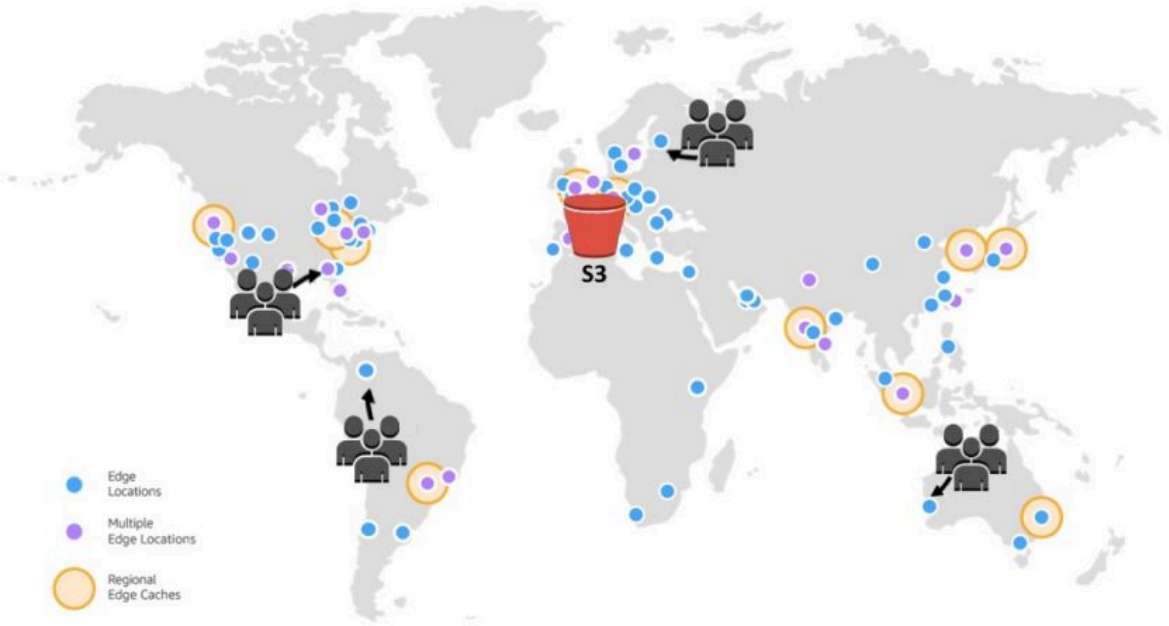
- Menyebarkan situs web statis ke AWS Amplify Hosting dari bucket tujuan umum S3 - Kami menyarankan Anda menggunakan AWS Amplify Hosting untuk meng-host konten situs web statis yang disimpan di S3. Amplify Hosting adalah layanan yang dikelola sepenuhnya yang memudahkan penyebaran situs web Anda di jaringan pengiriman konten (CDN) yang tersedia secara global yang didukung oleh Amazon CloudFront, memungkinkan hosting situs web statis yang aman tanpa pengaturan ekstensif. Dengan AWS Amplify Hosting, Anda dapat memilih lokasi objek dalam bucket tujuan umum, menyebarkan konten Anda ke CDN terkelola, dan membuat URL HTTPS publik agar situs web Anda dapat diakses di mana saja. Untuk informasi selengkapnya, lihat Menerapkan situs web statis dari S3 menggunakan konsol Amplify di Panduan Pengguna AWS Amplify Hosting.

Tutorial: Hosting video streaming sesuai permintaan dengan Amazon S3, Amazon, dan CloudFront Amazon Route 53

Anda dapat menggunakan Amazon S3 dengan Amazon CloudFront untuk meng-host video untuk dilihat sesuai permintaan dengan cara yang aman dan terukur. Streaming video on demand (VOD) berarti konten video Anda disimpan di server dan pemirsa dapat menontonnya kapan saja.

CloudFront adalah layanan jaringan pengiriman konten (CDN) yang cepat, sangat aman, dan dapat diprogram. CloudFront dapat mengirimkan konten Anda dengan aman melalui HTTPS dari semua lokasi CloudFront tepi di seluruh dunia. Untuk informasi selengkapnya CloudFront, lihat Apa itu Amazon CloudFront? di Panduan CloudFront Pengembang Amazon.

CloudFront caching mengurangi jumlah permintaan yang harus ditanggapi oleh server asal Anda secara langsung. Saat pemirsa (pengguna akhir) meminta video yang Anda tayang CloudFront, permintaan tersebut diarahkan ke lokasi tepi terdekat yang lebih dekat ke tempat penampil berada. CloudFront menyajikan video dari cache-nya, mengambilnya dari bucket S3 hanya jika belum di-cache. Fitur manajemen caching ini mempercepat pengiriman video Anda ke pemirsa secara global dengan latensi rendah, throughput tinggi, dan kecepatan transfer tinggi. Untuk informasi selengkapnya tentang manajemen CloudFront caching, lihat Mengoptimalkan caching dan ketersediaan di Panduan Pengembang Amazon CloudFront .



Tujuan

Dalam tutorial ini, Anda mengonfigurasi bucket S3 untuk meng-host streaming video sesuai permintaan menggunakan CloudFront untuk pengiriman dan Amazon Route 53 untuk Sistem Nama Domain (DNS) dan manajemen domain khusus.

Topik

- Prasyarat: Daftarkan dan konfigurasikan domain khusus dengan Route 53
- Langkah 1: Buat ember S3
- Langkah 2: Unggah video ke bucket S3
- Langkah 3: Buat identitas akses CloudFront asal
- Langkah 4: Buat CloudFront distribusi
- Langkah 5: Akses video melalui CloudFront distribusi
- Langkah 6: Konfigurasikan CloudFront distribusi Anda untuk menggunakan nama domain kustom Anda
- Langkah 7: Akses video S3 melalui CloudFront distribusi dengan nama domain khusus
- (Opsional) Langkah 8: Lihat data tentang permintaan yang diterima oleh CloudFront distribusi Anda

- Langkah 9: Membersihkan
- Langkah selanjutnya

Prasyarat: Daftarkan dan konfigurasikan domain khusus dengan Route 53

Sebelum Anda memulai tutorial ini, Anda harus mendaftar dan mengkonfigurasi domain khusus (misalnya,**example.com**) dengan Route 53 sehingga Anda dapat mengonfigurasi CloudFront distribusi Anda untuk menggunakan nama domain khusus nanti.

Tanpa nama domain khusus, video S3 Anda dapat diakses publik dan di-host melalui CloudFront URL yang terlihat mirip dengan berikut ini:

```
https://CloudFront distribution domain name/Path to an S3 video
```

Misalnya, **https://d111111abcdef8.cloudfront.net/sample.mp4**.

Setelah Anda mengonfigurasi CloudFront distribusi Anda untuk menggunakan nama domain khusus yang dikonfigurasi dengan Route 53, video S3 Anda dapat diakses publik dan di-host melalui CloudFront URL yang terlihat mirip dengan berikut ini:

```
https://CloudFront distribution alternate domain name/Path to an S3 video
```

Misalnya, **https://www.example.com/sample.mp4**. Nama domain khusus lebih sederhana dan lebih intuitif untuk digunakan pemirsa Anda.

Untuk mendaftarkan domain khusus, lihat Mendaftarkan domain baru menggunakan Route 53 di Panduan Pengembang Amazon Route 53.

Ketika Anda mendaftarkan nama domain dengan Route 53, Route 53 membuat zona host untuk Anda, yang akan Anda gunakan nanti dalam tutorial ini. Zona yang dihosting ini adalah tempat Anda menyimpan informasi tentang cara merutekan lalu lintas untuk domain Anda, misalnya, ke EC2 instans Amazon atau CloudFront distribusi.

Ada biaya yang terkait dengan pendaftaran domain, zona host Anda, dan kueri DNS yang diterima oleh domain Anda. Untuk informasi lebih lanjut, lihat Harga Amazon Route 53.

> **Note**
>
> Ketika Anda mendaftarkan domain, biayanya segera dan tidak dapat diubah. Anda dapat memilih untuk tidak memperbarui domain secara otomatis, tetapi Anda membayar di muka

dan memilikinya untuk tahun ini. Untuk informasi selengkapnya, lihat Mendaftarkan nama domain di Panduan Pengembang Amazon Route 53.

Langkah 1: Buat ember S3

Buat ember untuk menyimpan video asli yang Anda rencanakan untuk streaming.

Untuk membuat bucket

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bilah navigasi di bagian atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah tempat Anda ingin membuat ember.

   > **Note**
   >
   > Untuk meminimalkan latensi dan biaya serta memenuhi persyaratan regulasi, pilih Wilayah yang dekat dengan Anda. Objek yang disimpan di Wilayah tidak pernah keluar dari Wilayah kecuali Anda secara tegas mentransfer atau mereplikasinya ke Wilayah lain. Untuk daftar Amazon S3 Wilayah AWS, lihat Layanan AWS titik akhir di. Referensi Umum Amazon Web Services

3. Di panel navigasi kiri, pilih Bucket tujuan umum.
4. Pilih Buat bucket. Halaman Buat bucket terbuka.
5. Untuk nama Bucket, masukkan nama untuk bucket Anda (misalnya,**`tutorial-bucket`**).

   Untuk informasi selengkapnya tentang penamaan bucket di Amazon S3, lihat. Aturan penamaan bucket tujuan umum
6. Untuk Wilayah, pilih Wilayah AWS tempat Anda ingin ember berada.

   Jika memungkinkan, Anda harus memilih Wilayah yang paling dekat dengan mayoritas pemirsa Anda. Untuk informasi selengkapnya tentang Wilayah bucket, lihatIkhtisar ember tujuan umum.
7. Untuk pengaturan Blokir Akses Publik untuk bucket ini, pertahankan pengaturan default (Blokir semua akses publik diaktifkan).

Bahkan dengan Blokir semua akses publik diaktifkan, pemirsa masih dapat mengakses video yang diunggah. CloudFront Fitur ini merupakan keuntungan utama menggunakan CloudFront untuk meng-host video yang disimpan di S3.

Kami menyarankan agar Anda tetap mengaktifkan semua pengaturan kecuali Anda perlu mematikan satu atau beberapa pengaturan untuk kasus penggunaan Anda. Untuk informasi lebih lanjut tentang pemblokiran akses publik, lihat Melakukan blok akses publik ke penyimpanan Amazon S3 Anda.

8. Untuk pengaturan yang tersisa, pertahankan default.

   (Opsional) Jika Anda ingin mengonfigurasi pengaturan bucket tambahan untuk kasus penggunaan spesifik Anda, lihatMembuat ember tujuan umum.

9. Pilih Buat bucket.

Langkah 2: Unggah video ke bucket S3

Prosedur berikut menjelaskan cara mengunggah file video ke bucket S3 dengan menggunakan konsol. Jika Anda mengunggah banyak file video besar ke S3, Anda mungkin ingin menggunakan Amazon S3 Transfer Acceleration untuk mengonfigurasi transfer file yang cepat dan aman. Transfer Acceleration dapat mempercepat pengunggahan video ke bucket S3 Anda untuk transfer jarak jauh video yang lebih besar. Untuk informasi selengkapnya, lihat Mengonfigurasi transfer file yang cepat dan aman menggunakan Amazon S3 Transfer Acceleration.

Untuk mengunggah file ke bucket

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Dalam daftar Bucket tujuan umum, pilih nama bucket yang Anda buat di Langkah 1 (misalnya,**`tutorial-bucket`**) untuk mengunggah file Anda.
4. Di tab Objek untuk bucket Anda, pilih Unggah.
5. Pada halaman Unggah, di bawah File dan folder, pilih Tambahkan file.
6. Pilih file yang akan diunggah, lalu pilih Buka.

   Misalnya, Anda dapat mengunggah file video bernama`sample.mp4`.

7. Pilih Unggah.

Langkah 3: Buat identitas akses CloudFront asal

Untuk membatasi akses langsung ke video dari bucket S3 Anda, buat CloudFront pengguna khusus yang disebut identitas akses asal (OAI). Anda akan mengaitkan OAI dengan distribusi Anda nanti dalam tutorial ini. Dengan menggunakan OAI, Anda memastikan bahwa pemirsa tidak dapat mem-bypass CloudFront dan mendapatkan video langsung dari bucket S3. Hanya CloudFront OAI yang dapat mengakses file di bucket S3. Untuk informasi selengkapnya, lihat Membatasi akses ke konten Amazon S3 dengan menggunakan OAI di Panduan Pengembang Amazon CloudFront .

Untuk membuat CloudFront OAI

1. Masuk ke AWS Management Console dan buka CloudFront konsol dihttps://console.aws.amazon.com/cloudfront/v4/home.
2. Di panel navigasi kiri, di bawah bagian Keamanan, pilih Akses asal.
3. Di bawah tab Identitas, pilih Buat identitas akses asal.
4. Masukkan nama (misalnya,**`S3-OAI`**) untuk identitas akses asal baru.
5. Pilih Buat.

Langkah 4: Buat CloudFront distribusi

Untuk digunakan CloudFront untuk melayani dan mendistribusikan video di bucket S3 Anda, Anda harus membuat CloudFront distribusi.

Sublangkah

- Buat CloudFront distribusi
- Tinjau kebijakan bucket

Buat CloudFront distribusi

1. Masuk ke AWS Management Console dan buka CloudFront konsol dihttps://console.aws.amazon.com/cloudfront/v4/home.
2. Di panel navigasi kiri, pilih Distribusi.
3. Pilih Buat Distribusi.
4. Di bagian Origin, untuk domain Origin, pilih nama domain asal S3 Anda, yang dimulai dengan nama bucket S3 yang Anda buat di Langkah 1 (misalnya,**`tutorial-bucket`**).
5. Untuk akses Origin, pilih Identitas akses lama.

6. Di bawah Origin Access Identity, pilih identitas akses asal yang Anda buat di Langkah 3 (misalnya,**`S3-OAI`**).
7. Di bawah kebijakan Bucket, pilih Ya, perbarui kebijakan bucket.
8. Di bagian perilaku cache default, di bawah kebijakan protokol Viewer, pilih Redirect HTTP ke HTTPS.

   Ketika Anda memilih fitur ini, permintaan HTTP secara otomatis dialihkan ke HTTPS untuk mengamankan situs web Anda dan melindungi data pemirsa Anda.
9. Untuk pengaturan lain di bagian perilaku cache default, pertahankan nilai default.

   (Opsional) Anda dapat mengontrol berapa lama file Anda tetap dalam CloudFront cache sebelum CloudFront meneruskan permintaan lain ke asal Anda. Mengurangi durasi memungkinkan Anda untuk melayani konten dinamis. Meningkatkan durasi berarti pemirsa Anda mendapatkan kinerja yang lebih baik karena file Anda lebih mungkin ditayangkan langsung dari cache tepi. Durasi yang lebih lama juga mengurangi beban yang berasal dari Anda. Untuk informasi selengkapnya, lihat Mengelola berapa lama konten tetap berada di cache (kedaluwarsa) di Panduan CloudFront Pengembang Amazon.
10. Untuk bagian lain, pertahankan pengaturan yang tersisa disetel ke default.

    Untuk informasi selengkapnya tentang opsi pengaturan yang berbeda, lihat Nilai yang Anda Tentukan Saat Membuat atau Memperbarui Distribusi di Panduan CloudFront Pengembang Amazon.
11. Di bagian bawah halaman, pilih Buat distribusi.
12. Pada tab Umum untuk CloudFront distribusi Anda, di bawah Detail, nilai kolom terakhir yang dimodifikasi untuk distribusi Anda berubah dari Deploying ke stempel waktu saat distribusi terakhir diubah. Proses ini biasanya memakan waktu beberapa menit.

Tinjau kebijakan bucket

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Dalam daftar Bucket, pilih nama bucket yang Anda gunakan sebelumnya sebagai asal CloudFront distribusi Anda (misalnya,**`tutorial-bucket`**).
4. Pilih tab Izin.

5. Di bagian Kebijakan Bucket, konfirmasikan bahwa Anda melihat pernyataan yang mirip dengan yang berikut ini dalam teks kebijakan bucket:

```
{
    "Version": "2008-10-17",
    "Id": "PolicyForCloudFrontPrivateContent",
    "Statement": [
        {
            "Sid": "1",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access
 Identity EH1HDMB1FH2TC"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::tutorial-bucket/*"
        }
    ]
}
```

   Ini adalah pernyataan bahwa CloudFront distribusi Anda ditambahkan ke kebijakan bucket saat Anda memilih Ya, perbarui kebijakan bucket lebih awal.

   Pembaruan kebijakan bucket ini menunjukkan bahwa Anda berhasil mengonfigurasi CloudFront distribusi untuk membatasi akses ke bucket S3. Karena pembatasan ini, objek dalam ember hanya dapat diakses melalui CloudFront distribusi Anda.

Langkah 5: Akses video melalui CloudFront distribusi

Sekarang, CloudFront dapat menyajikan video yang disimpan di bucket S3 Anda. Untuk mengakses video Anda CloudFront, Anda harus menggabungkan nama domain CloudFront distribusi Anda dengan jalur ke video di bucket S3.

Untuk membuat URL ke video S3 menggunakan nama domain CloudFront distribusi

1. Masuk ke AWS Management Console dan buka CloudFront konsol dihttps://console.aws.amazon.com/cloudfront/v4/home.
2. Di panel navigasi kiri, pilih Distribusi.
3. Untuk mendapatkan nama domain distribusi, lakukan hal berikut:

a. Di kolom Origins, temukan CloudFront distribusi yang benar dengan mencari nama asalnya, yang dimulai dengan bucket S3 yang Anda buat di Langkah 1 (misalnya,**`tutorial-bucket`**).

b. Setelah menemukan distribusi dalam daftar, perluas kolom Nama domain untuk menyalin nilai nama domain untuk CloudFront distribusi Anda.

4. Di tab browser baru, tempel nama domain distribusi yang Anda salin.
5. Kembali ke tab browser sebelumnya, dan buka konsol S3 di https://console.aws.amazon.com/s3/.
6. Di panel navigasi kiri, pilih Bucket.
7. Dalam daftar Bucket, pilih nama bucket yang Anda buat di Langkah 1 (misalnya,**`tutorial-bucket`**).
8. Dalam daftar Objek, pilih nama video yang Anda unggah di Langkah 2 (misalnya,`sample.mp4`).
9. Pada halaman detail objek, di bagian Ikhtisar objek, salin nilai Kunci. Nilai ini adalah jalur ke objek video yang diunggah di bucket S3.
10. Kembali ke tab browser tempat Anda sebelumnya menempelkan nama domain distribusi, masukkan garis miring ke depan (**`/`**) setelah nama domain distribusi, lalu tempel jalur ke video yang Anda salin sebelumnya (misalnya,). `sample.mp4`

    Sekarang, video S3 Anda dapat diakses publik dan di-host melalui CloudFront URL yang terlihat mirip dengan berikut ini:

    ```
    https://CloudFront distribution domain name/Path to the S3 video
    ```

    Ganti *`CloudFront distribution domain name`* dan *`Path to the S3 video`* dengan nilai yang sesuai. Contoh URL adalah**`https://d111111abcdef8.cloudfront.net/sample.mp4`**.

Langkah 6: Konfigurasikan CloudFront distribusi Anda untuk menggunakan nama domain kustom Anda

Untuk menggunakan nama domain Anda sendiri alih-alih nama CloudFront domain di URL untuk mengakses video S3, tambahkan nama domain alternatif ke CloudFront distribusi Anda.

Sublangkah

- Minta sertifikat SSL
- Tambahkan nama domain alternatif ke CloudFront distribusi Anda
- Buat catatan DNS untuk merutekan lalu lintas dari nama domain alternatif Anda ke nama domain CloudFront distribusi Anda
- Periksa apakah IPv6 diaktifkan untuk distribusi Anda dan buat catatan DNS lain jika diperlukan

Minta sertifikat SSL

Untuk memungkinkan pemirsa menggunakan HTTPS dan nama domain khusus Anda di URL untuk streaming video Anda, gunakan AWS Certificate Manager (ACM) untuk meminta sertifikat Secure Sockets Layer (SSL). Sertifikat SSL menetapkan koneksi jaringan terenkripsi ke situs web.

1. Masuk ke AWS Management Console dan buka konsol ACM di https://console.aws.amazon.com/acm/.
2. Jika halaman pengantar muncul, di bawah Sertifikat penyediaan, pilih Memulai.
3. Pada halaman Minta sertifikat, pilih Minta sertifikat publik, lalu pilih Minta sertifikat.
4. Pada halaman Tambahkan nama domain, masukkan nama domain yang memenuhi syarat (FQDN) dari situs yang ingin Anda amankan dengan sertifikat SSL/TLS. Anda dapat menggunakan tanda bintang (*) untuk meminta sertifikat wildcard untuk melindungi beberapa nama situs dalam domain yang sama. Untuk tutorial ini, masukkan ***** dan nama domain khusus yang Anda konfigurasikan di Prasyarat. Misalnya, masukkan***.example.com**, lalu pilih Berikutnya.

   Untuk informasi selengkapnya, lihat Untuk meminta sertifikat publik ACM (konsol) di Panduan AWS Certificate Manager Pengguna.
5. Pada halaman Pilih metode validasi, pilih validasi DNS. Lalu, pilih Selanjutnya.

   Jika Anda dapat mengedit konfigurasi DNS Anda, kami sarankan Anda menggunakan validasi domain DNS daripada validasi email. Validasi DNS memiliki banyak manfaat dibandingkan validasi email. Untuk informasi selengkapnya, lihat Opsi 1: Validasi DNS di AWS Certificate Manager Panduan Pengguna.
6. (Opsional) Pada halaman Tambahkan tag, beri tag sertifikat Anda dengan metadata.
7. Pilih Tinjau.
8. Pada halaman Tinjauan, verifikasi bahwa informasi di bawah nama Domain dan metode Validasi sudah benar. Kemudian, pilih Konfirmasi dan minta.

Halaman Validasi menunjukkan bahwa permintaan Anda sedang diproses dan domain sertifikat sedang divalidasi. Sertifikat menunggu validasi berada dalam status validasi Tertunda.

9. Pada halaman Validasi, pilih panah bawah di sebelah kiri nama domain kustom Anda, lalu pilih Buat catatan di Route 53 untuk memvalidasi kepemilikan domain Anda melalui DNS.

   Melakukan hal ini menambahkan catatan CNAME yang disediakan oleh AWS Certificate Manager ke konfigurasi DNS Anda.

10. Dalam Buat catatan di Route 53 kotak dialog, pilih Buat.

    Halaman Validasi harus menampilkan pemberitahuan status Sukses di bagian bawah.

11. Pilih Lanjutkan untuk melihat halaman daftar Sertifikat.

    Status untuk sertifikat baru Anda berubah dari validasi Tertunda menjadi Diterbitkan dalam waktu 30 menit.

Tambahkan nama domain alternatif ke CloudFront distribusi Anda

1. Masuk ke AWS Management Console dan buka CloudFront konsol dihttps://console.aws.amazon.com/cloudfront/v4/home.
2. Di panel navigasi kiri, pilih Distribusi.
3. Pilih ID untuk distribusi yang Anda buat di Langkah 4.
4. Pada tab Umum, buka bagian Pengaturan, dan pilih Edit.
5. Pada halaman Edit pengaturan, untuk Nama domain alternatif (CNAME) - opsional, pilih Tambahkan item untuk menambahkan nama domain khusus yang ingin Anda gunakan di URL untuk video S3 yang disajikan oleh distribusi ini CloudFront .

   Dalam tutorial ini, misalnya, jika Anda ingin merutekan lalu lintas untuk subdomain, sepertiwww.example.com, masukkan nama subdomain (www) dengan nama domain (example.com). Secara khusus, masukkan**www.example.com**.

   > **Note**
   >
   > Nama domain alternatif (CNAME) yang Anda tambahkan harus dicakup oleh sertifikat SSL yang sebelumnya Anda lampirkan ke distribusi Anda CloudFront.

6. Untuk sertifikat SSL Kustom - opsional, pilih sertifikat SSL yang Anda minta sebelumnya (misalnya,**`*.example.com`**).

   > **Note**
   >
   > Jika Anda tidak melihat sertifikat SSL segera setelah Anda memintanya, tunggu 30 menit, lalu segarkan daftar hingga sertifikat SSL tersedia untuk Anda pilih.

7. Pertahankan pengaturan yang tersisa disetel ke default. Pilih Simpan perubahan.
8. Pada tab Umum untuk distribusi, tunggu nilai Terakhir diubah dari Deploying ke stempel waktu saat distribusi terakhir diubah.

Buat catatan DNS untuk merutekan lalu lintas dari nama domain alternatif Anda ke nama domain CloudFront distribusi Anda

1. Masuk ke AWS Management Console dan buka konsol Route 53 di https://console.aws.amazon.com/route53/.
2. Di panel navigasi kiri, pilih Zona yang dihosting.
3. Pada halaman Zona yang di-host, pilih nama zona host yang dibuat Route 53 untuk Anda di Prasyarat (misalnya,). **`example.com`**
4. Pilih Buat catatan, lalu gunakan metode Quick create record.
5. Untuk nama Rekam, pertahankan nilai untuk nama rekaman sama dengan nama domain alternatif dari CloudFront distribusi yang Anda tambahkan sebelumnya.

   Dalam tutorial ini, untuk merutekan lalu lintas ke subdomain, seperti`www.example.com`, masukkan nama subdomain tanpa nama domain. Misalnya, masukkan hanya **`www`** di bidang teks sebelum nama domain kustom Anda.
6. Untuk jenis Rekam, pilih A - Rute lalu lintas ke IPv4 alamat dan beberapa AWS sumber daya.
7. Untuk Nilai, pilih sakelar Alias untuk mengaktifkan sumber daya alias.
8. Di bawah Rute lalu lintas ke, pilih Alias untuk CloudFront didistribusikan dari daftar dropdown.
9. Di kotak pencarian yang bertuliskan Pilih distribusi, pilih nama domain CloudFront distribusi yang Anda buat di Langkah 4.

   Untuk menemukan nama domain CloudFront distribusi Anda, lakukan hal berikut:

a. Di tab browser baru, masuk ke AWS Management Console dan buka CloudFront konsol dihttps://console.aws.amazon.com/cloudfront/v3/home.
b. Di panel navigasi kiri, pilih Distribusi.
c. Di kolom Origins, temukan CloudFront distribusi yang benar dengan mencari nama asalnya, yang dimulai dengan bucket S3 yang Anda buat di Langkah 1 (misalnya,**`tutorial-bucket`**).
d. Setelah menemukan distribusi dalam daftar, perluas kolom Nama domain untuk melihat nilai nama domain untuk CloudFront distribusi Anda.

10. Pada halaman Buat catatan di konsol Route 53, untuk pengaturan yang tersisa, pertahankan defaultnya.
11. Pilih Create records (Buat catatan).

Periksa apakah IPv6 diaktifkan untuk distribusi Anda dan buat catatan DNS lain jika diperlukan

Jika IPv6 diaktifkan untuk distribusi Anda, Anda harus membuat catatan DNS lain.

1. Untuk memeriksa IPv6 apakah distribusi diaktifkan, lakukan hal berikut:

   a. Masuk ke AWS Management Console dan buka CloudFront konsol dihttps://console.aws.amazon.com/cloudfront/v4/home.
   b. Di panel navigasi kiri, pilih Distribusi.
   c. Pilih ID CloudFront distribusi yang Anda buat di Langkah 4.
   d. Pada tab Umum, di bawah Pengaturan, periksa apakah IPv6diatur ke Diaktifkan.

   Jika IPv6 diaktifkan untuk distribusi Anda, Anda harus membuat catatan DNS lain.
2. Jika IPv6 diaktifkan untuk distribusi Anda, lakukan hal berikut untuk membuat catatan DNS:

   a. Masuk ke AWS Management Console dan buka konsol Route 53 di https://console.aws.amazon.com/route53/.
   b. Di panel navigasi kiri, pilih Zona yang dihosting.
   c. Pada halaman Zona yang di-host, pilih nama zona host yang dibuat Route 53 untuk Anda di Prasyarat (misalnya,). **`example.com`**
   d. Pilih Buat catatan, lalu gunakan metode Quick create record.
   e. Untuk nama Rekam, di bidang teks sebelum nama domain kustom Anda, ketikkan nilai yang sama dengan yang Anda ketik saat membuat catatan IPv4 DNS sebelumnya. Misalnya,

dalam tutorial ini, untuk merutekan lalu lintas untuk subdomain`www.example.com`, masukkan saja**`www`**.

f. Untuk jenis Rekam, pilih AAAA - Rute lalu lintas ke IPv6 alamat dan beberapa AWS sumber daya.

g. Untuk Nilai, pilih sakelar Alias untuk mengaktifkan sumber daya alias.

h. Di bawah Rute lalu lintas ke, pilih Alias untuk CloudFront didistribusikan dari daftar dropdown.

i. Di kotak pencarian yang bertuliskan Pilih distribusi, pilih nama domain CloudFront distribusi yang Anda buat di Langkah 4.

j. Untuk pengaturan yang tersisa, pertahankan default.

k. Pilih Create records (Buat catatan).

Langkah 7: Akses video S3 melalui CloudFront distribusi dengan nama domain khusus

Untuk mengakses video S3 menggunakan URL khusus, Anda harus menggabungkan nama domain alternatif Anda dengan jalur ke video di bucket S3.

Untuk membuat URL khusus untuk mengakses video S3 melalui distribusi CloudFront

1. Masuk ke AWS Management Console dan buka CloudFront konsol dihttps://console.aws.amazon.com/cloudfront/v4/home.
2. Di panel navigasi kiri, pilih Distribusi.
3. Untuk mendapatkan nama domain alternatif dari CloudFront distribusi Anda, lakukan hal berikut:

   a. Di kolom Origins, temukan CloudFront distribusi yang benar dengan mencari nama asalnya, yang dimulai dengan nama bucket S3 untuk bucket yang Anda buat di Langkah 1 (misalnya,**`tutorial-bucket`**).

   b. Setelah menemukan distribusi dalam daftar, perluas kolom Nama domain alternatif untuk menyalin nilai nama domain alternatif CloudFront distribusi Anda.
4. Di tab browser baru, tempel nama domain alternatif CloudFront distribusi.
5. Kembali ke tab browser sebelumnya, dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
6. Temukan jalur ke video S3 Anda, seperti yang dijelaskan pada Langkah 5.
7. Kembali ke tab browser tempat Anda sebelumnya menempelkan nama domain alternatif, masukkan garis miring (**`/`**), lalu tempel jalur ke video S3 Anda (misalnya,). `sample.mp4`

Sekarang, video S3 Anda dapat diakses publik dan di-host melalui CloudFront URL khusus yang terlihat mirip dengan berikut ini:

```
https://CloudFront distribution alternate domain name/Path to the S3 video
```

Ganti *CloudFront distribution alternate domain name* dan *Path to the S3 video* dengan nilai yang sesuai. Contoh URL adalah**https://www.example.com/sample.mp4**.

(Opsional) Langkah 8: Lihat data tentang permintaan yang diterima oleh CloudFront distribusi Anda

Untuk melihat data tentang permintaan yang diterima oleh CloudFront distribusi Anda

1. Masuk ke AWS Management Console dan buka CloudFront konsol dihttps://console.aws.amazon.com/cloudfront/v4/home.
2. Di panel navigasi kiri, di bawah Laporan & analitik, pilih laporan dari konsol, mulai dari statistik Cache, Objek Populer, Perujuk Teratas, Penggunaan, dan Pemirsa.

   Anda dapat memfilter setiap dasbor laporan. Untuk informasi selengkapnya, lihat CloudFront Laporan di Konsol di Panduan CloudFront Pengembang Amazon.
3. Untuk memfilter data, pilih ID CloudFront distribusi yang Anda buat di Langkah 4.

Langkah 9: Membersihkan

Jika Anda meng-host video streaming S3 menggunakan CloudFront dan Route 53 hanya sebagai latihan pembelajaran, hapus AWS sumber daya yang Anda alokasikan sehingga Anda tidak lagi dikenakan biaya.

> **Note**
>
> Ketika Anda mendaftarkan domain, biayanya segera dan tidak dapat diubah. Anda dapat memilih untuk tidak memperbarui domain secara otomatis, tetapi Anda membayar di muka dan memilikinya untuk tahun ini. Untuk informasi selengkapnya, lihat Mendaftarkan nama domain di Panduan Pengembang Amazon Route 53.

Sublangkah

- Hapus CloudFront distribusi
- Hapus catatan DNS
- Hapus zona yang dihosting publik untuk domain kustom Anda
- Hapus nama domain khusus dari Route 53
- Hapus video asli di bucket sumber S3
- Hapus bucket sumber S3

Hapus CloudFront distribusi

1. Masuk ke AWS Management Console dan buka CloudFront konsol dihttps://console.aws.amazon.com/cloudfront/v4/home.
2. Di panel navigasi kiri, pilih Distribusi.
3. Di kolom Origins, temukan CloudFront distribusi yang benar dengan mencari nama asalnya, yang dimulai dengan nama bucket S3 untuk bucket yang Anda buat di Langkah 1 (misalnya,**`tutorial-bucket`**).
4. Untuk menghapus CloudFront distribusi, Anda harus menonaktifkannya terlebih dahulu.
   - Jika nilai kolom Status Diaktifkan dan nilai Terakhir diubah adalah stempel waktu saat distribusi terakhir diubah, lanjutkan untuk menonaktifkan distribusi sebelum menghapusnya.
   - Jika nilai Status Diaktifkan dan nilai Terakhir diubah adalah Deploying, tunggu hingga nilai Status berubah menjadi stempel waktu saat distribusi terakhir diubah. Kemudian lanjutkan untuk menonaktifkan distribusi sebelum menghapusnya.
5. Untuk menonaktifkan CloudFront distribusi, lakukan hal berikut:
   a. Dalam daftar Distribusi, pilih kotak centang di sebelah ID untuk distribusi yang ingin Anda hapus.
   b. Untuk menonaktifkan distribusi, pilih Nonaktifkan, lalu pilih Nonaktifkan untuk mengonfirmasi.

      Jika Anda menonaktifkan distribusi yang memiliki nama domain alternatif yang terkait dengannya, CloudFront berhenti menerima lalu lintas untuk nama domain tersebut (seperti`www.example.com`), meskipun distribusi lain memiliki nama domain alternatif dengan wildcard (`*`) yang cocok dengan domain yang sama (seperti`*.example.com`).

c. Nilai Status segera berubah menjadi Dinonaktifkan. Tunggu hingga nilai Perubahan terakhir diubah dari Deploying ke stempel waktu saat distribusi terakhir diubah.

   Karena CloudFront harus menyebarkan perubahan ini ke semua lokasi tepi, mungkin perlu beberapa menit sebelum pembaruan selesai dan opsi Hapus tersedia bagi Anda untuk menghapus distribusi.

6. Untuk menghapus distribusi yang dinonaktifkan, lakukan hal berikut:

   a. Pilih kotak centang di sebelah ID untuk distribusi yang ingin Anda hapus.

   b. Pilih Hapus, lalu pilih Hapus untuk mengonfirmasi.

Hapus catatan DNS

Jika Anda ingin menghapus zona yang dihosting publik untuk domain (termasuk catatan DNS), lihat Hapus zona yang dihosting publik untuk domain kustom Anda di Panduan Pengembang Amazon Route 53. Jika Anda hanya ingin menghapus catatan DNS yang dibuat pada Langkah 6, lakukan hal berikut:

1. Masuk ke AWS Management Console dan buka konsol Route 53 di https://console.aws.amazon.com/route53/.
2. Di panel navigasi kiri, pilih Zona yang dihosting.
3. Pada halaman Zona yang di-host, pilih nama zona host yang dibuat Route 53 untuk Anda di Prasyarat (misalnya,). **`example.com`**
4. Dalam daftar catatan, pilih kotak centang di samping catatan yang ingin Anda hapus (catatan yang Anda buat di Langkah 6).

   > **Note**
   >
   > Anda tidak dapat menghapus catatan yang memiliki nilai Type NS atau SOA.

5. Pilih Hapus catatan.
6. Untuk mengonfirmasi penghapusan, pilih Hapus.

   Perubahan pada catatan membutuhkan waktu untuk menyebar ke server DNS Route 53. Saat ini, satu-satunya cara untuk memverifikasi bahwa perubahan Anda telah disebarkan adalah dengan menggunakan tindakan GetChange API. Perubahan biasanya menyebar ke semua server nama Route 53 dalam waktu 60 detik.

Hapus zona yang dihosting publik untuk domain kustom Anda

Warning

Jika Anda ingin menyimpan pendaftaran domain tetapi berhenti merutekan lalu lintas internet ke situs web atau aplikasi web Anda, kami sarankan Anda menghapus catatan di zona yang dihosting (seperti yang dijelaskan di bagian sebelumnya) alih-alih menghapus zona yang dihosting.

Jika Anda menghapus zona yang dihosting, orang lain dapat menggunakan domain dan mengarahkan lalu lintas ke sumber daya mereka sendiri menggunakan nama domain Anda. Selain itu, jika Anda menghapus zona yang dihosting, Anda tidak dapat membatalkan menghapusnya. Anda harus membuat zona host baru dan memperbarui server nama untuk pendaftaran domain Anda, yang dapat memakan waktu hingga 48 jam untuk diterapkan.

Jika Anda ingin membuat domain tidak tersedia di internet, pertama-tama Anda dapat mentransfer layanan DNS Anda ke layanan DNS gratis dan kemudian menghapus zona yang dihosting Route 53. Hal ini mencegah kueri DNS agar tidak salah dirutekan di masa mendatang.

1. Jika domain terdaftar di Route 53, lihat Menambahkan atau mengubah server nama dan merekatkan catatan untuk domain di Panduan Pengembang Amazon Route 53 untuk informasi tentang cara mengganti server nama Route 53 dengan server nama untuk layanan DNS baru.
2. Jika domain yang terdaftar dengan registrar lain, gunakan metode yang disediakan oleh registrar untuk mengubah server nama domain.

Note

Jika Anda menghapus zona yang dihosting untuk subdomain (`www.example.com`), Anda tidak perlu mengubah server nama untuk domain ()`example.com`.

1. Masuk ke AWS Management Console dan buka konsol Route 53 di https://console.aws.amazon.com/route53/.
2. Di panel navigasi kiri, pilih Zona yang dihosting.
3. Pada halaman Zona yang dihosting, pilih nama zona yang dihosting yang ingin Anda hapus.

4. Pada tab Rekaman untuk zona yang dihosting, konfirmasikan bahwa zona yang dihosting yang ingin Anda hapus hanya berisi catatan NS dan SOA.

   Jika berisi catatan tambahan, hapus terlebih dahulu.

   Jika Anda membuat catatan NS untuk subdomain di zona yang dihosting, hapus catatan tersebut juga.

5. Pada tab penandatanganan DNSSEC untuk zona yang dihosting, nonaktifkan penandatanganan DNNSSEC jika diaktifkan. Untuk informasi selengkapnya, lihat Menonaktifkan penandatanganan DNSSEC di Panduan Pengembang Amazon Route 53.
6. Di bagian atas halaman detail zona yang dihosting, pilih Hapus zona.
7. Untuk mengonfirmasi penghapusan, masukkan**`delete`**, lalu pilih Hapus.

Hapus nama domain khusus dari Route 53

Untuk sebagian besar domain tingkat atas (TLDs), Anda dapat menghapus pendaftaran jika Anda tidak lagi menginginkannya. Jika Anda menghapus pendaftaran nama domain dari Route 53 sebelum pendaftaran dijadwalkan kedaluwarsa, AWS tidak mengembalikan biaya pendaftaran. Untuk informasi selengkapnya, lihat Menghapus pendaftaran nama domain di Panduan Pengembang Amazon Route 53.

> **Important**
>
> Jika Anda ingin mentransfer domain antara Akun AWS atau mentransfer domain ke registrar lain, jangan hapus domain dan berharap untuk segera mendaftarkannya kembali. Sebagai gantinya, lihat dokumentasi yang berlaku di Panduan Pengembang Amazon Route 53:
>
> - Mentransfer domain ke yang berbeda Akun AWS
> - Mentransfer domain dari Amazon Route 53 ke registrar lain

Hapus video asli di bucket sumber S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.

3. Dalam daftar nama Bucket, pilih nama bucket tempat Anda mengunggah video di Langkah 2 (misalnya,**`tutorial-bucket`**).
4. Pada tab Objek, pilih kotak centang di sebelah nama objek yang ingin Anda hapus (misalnya,`sample.mp4`).
5. Pilih Hapus.
6. Di bawah Hapus objek secara permanen? , masukkan **`permanently delete`** untuk mengonfirmasi bahwa Anda ingin menghapus objek ini.
7. Pilih Hapus objek.

Hapus bucket sumber S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Dalam daftar Bucket, pilih tombol opsi di sebelah nama bucket yang Anda buat di Langkah 1 (misalnya,**`tutorial-bucket`**).
4. Pilih Hapus.
5. Di halaman Hapus bucket, konfirmasikan bahwa Anda ingin menghapus bucket dengan memasukkan nama bucket ke dalam bidang teks, lalu pilih Hapus bucket.

Langkah selanjutnya

Setelah Anda menyelesaikan tutorial ini, Anda dapat menjelajahi lebih lanjut kasus penggunaan terkait berikut:

- Transkode video S3 ke dalam format streaming yang dibutuhkan oleh televisi tertentu atau perangkat yang terhubung sebelum menghosting video ini dengan CloudFront distribusi.

  Untuk menggunakan Operasi Batch Amazon S3, AWS Lambda dan AWS Elemental MediaConvert mentranskode kumpulan video ke berbagai format media keluaran, lihat. Tutorial: Video transcoding batch dengan Operasi Batch S3
- Host objek lain yang disimpan di S3, seperti gambar, audio, grafik gerak, style sheet, HTML, JavaScript, aplikasi React, dan sebagainya, menggunakan CloudFront dan Route 53.

  Misalnya, lihat Tutorial: Mengonfigurasi situs web statis menggunakan domain kustom yang terdaftar di Route 53 danMempercepat situs web Anda dengan Amazon CloudFront.

- Gunakan Amazon S3 Transfer Acceleration untuk mengonfigurasi transfer file yang cepat dan aman. Transfer Acceleration dapat mempercepat pengunggahan video ke bucket S3 Anda untuk transfer jarak jauh video yang lebih besar. Transfer Acceleration meningkatkan kinerja transfer dengan merutekan lalu lintas melalui lokasi tepi yang didistribusikan CloudFront secara global dan melalui AWS jaringan backbone. Ini juga menggunakan optimasi protokol jaringan. Untuk informasi selengkapnya, lihat Mengonfigurasi transfer file yang cepat dan aman menggunakan Amazon S3 Transfer Acceleration.

Tutorial: Mengonfigurasi situs web statis untuk Amazon S3

> **Important**
>
> Amazon S3 sudah menerapkan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3) sebagai tingkat dasar enkripsi bagi setiap bucket di Amazon S3. Mulai 5 Januari 2023, semua unggahan objek baru ke Amazon S3 secara otomatis akan dienkripsi tanpa biaya tambahan dan tidak akan berdampak pada kinerja. Status enkripsi otomatis untuk konfigurasi enkripsi default bucket S3 dan untuk unggahan objek baru tersedia di AWS CloudTrail log, S3 Inventory, S3 Storage Lens, konsol Amazon S3, dan sebagai header respons API Amazon S3 tambahan di dan. AWS Command Line Interface AWS SDKs Untuk informasi selengkapnya, lihat FAQ enkripsi default.

Anda dapat mengonfigurasi bucket Amazon S3 agar berfungsi seperti situs web. Contoh ini memandu Anda melalui langkah-langkah untuk menghosting situs web di Amazon S3.

> **Important**
>
> Tutorial berikut ini mengharuskan Blokir Akses Publik dinonaktifkan. Kami menyarankan agar Anda menyimpan Blokir Akses Publik diaktifkan. Jika Anda ingin mengaktifkan keempat pengaturan Blokir Akses Publik dan meng-host situs web statis, Anda dapat menggunakan Amazon CloudFront Origin Access Control (OAC). Amazon CloudFront menyediakan kemampuan yang diperlukan untuk menyiapkan situs web statis yang aman. Situs web statis Amazon S3 hanya mendukung titik akhir HTTP. Amazon CloudFront menggunakan penyimpanan Amazon S3 yang tahan lama sambil menyediakan header keamanan tambahan, seperti HTTPS. HTTPS menambahkan keamanan dengan mengenkripsi permintaan HTTP normal, dan melindungi dari serangan siber umum. Untuk informasi

selengkapnya, lihat Memulai situs web statis aman di Panduan CloudFront Pengembang Amazon.

Topik

- Langkah 1: Buat bucket
- Langkah 2: Mengaktifkan hosting situs web statis
- Langkah 3: Mengedit pengaturan Blokir Akses Publik
- Langkah 4: Menambahkan kebijakan bucket yang membuat konten bucket Anda tersedia untuk umum
- Langkah 5: Mengonfigurasi dokumen indeks
- Langkah 6: Mengonfigurasi dokumen kesalahan
- Langkah 7: Menguji titik akhir situs web Anda
- Langkah 8: Membersihkan

Langkah 1: Buat bucket

Petunjuk berikut ini memberikan ikhtisar tentang cara membuat bucket Anda untuk menghosting situs web. Untuk detail, step-by-step instruksi tentang membuat ember, lihatMembuat ember tujuan umum.

Untuk membuat bucket

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Pilih Buat bucket.
3. Masukkan Nama bucket (misalnya, **`example.com`**).
4. Pilih Wilayah tempat Anda ingin membuat bucket.

   Pilih Wilayah yang secara geografis dekat dengan Anda, untuk meminimalkan latensi serta biaya, atau untuk memenuhi persyaratan peraturan. Wilayah yang Anda pilih menentukan titik akhir situs web Amazon S3 Anda. Untuk informasi selengkapnya, lihat Titik akhir situs web.
5. Untuk menerima pengaturan default dan membuat bucket, pilih Buat.

Langkah 2: Mengaktifkan hosting situs web statis

Setelah membuat bucket, Anda dapat mengaktifkan hosting situs web statis untuk bucket Anda. Anda dapat membuat bucket baru, atau menggunakan bucket yang sudah ada.

Untuk mengaktifkan hosting situs web statis

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Dalam daftar bucket, pilih nama bucket yang ingin Anda aktifkan hosting situs web statis.
4. Pilih Properti.
5. Di bagian bawah Hosting situs web statis, pilih Edit.
6. Pilih Gunakan bucket ini untuk menghosting situs web.
7. Di bagian bawah Hosting situs web statis, pilih Aktifkan.
8. Di Dokumen indeks, masukkan nama file dokumen indeks, biasanya `index.html`.

   Nama dokumen indeks peka huruf besar/kecil, dan harus sama persis dengan nama file dokumen indeks HTML yang ingin Anda unggah ke bucket S3 Anda. Saat Anda mengonfigurasi bucket untuk hosting situs web, Anda harus menentukan dokumen indeks. Amazon S3 mengembalikan dokumen indeks ini ketika permintaan dibuat ke domain root atau subfolder mana pun. Untuk informasi selengkapnya, lihat Mengonfigurasi dokumen indeks.
9. Agar dapat menyediakan dokumen kesalahan kustom Anda sendiri untuk kesalahan kelas 4XX, di Dokumen kesalahan, masukkan nama file dokumen kesalahan kustom.

   Nama dokumen kesalahan peka huruf besar/kecil, dan harus sama persis dengan nama file dokumen indeks HTML yang ingin Anda unggah ke bucket S3 Anda. Jika Anda tidak menentukan dokumen kesalahan khusus dan terjadi kesalahan, Amazon S3 mengembalikan dokumen kesalahan HTML default. Untuk informasi selengkapnya, lihat Mengonfigurasi dokumen kesalahan khusus.
10. (Opsional) Jika Anda ingin menentukan aturan pengalihan lanjutan, dalam Aturan pengalihan, masukkan JSON untuk menjelaskan aturannya.

    Misalnya, Anda dapat merutekan permintaan secara kondisional dengan nama kunci atau prefiks objek tertentu dalam permintaan tersebut. Untuk informasi selengkapnya, lihat Konfigurasikan aturan pengalihan untuk menggunakan pengalihan bersyarat lanjutan.

11. Pilih Simpan perubahan.

    Amazon S3 memungkinkan hosting situs web statis untuk bucket Anda. Di bagian bawah halaman, di bawah Hosting situs web statis, Anda melihat titik akhir situs web untuk bucket Anda.

12. Di bagian bawah Hosting situs web statis, perhatikan Titik Akhir.

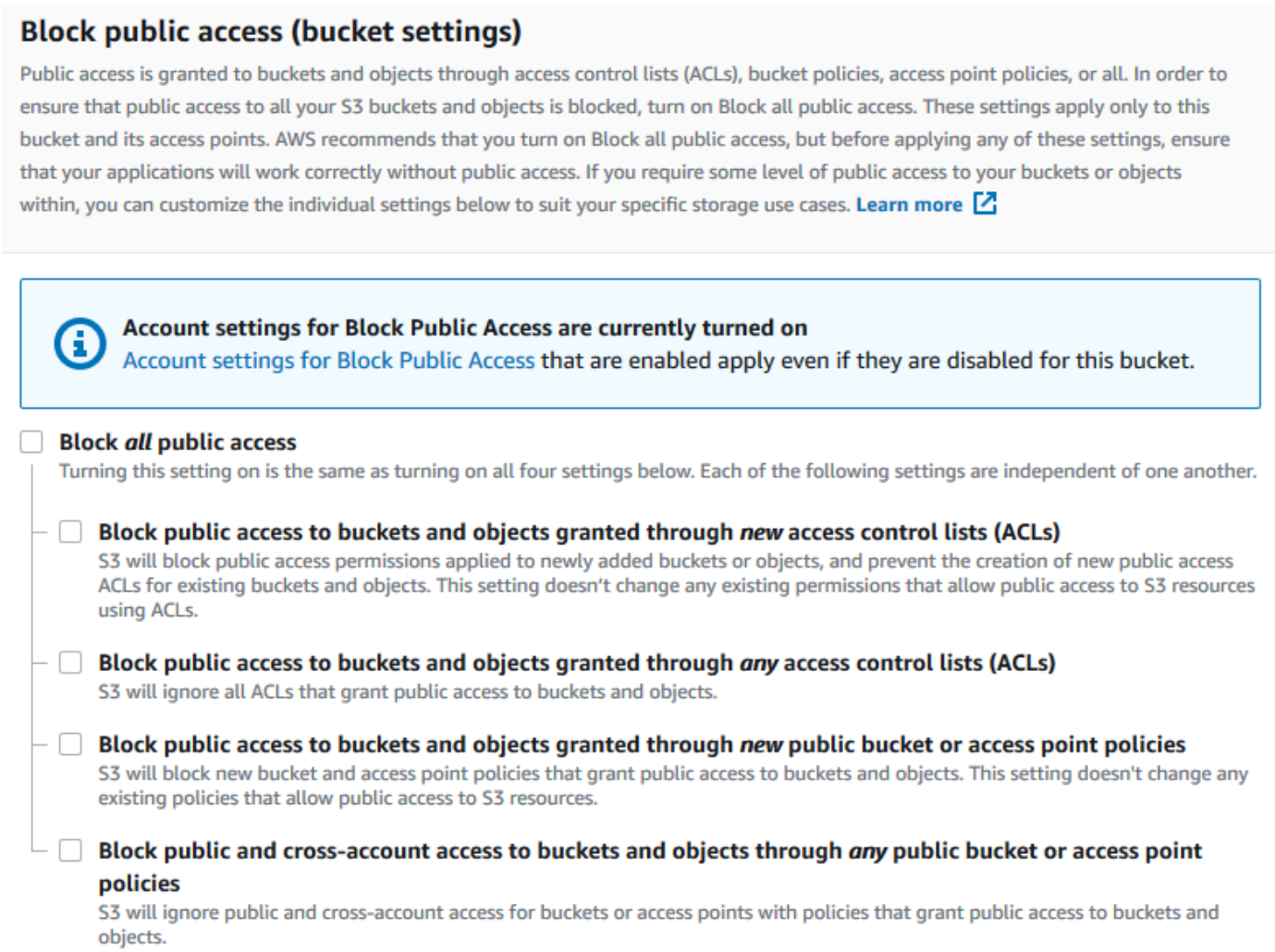
    Titik Akhir adalah titik akhir situs web Amazon S3 untuk bucket Anda. Setelah Anda menyelesaikan konfigurasi bucket Anda sebagai situs web statis, Anda dapat menggunakan titik akhir ini untuk menguji situs web Anda.

Langkah 3: Mengedit pengaturan Blokir Akses Publik

Secara default, Amazon S3 memblokir akses publik ke akun dan bucket Anda. Jika Anda ingin menggunakan bucket untuk menghosting situs web statis, Anda dapat menggunakan langkah-langkah ini untuk mengedit pengaturan blokir akses publik Anda.

> **Warning**
>
> Sebelum Anda menyelesaikan langkah-langkah ini, tinjau Melakukan blok akses publik ke penyimpanan Amazon S3 Anda untuk memastikan bahwa Anda memahami dan menerima risiko yang terkait dengan mengizinkan akses publik. Saat Anda mematikan pengaturan blokir akses publik untuk membuat bucket Anda menjadi publik, siapa pun di internet dapat mengakses bucket Anda. Kami sarankan agar Anda memblokir semua akses publik ke bucket Anda.

1. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Pilih nama bucket yang telah Anda konfigurasikan sebagai situs web statis.
3. Pilih Izin.
4. Di bagian bawah Blokir akses publik (pengaturan bucket), pilih Edit.
5. Kosongkan Blokir semua akses publik, lalu pilih Simpan perubahan.



Amazon S3 mematikan pengaturan Blokir Akses Publik untuk bucket Anda. Untuk membuat situs web statis publik, Anda mungkin juga harus mengedit pengaturan Blokir Akses Publik untuk akun Anda sebelum menambahkan kebijakan bucket. Jika pengaturan Blokir Akses Publik untuk akun Anda saat ini diaktifkan, Anda akan melihat catatan di bawah Blokir akses publik (pengaturan bucket).

Langkah 4: Menambahkan kebijakan bucket yang membuat konten bucket Anda tersedia untuk umum

Setelah Anda mengedit pengaturan Blokir Akses Publik S3, Anda dapat menambahkan kebijakan bucket untuk memberikan akses baca publik ke bucket Anda. Saat Anda memberikan akses baca publik, siapa pun di internet dapat mengakses bucket Anda.

> **Important**
>
> Kebijakan berikut ini hanya merupakan contoh, dan memungkinkan akses penuh ke konten bucket Anda. Sebelum melanjutkan langkah ini, tinjau Bagaimana saya dapat mengamankan file dalam bucket Amazon S3 saya? untuk memastikan bahwa Anda telah memahami praktik terbaik untuk mengamankan file dalam bucket S3, dan risiko yang terlibat dalam pemberian akses publik.

1. Di bagian bawah Bucket, pilih nama bucket Anda.
2. Pilih Izin.
3. Di Bawah Kebijakan bucket, pilih Edit.
4. Untuk memberikan akses baca bagi publik untuk situs web Anda, salin kebijakan kelompok berikut, dan tempelkan di Editor kebijakan bucket.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::Bucket-Name/*"
            ]
        }
    ]
}
```

5. Perbarui `Resource` dengan nama bucket Anda.

   Dalam contoh kebijakan bucket sebelumnya, *Bucket-Name* adalah placeholder untuk nama bucket. Untuk menggunakan kebijakan bucket ini dengan bucket Anda sendiri, Anda harus memperbarui nama ini agar sesuai dengan nama bucket Anda.

6. Pilih Simpan perubahan.

Pesan akan muncul, yang menunjukkan bahwa kebijakan bucket telah berhasil ditambahkan.

Jika Anda melihat kesalahan yang mengatakan `Policy has invalid resource`, konfirmasikan bahwa nama bucket dalam kebijakan bucket tersebut sesuai dengan nama bucket Anda. Untuk informasi tentang menambahkan kebijakan bucket, lihat Bagaimana cara menambahkan kebijakan S3 bucket?

Jika Anda mendapatkan pesan kesalahan dan tidak dapat menyimpan kebijakan bucket, periksa pengaturan akun dan bucket Blokir Akses Publik untuk mengonfirmasi bahwa Anda mengizinkan akses publik ke bucket.

Langkah 5: Mengonfigurasi dokumen indeks

Saat mengaktifkan hosting situs web statis untuk bucket Anda, Anda memasukkan nama dokumen indeks (misalnya, **`index.html`**). Setelah Anda mengaktifkan hosting situs web statis untuk bucket, Anda mengunggah file HTML dengan nama dokumen indeks ke bucket Anda.

Untuk mengonfigurasi dokumen indeks

1. Buat file `index.html`.

   Jika Anda tidak memiliki file `index.html`, Anda dapat menggunakan HTML berikut ini untuk membuatnya:

   ```
   <html xmlns="http://www.w3.org/1999/xhtml" >
   <head>
       <title>My Website Home Page</title>
   </head>
   <body>
     <h1>Welcome to my website</h1>
     <p>Now hosted on Amazon S3!</p>
   </body>
   </html>
   ```

2. Simpan file indeks secara lokal.

   Nama file dokumen indeks harus sama persis dengan nama dokumen indeks yang Anda masukkan ke dalam kotak dialog Hosting situs web statis. Nama dokumen indeks peka huruf besar/kecil. Misalnya, jika Anda memasukkan `index.html` untuk Dokumen indeks dalam

kotak dialog Hosting situs web statis, nama file dokumen indeks Anda juga harus dan bukan `Index.html`.

3. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
4. Di panel navigasi kiri, pilih Bucket tujuan umum.
5. Dalam daftar bucket, pilih nama bucket yang ingin Anda gunakan untuk meng-host situs web statis.
6. Aktifkan hosting situs web statis untuk bucket Anda, lalu masukkan nama persis dokumen indeks Anda (misalnya, `index.html`). Untuk informasi selengkapnya, lihat Mengaktifkan hosting situs web.

   Setelah mengaktifkan hosting situs web statis, lanjutkan ke langkah 6.
7. Untuk mengunggah dokumen indeks ke bucket Anda, lakukan salah satu hal berikut ini:

   - Seret dan jatuhkan file indeks ke dalam daftar bucket konsol.
   - Pilih Unggah, dan ikuti petunjuk untuk memilih dan mengunggah file indeks.

   Untuk step-by-step instruksi, lihatMengunggah Objek.
8. (Opsional) Mengunggah konten situs web lain ke bucket Anda.

Langkah 6: Mengonfigurasi dokumen kesalahan

Saat mengaktifkan hosting situs web statis untuk bucket Anda, Anda memasukkan nama dokumen kesalahan (misalnya, **`404.html`**). Setelah Anda mengaktifkan hosting situs web statis untuk bucket, Anda mengunggah file HTML dengan nama dokumen indeks ke bucket Anda.

Untuk mengonfigurasi dokumen kesalahan

1. Membuat dokumen kesalahan, misalnya `404.html`.
2. Simpan file dokumen kesalahan secara lokal.

   Nama dokumen kesalahan peka huruf besar/kecil, dan harus sama persis dengan nama yang Anda masukkan saat Anda mengaktifkan hosting situs web statis. Misalnya, jika Anda memasukkan `404.html` sebagai nama Dokumen kesalahan di kotak dialog Hosting situs web statis, nama file dokumen kesalahan Anda juga harus bernama `404.html`.

3. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
4. Di panel navigasi kiri, pilih Bucket tujuan umum.
5. Dalam daftar bucket, pilih nama bucket yang ingin Anda gunakan untuk meng-host situs web statis.
6. Aktifkan hosting situs web statis untuk bucket Anda, lalu masukkan nama persis dokumen indeks Anda (misalnya, `404.html`). Untuk informasi selengkapnya, silakan lihat Mengaktifkan hosting situs web dan Mengonfigurasi dokumen kesalahan khusus.

   Setelah mengaktifkan hosting situs web statis, lanjutkan ke langkah 6.
7. Untuk mengunggah dokumen kesalahan ke bucket Anda, lakukan salah satu hal berikut ini:

   - Seret dan jatuhkan file dokumen kesalahan ke dalam daftar bucket konsol.
   - Pilih Unggah, dan ikuti petunjuk untuk memilih dan mengunggah file indeks.

   Untuk step-by-step instruksi, lihatMengunggah Objek.

Langkah 7: Menguji titik akhir situs web Anda

Setelah Anda mengonfigurasi hosting situs web statis untuk bucket, Anda dapat menguji titik akhir situs web Anda.

> **Note**
>
> Amazon S3 tidak mendukung akses HTTPS ke situs web. Jika Anda ingin menggunakan HTTPS, Anda dapat menggunakan Amazon CloudFront untuk melayani situs web statis yang dihosting di Amazon S3.
> Untuk informasi selengkapnya, lihat Bagaimana CloudFront cara menggunakan situs web statis yang dihosting di Amazon S3? dan Memerlukan HTTPS untuk komunikasi antara pemirsa dan CloudFront.

1. Di bagian bawah Bucket, pilih nama bucket Anda.
2. Pilih Properti.
3. Di bagian bawah halaman, di bawah Hosting situs web statis, pilih Titik akhir situs web bucket.

Dokumen indeks Anda terbuka di jendela browser terpisah.

Anda sekarang memiliki situs web yang di-host di Amazon S3. Situs web ini tersedia di titik akhir situs web Amazon S3. Namun, Anda mungkin memiliki domain, seperti `example.com`, yang ingin Anda gunakan untuk menyajikan konten dari situs web yang Anda buat. Anda mungkin juga ingin menggunakan dukungan domain root Amazon S3 untuk melayani permintaan untuk keduanya `http://www.example.com` dan `http://example.com`. Ini memerlukan langkah-langkah tambahan. Sebagai contoh, lihat Tutorial: Mengonfigurasi situs web statis menggunakan domain kustom yang terdaftar di Route 53.

Langkah 8: Membersihkan

Jika Anda membuat situs web statis hanya sebagai latihan pembelajaran, hapus sumber daya AWS yang Anda alokasikan, sehingga Anda tidak lagi dikenakan biaya. Setelah Anda menghapus AWS sumber daya Anda, situs web Anda tidak lagi tersedia. Untuk informasi selengkapnya, lihat Menghapus bucket tujuan umum.

Tutorial: Mengonfigurasi situs web statis menggunakan domain kustom yang terdaftar di Route 53

Misalnya, Anda ingin menghosting situs web statis di Amazon S3. Anda telah mendaftarkan domain dengan Amazon Route 53 (misalnya, `example.com`), dan Anda ingin meminta `http://www.example.com` dan `http://example.com` untuk dilayani dari konten Amazon S3. Anda dapat menggunakan panduan ini untuk mempelajari cara menghosting situs web statis, dan membuat pengalihan ke Amazon S3 untuk situs web dengan nama domain kustom yang terdaftar di Route 53. Anda dapat bekerja dengan situs web yang ada yang ingin Anda hosting di Amazon S3, atau menggunakan panduan ini untuk memulai dari awal.

Setelah Anda menyelesaikan panduan ini, Anda dapat menggunakan Amazon secara opsional CloudFront untuk meningkatkan kinerja situs web Anda. Untuk informasi selengkapnya, lihat Mempercepat situs web Anda dengan Amazon CloudFront.

> **Note**
>
> Titik akhir situs web Amazon S3 tidak mendukung HTTPS atau titik akses. Jika Anda ingin menggunakan HTTPS, Anda dapat menggunakan Amazon CloudFront untuk melayani situs web statis yang dihosting di Amazon S3.

Untuk tutorial tentang cara meng-host konten Anda dengan aman CloudFront dan Amazon S3, lihat. Tutorial: Hosting video streaming sesuai permintaan dengan Amazon S3, Amazon, dan CloudFront Amazon Route 53 Untuk informasi lebih lanjut, lihat Bagaimana cara menggunakan CloudFront untuk melayani situs web statis yang dihosting di Amazon S3? dan Memerlukan HTTPS untuk komunikasi antara penampil dan CloudFront.

Mengotomatiskan pengaturan situs web statis dengan templat AWS CloudFormation

Anda dapat menggunakan AWS CloudFormation template untuk mengotomatiskan pengaturan situs web statis Anda. AWS CloudFormation Template mengatur komponen yang Anda butuhkan untuk meng-host situs web statis yang aman sehingga Anda dapat lebih fokus pada konten situs web Anda dan lebih sedikit pada konfigurasi komponen.

AWS CloudFormation Template mencakup komponen-komponen berikut:

- Amazon S3–Membuat bucket Amazon S3 untuk menghosting situs web statis Anda.
- CloudFront — Membuat CloudFront distribusi untuk mempercepat situs web statis Anda.
- Lambda@Edge–Menggunakan Lambda@Edge untuk menambahkan header keamanan ke setiap respons server. Header keamanan adalah sekelompok header di respons server web yang memberi tahu browser web untuk mengambil tindakan pencegahan keamanan ekstra. Untuk informasi selengkapnya, lihat posting blog Menambahkan header keamanan HTTP menggunakan Lambda @Edge dan Amazon. CloudFront

AWS CloudFormation Template ini tersedia untuk Anda unduh dan gunakan. Untuk informasi dan petunjuk, lihat Memulai situs web statis aman di Panduan CloudFront Pengembang Amazon.

Topik

- Sebelum Anda mulai
- Langkah 1: Mendaftarkan domain kustom dengan Route 53
- Langkah 2: Membuat dua bucket
- Langkah 3: Mengonfigurasi bucket domain root Anda untuk hosting situs web
- Langkah 4: Mengonfigurasi bucket subdomain Anda untuk pengalihan situs web
- Langkah 5: Mengonfigurasi pencatatan untuk lalu lintas situs web
- Langkah 6: Mengunggah konten indeks dan situs web
- Langkah 7: Mengunggah dokumen kesalahan

- Langkah 8: Edit pengaturan Blokir Akses Publik S3
- Langkah 9: Melampirkan kebijakan bucket
- Langkah 10: Menguji titik akhir domain Anda
- Langkah 11: Menambahkan catatan alias untuk domain dan subdomain Anda
- Langkah 12: Menguji situs webnya
- Mempercepat situs web Anda dengan Amazon CloudFront
- Membersihkan sumber daya contoh Anda

Sebelum Anda mulai

Saat Anda mengikuti langkah-langkah dalam contoh ini, Anda bekerja dengan layanan berikut:

Amazon Route 53–Anda menggunakan Route 53 untuk mendaftarkan domain dan menentukan di mana Anda ingin mengirimkan lalu lintas internet untuk domain Anda. Contoh ini menunjukkan cara untuk membuat catatan alias Route 53 yang merutekan lalu lintas untuk domain Anda (`example.com`) dan subdomain (`www.example.com`) ke bucket Amazon S3 yang berisi file HTML.

Amazon S3–Anda menggunakan Amazon S3 untuk membuat bucket, mengunggah halaman situs web sampel, mengonfigurasi izin sehingga semua orang dapat melihat konten, dan kemudian mengonfigurasi bucket untuk hosting situs web.

Langkah 1: Mendaftarkan domain kustom dengan Route 53

Jika Anda belum memiliki nama domain yang terdaftar, seperti `example.com`, daftarkan satu dengan Route 53. Untuk informasi selengkapnya, lihat Mendaftarkan nama domain di Panduan Pengembang Amazon Route 53. Setelah mendaftarkan nama domain Anda, Anda dapat membuat dan mengonfigurasi bucket Amazon S3 Anda untuk hosting situs web.

Langkah 2: Membuat dua bucket

Untuk mendukung permintaan dari domain root dan subdomain, Anda membuat dua bucket.

- Bucket domain–`example.com`
- Bucket subdomain–`www.example.com`

Nama bucket ini harus sama persis dengan nama domain Anda. Dalam contoh ini, nama domain adalah `example.com`. Anda menghosting konten Anda dari bucket domain root (`example.com`).

Anda membuat permintaan pengalihan untuk bucket subdomain (`www.example.com`). Jika seseorang masuk ke `www.example.com` dalam browser mereka, mereka akan diarahkan ke `example.com` dan melihat konten yang di-host di bucket Amazon S3 dengan nama itu.

Untuk membuat bucket Anda untuk hosting situs web

Petunjuk berikut ini memberikan ikhtisar tentang cara membuat bucket Anda untuk menghosting situs web. Untuk detail, step-by-step instruksi tentang membuat ember, lihatMembuat ember tujuan umum.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Buat bucket domain root Anda:

   a. Di bilah navigasi di bagian atas halaman, pilih nama yang saat ini ditampilkan Wilayah AWS. Selanjutnya, pilih Wilayah tempat Anda ingin membuat ember.

   > **Note**
   >
   > Untuk meminimalkan latensi dan biaya serta memenuhi persyaratan regulasi, pilih Wilayah yang dekat dengan Anda. Objek yang disimpan di Wilayah tidak pernah keluar dari Wilayah kecuali Anda secara tegas mentransfer atau mereplikasinya ke Wilayah lain. Untuk daftar Amazon S3 Wilayah AWS, lihat Layanan AWS titik akhir di. Referensi Umum Amazon Web Services

   b. Di panel navigasi kiri, pilih Bucket tujuan umum.

   c. Pilih Buat bucket. Halaman Buat bucket terbuka.

   d. Masukkan Nama bucket (misalnya, **`example.com`**).

   e. Pilih Wilayah tempat Anda ingin membuat bucket.

      Pilih Wilayah yang secara geografis dekat dengan Anda, untuk meminimalkan latensi serta biaya, atau untuk memenuhi persyaratan peraturan. Wilayah yang Anda pilih menentukan titik akhir situs web Amazon S3 Anda. Untuk informasi selengkapnya, lihat Titik akhir situs web.

   f. Untuk menerima pengaturan default dan membuat bucket, pilih Buat.

3. Buat bucket subdomain Anda:

   a. Pilih Buat bucket.

   b. Masukkan Nama bucket (misalnya, **`www.example.com`**).

c. Pilih Wilayah tempat Anda ingin membuat bucket.

   Pilih Wilayah yang secara geografis dekat dengan Anda, untuk meminimalkan latensi serta biaya, atau untuk memenuhi persyaratan peraturan. Wilayah yang Anda pilih menentukan titik akhir situs web Amazon S3 Anda. Untuk informasi selengkapnya, lihat Titik akhir situs web.

d. Untuk menerima pengaturan default dan membuat bucket, pilih Buat.

Pada langkah berikutnya, Anda mengonfigurasi `example.com` untuk hosting situs web.

Langkah 3: Mengonfigurasi bucket domain root Anda untuk hosting situs web

Pada langkah ini, Anda mengonfigurasi bucket domain root (`example.com`) sebagai situs web. Bucket ini akan berisi konten situs web Anda. Saat Anda mengonfigurasikan bucket untuk hosting situs web, Anda dapat mengakses situs web menggunakan Titik akhir situs web.

Untuk mengaktifkan hosting situs web statis

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket tujuan umum.
3. Dalam daftar bucket, pilih nama bucket yang ingin Anda aktifkan hosting situs web statis.
4. Pilih Properti.
5. Di bagian bawah Hosting situs web statis, pilih Edit.
6. Pilih Gunakan bucket ini untuk menghosting situs web.
7. Di bagian bawah Hosting situs web statis, pilih Aktifkan.
8. Di Dokumen indeks, masukkan nama file dokumen indeks, biasanya `index.html`.

   Nama dokumen indeks peka huruf besar/kecil, dan harus sama persis dengan nama file dokumen indeks HTML yang ingin Anda unggah ke bucket S3 Anda. Saat Anda mengonfigurasi bucket untuk hosting situs web, Anda harus menentukan dokumen indeks. Amazon S3 mengembalikan dokumen indeks ini ketika permintaan dibuat ke domain root atau subfolder mana pun. Untuk informasi selengkapnya, lihat Mengonfigurasi dokumen indeks.
9. Agar dapat menyediakan dokumen kesalahan kustom Anda sendiri untuk kesalahan kelas 4XX, di Dokumen kesalahan, masukkan nama file dokumen kesalahan kustom.

Nama dokumen kesalahan peka huruf besar/kecil, dan harus sama persis dengan nama file dokumen indeks HTML yang ingin Anda unggah ke bucket S3 Anda. Jika Anda tidak menentukan dokumen kesalahan khusus dan terjadi kesalahan, Amazon S3 mengembalikan dokumen kesalahan HTML default. Untuk informasi selengkapnya, lihat Mengonfigurasi dokumen kesalahan khusus.

10. (Opsional) Jika Anda ingin menentukan aturan pengalihan lanjutan, dalam Aturan pengalihan, masukkan JSON untuk menjelaskan aturannya.

    Misalnya, Anda dapat merutekan permintaan secara kondisional dengan nama kunci atau prefiks objek tertentu dalam permintaan tersebut. Untuk informasi selengkapnya, lihat Konfigurasikan aturan pengalihan untuk menggunakan pengalihan bersyarat lanjutan.

11. Pilih Simpan perubahan.

    Amazon S3 memungkinkan hosting situs web statis untuk bucket Anda. Di bagian bawah halaman, di bawah Hosting situs web statis, Anda melihat titik akhir situs web untuk bucket Anda.

12. Di bagian bawah Hosting situs web statis, perhatikan Titik Akhir.

    Titik Akhir adalah titik akhir situs web Amazon S3 untuk bucket Anda. Setelah Anda menyelesaikan konfigurasi bucket Anda sebagai situs web statis, Anda dapat menggunakan titik akhir ini untuk menguji situs web Anda.

Setelah Anda mengedit pengaturan blokir akses publik dan menambahkan kebijakan bucket yang memungkinkan akses baca publik, Anda dapat menggunakan titik akhir situs web untuk mengakses situs web Anda.

Pada langkah berikutnya, Anda mengonfigurasi subdomain (`www.example.com`) untuk mengalihkan permintaan ke domain Anda (`example.com`).

Langkah 4: Mengonfigurasi bucket subdomain Anda untuk pengalihan situs web

Setelah Anda mengonfigurasi bucket domain root untuk hosting situs web, Anda dapat mengonfigurasikan bucket subdomain untuk mengalihkan semua permintaan ke domain. Dalam contoh ini, semua permintaan untuk `www.example.com` diarahkan kembali ke `example.com`.

Untuk mengonfigurasi permintaan pengalihan

1. Di konsol Amazon S3, dalam daftar bucket tujuan umum, pilih nama bucket subdomain Anda (`www.example.com`dalam contoh ini).

2. Pilih Properti.
3. Di bagian bawah Hosting situs web statis, pilih Edit.
4. Pilih Alihkan permintaan objek.
5. Di Bucket target, masukkan domain root Anda, misalnya, **`example.com`**.
6. Untuk Protokol, pilih http.
7. Pilih Simpan perubahan.

Langkah 5: Mengonfigurasi pencatatan untuk lalu lintas situs web

Jika ingin melacak jumlah pengunjung yang mengakses situs web Anda, Anda dapat mengaktifkan log untuk bucket domain root Anda secara opsional. Untuk informasi selengkapnya, lihat Pencatatan permintaan dengan pencatatan akses server. Jika Anda berencana menggunakan Amazon CloudFront untuk mempercepat situs web Anda, Anda juga dapat menggunakan CloudFront logging.

Untuk mengaktifkan log akses server untuk bucket domain root Anda

1. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Dalam Wilayah yang sama tempat Anda membuat bucket yang dikonfigurasi sebagai situs web statis, buat bucket untuk log, misalnya `logs.example.com`.
3. Buat folder untuk file log pencatatan akses server (misalnya, `logs`).
4. (Opsional) Jika Anda ingin menggunakan CloudFront untuk meningkatkan kinerja situs web Anda, buat folder untuk file CloudFront log (misalnya,`cdn`).

   > **Important**
   >
   > Saat Anda membuat atau memperbarui distribusi dan mengaktifkan CloudFront pencatatan, CloudFront perbarui daftar kontrol akses bucket (ACL) untuk memberikan `FULL_CONTROL` izin `awslogsdelivery` akun untuk menulis log ke bucket Anda. Untuk informasi selengkapnya, lihat Izin yang diperlukan untuk mengonfigurasi pencatatan standar dan mengakses file log Anda di Panduan CloudFront Pengembang Amazon. Jika bucket yang menyimpan log menggunakan pengaturan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek S3 untuk dinonaktifkan ACLs, CloudFront tidak dapat menulis log ke bucket. Untuk informasi selengkapnya, lihat Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.

5. Di Bucket Anda, pilih bucket domain root Anda.

6. Pilih Properti.
7. Di bagian bawah Pencatatan akses server, pilih Edit.
8. Pilih Aktifkan.
9. Under the Target bucket, choose the bucket and folder destination for the server access:
   - Jelajahi ke lokasi folder dan bucket:
     1. Pilih Jelajahi S3.
     2. Pilih nama bucket, lalu pilih folder log.
     3. Pilih Pilih jalur.
   - Masukkan alur bucket S3, misalnya, `s3://logs.example.com/logs/`.
10. Pilih Simpan perubahan.

    Dalam bucket log, sekarang dapat mengakses log Anda. Amazon S3 menulis log akses situs web ke bucket log Anda setiap 2 jam.

Langkah 6: Mengunggah konten indeks dan situs web

Dalam langkah ini, Anda mengunggah dokumen indeks dan konten situs web opsional ke bucket domain root Anda.

Saat mengaktifkan hosting situs web statis untuk bucket Anda, Anda memasukkan nama dokumen indeks (misalnya, **`index.html`**). Setelah Anda mengaktifkan hosting situs web statis untuk bucket, Anda mengunggah file HTML dengan nama dokumen indeks ke bucket Anda.

Untuk mengonfigurasi dokumen indeks

1. Buat file `index.html`.

   Jika Anda tidak memiliki file `index.html`, Anda dapat menggunakan HTML berikut ini untuk membuatnya:

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
    <title>My Website Home Page</title>
</head>
<body>
  <h1>Welcome to my website</h1>
  <p>Now hosted on Amazon S3!</p>
```

```
</body>
</html>
```

2. Simpan file indeks secara lokal.

   Nama file dokumen indeks harus sama persis dengan nama dokumen indeks yang Anda masukkan ke dalam kotak dialog Hosting situs web statis. Nama dokumen indeks peka huruf besar/kecil. Misalnya, jika Anda memasukkan `index.html` untuk Dokumen indeks dalam kotak dialog Hosting situs web statis, nama file dokumen indeks Anda juga harus dan bukan `Index.html`.
3. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
4. Di panel navigasi kiri, pilih Bucket tujuan umum.
5. Dalam daftar bucket, pilih nama bucket yang ingin Anda gunakan untuk meng-host situs web statis.
6. Aktifkan hosting situs web statis untuk bucket Anda, lalu masukkan nama persis dokumen indeks Anda (misalnya, `index.html`). Untuk informasi selengkapnya, lihat Mengaktifkan hosting situs web.

   Setelah mengaktifkan hosting situs web statis, lanjutkan ke langkah 6.
7. Untuk mengunggah dokumen indeks ke bucket Anda, lakukan salah satu hal berikut ini:

   - Seret dan jatuhkan file indeks ke dalam daftar bucket konsol.
   - Pilih Unggah, dan ikuti petunjuk untuk memilih dan mengunggah file indeks.

   Untuk step-by-step instruksi, lihatMengunggah Objek.
8. (Opsional) Mengunggah konten situs web lain ke bucket Anda.

Langkah 7: Mengunggah dokumen kesalahan

Saat mengaktifkan hosting situs web statis untuk bucket Anda, Anda memasukkan nama dokumen kesalahan (misalnya, **`404.html`**). Setelah Anda mengaktifkan hosting situs web statis untuk bucket, Anda mengunggah file HTML dengan nama dokumen indeks ke bucket Anda.

Untuk mengonfigurasi dokumen kesalahan

1. Membuat dokumen kesalahan, misalnya `404.html`.

2. Simpan file dokumen kesalahan secara lokal.

   Nama dokumen kesalahan peka huruf besar/kecil, dan harus sama persis dengan nama yang Anda masukkan saat Anda mengaktifkan hosting situs web statis. Misalnya, jika Anda memasukkan `404.html` sebagai nama Dokumen kesalahan di kotak dialog Hosting situs web statis, nama file dokumen kesalahan Anda juga harus bernama `404.html`.
3. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
4. Di panel navigasi kiri, pilih Bucket tujuan umum.
5. Dalam daftar bucket, pilih nama bucket yang ingin Anda gunakan untuk meng-host situs web statis.
6. Aktifkan hosting situs web statis untuk bucket Anda, lalu masukkan nama persis dokumen indeks Anda (misalnya, `404.html`). Untuk informasi selengkapnya, silakan lihat Mengaktifkan hosting situs web dan Mengonfigurasi dokumen kesalahan khusus.

   Setelah mengaktifkan hosting situs web statis, lanjutkan ke langkah 6.
7. Untuk mengunggah dokumen kesalahan ke bucket Anda, lakukan salah satu hal berikut ini:

   - Seret dan jatuhkan file dokumen kesalahan ke dalam daftar bucket konsol.
   - Pilih Unggah, dan ikuti petunjuk untuk memilih dan mengunggah file indeks.

   Untuk step-by-step instruksi, lihatMengunggah Objek.

Langkah 8: Edit pengaturan Blokir Akses Publik S3

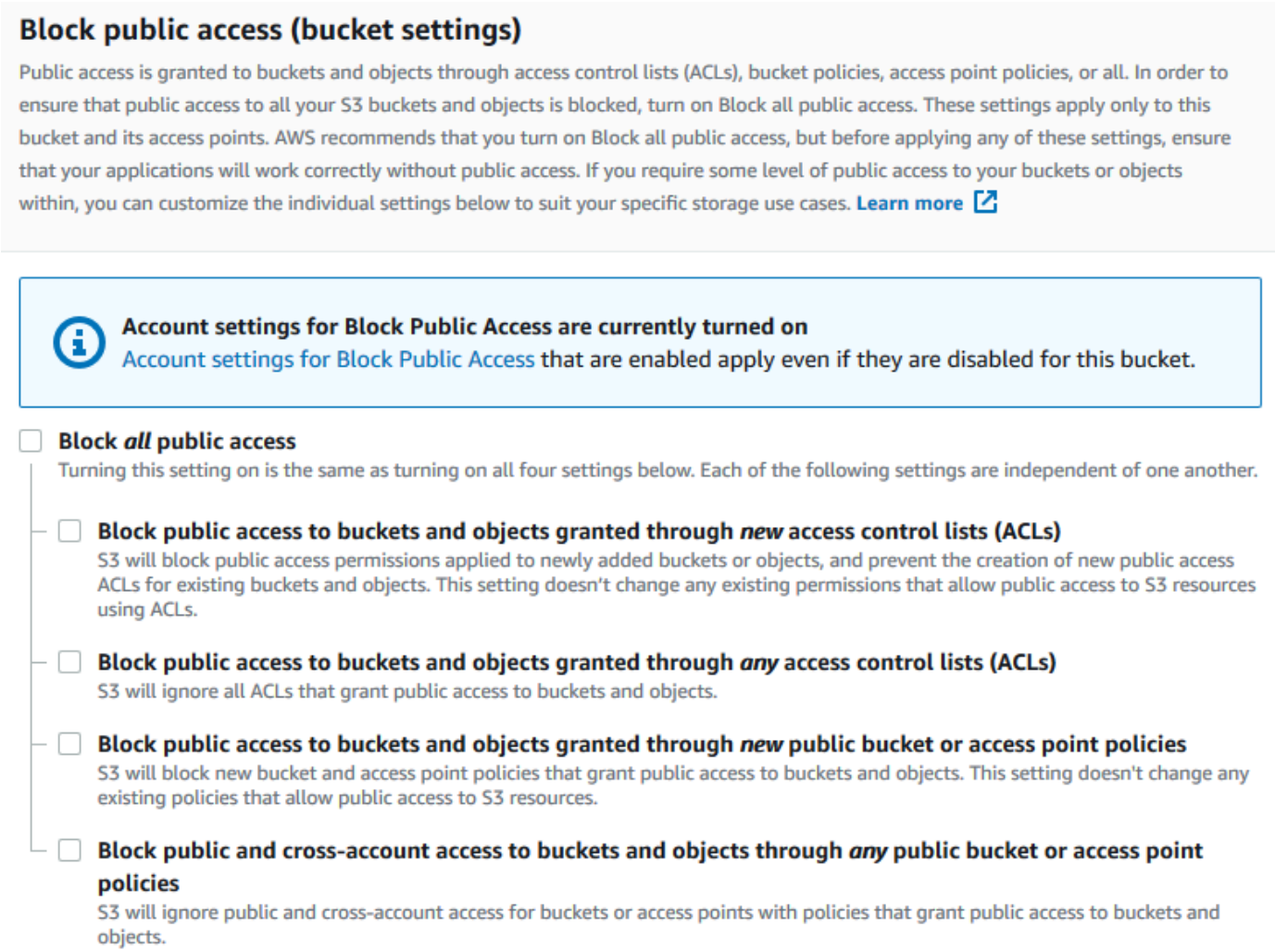
Dalam contoh ini, Anda mengedit pengaturan akses publik untuk bucket domain (`example.com`) untuk memungkinkan akses publik.

Secara default, Amazon S3 memblokir akses publik ke akun dan bucket Anda. Jika Anda ingin menggunakan bucket untuk menghosting situs web statis, Anda dapat menggunakan langkah-langkah ini untuk mengedit pengaturan blokir akses publik Anda.

> **Warning**
>
> Sebelum Anda menyelesaikan langkah-langkah ini, tinjau Melakukan blok akses publik ke penyimpanan Amazon S3 Anda untuk memastikan bahwa Anda memahami dan menerima risiko yang terkait dengan mengizinkan akses publik. Saat Anda mematikan pengaturan

blokir akses publik untuk membuat bucket Anda menjadi publik, siapa pun di internet dapat mengakses bucket Anda. Kami sarankan agar Anda memblokir semua akses publik ke bucket Anda.

1. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Pilih nama bucket yang telah Anda konfigurasikan sebagai situs web statis.
3. Pilih Izin.
4. Di bagian bawah Blokir akses publik (pengaturan bucket), pilih Edit.
5. Kosongkan Blokir semua akses publik, lalu pilih Simpan perubahan.



Amazon S3 mematikan pengaturan Blokir Akses Publik untuk bucket Anda. Untuk membuat situs web statis publik, Anda mungkin juga harus mengedit pengaturan Blokir Akses Publik untuk akun Anda sebelum menambahkan kebijakan bucket. Jika pengaturan Blokir Akses Publik untuk akun

Anda saat ini diaktifkan, Anda akan melihat catatan di bawah Blokir akses publik (pengaturan bucket).

Langkah 9: Melampirkan kebijakan bucket

Dalam contoh ini, Anda melampirkan kebijakan bucket ke bucket domain (`example.com`) untuk memungkinkan akses baca publik. Misalnya, *`Bucket-Name`* Anda mengganti kebijakan bucket contoh dengan nama bucket domain Anda`example.com`.

Setelah Anda mengedit pengaturan Blokir Akses Publik S3, Anda dapat menambahkan kebijakan bucket untuk memberikan akses baca publik ke bucket Anda. Saat Anda memberikan akses baca publik, siapa pun di internet dapat mengakses bucket Anda.

> **Important**
>
> Kebijakan berikut ini hanya merupakan contoh, dan memungkinkan akses penuh ke konten bucket Anda. Sebelum melanjutkan langkah ini, tinjau Bagaimana saya dapat mengamankan file dalam bucket Amazon S3 saya? untuk memastikan bahwa Anda telah memahami praktik terbaik untuk mengamankan file dalam bucket S3, dan risiko yang terlibat dalam pemberian akses publik.

1. Di bagian bawah Bucket, pilih nama bucket Anda.
2. Pilih Izin.
3. Di Bawah Kebijakan bucket, pilih Edit.
4. Untuk memberikan akses baca bagi publik untuk situs web Anda, salin kebijakan kelompok berikut, dan tempelkan di Editor kebijakan bucket.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject"
            ],
```

```
            "Resource": [
                "arn:aws:s3:::Bucket-Name/*"
            ]
        }
    ]
}
```

5. Perbarui `Resource` dengan nama bucket Anda.

   Dalam contoh kebijakan bucket sebelumnya, *`Bucket-Name`* adalah placeholder untuk nama bucket. Untuk menggunakan kebijakan bucket ini dengan bucket Anda sendiri, Anda harus memperbarui nama ini agar sesuai dengan nama bucket Anda.

6. Pilih Simpan perubahan.

   Pesan akan muncul, yang menunjukkan bahwa kebijakan bucket telah berhasil ditambahkan.

   Jika Anda melihat kesalahan yang mengatakan `Policy has invalid resource`, konfirmasikan bahwa nama bucket dalam kebijakan bucket tersebut sesuai dengan nama bucket Anda. Untuk informasi tentang menambahkan kebijakan bucket, lihat Bagaimana cara menambahkan kebijakan S3 bucket?

   Jika Anda mendapatkan pesan kesalahan dan tidak dapat menyimpan kebijakan bucket, periksa pengaturan akun dan bucket Blokir Akses Publik untuk mengonfirmasi bahwa Anda mengizinkan akses publik ke bucket.

Pada langkah berikutnya, Anda dapat mengetahui titik akhir situs web dan menguji titik akhir domain Anda.

Langkah 10: Menguji titik akhir domain Anda

Setelah Anda mengonfigurasikan bucket domain untuk menghosting situs web publik, Anda dapat menguji titik akhir. Untuk informasi selengkapnya, lihat Titik akhir situs web. Anda hanya dapat menguji titik akhir untuk bucket domain Anda, karena bucket subdomain Anda disiapkan untuk pengalihan situs web dan bukan hosting situs web statis.

> **Note**
>
> Amazon S3 tidak mendukung akses HTTPS ke situs web. Jika Anda ingin menggunakan HTTPS, Anda dapat menggunakan Amazon CloudFront untuk melayani situs web statis yang dihosting di Amazon S3.

Untuk informasi selengkapnya, lihat Bagaimana CloudFront cara menggunakan situs web statis yang dihosting di Amazon S3? dan Memerlukan HTTPS untuk komunikasi antara pemirsa dan CloudFront.

1. Di bagian bawah Bucket, pilih nama bucket Anda.
2. Pilih Properti.
3. Di bagian bawah halaman, di bawah Hosting situs web statis, pilih Titik akhir situs web bucket.

   Dokumen indeks Anda terbuka di jendela browser terpisah.

Pada langkah berikutnya, Anda menggunakan Amazon Route 53 untuk memungkinkan pelanggan menggunakan kedua kustom Anda URLs untuk menavigasi ke situs Anda.

Langkah 11: Menambahkan catatan alias untuk domain dan subdomain Anda

Pada langkah ini, Anda membuat catatan alias yang Anda tambahkan ke zona yang di-hosting untuk peta domain `example.com` dan `www.example.com` Anda. Alih-alih menggunakan alamat IP, catatan alias menggunakan titik akhir situs Amazon S3. Amazon Route 53 mempertahankan pemetaan antara catatan alias dan alamat IP tempat bucket Amazon S3 berada. Anda membuat dua rekaman alias, satu untuk domain root Anda, dan satu untuk subdomain Anda.

Menambahkan catatan alias untuk domain root dan subdomain Anda

Untuk menambahkan rekaman alias untuk domain akar Anda (**`example.com`**)

1. Buka konsol Route 53 di https://console.aws.amazon.com/route53/.

   > **Note**
   >
   > Jika Anda belum menggunakan Route 53, lihat Langkah 1: Mendaftarkan domain di Panduan Pengembang Amazon Route 53. Setelah menyelesaikan pengaturan Anda, Anda dapat melanjutkan petunjuknya.

2. Pilih Zona yang di-hosting.
3. Dalam daftar zona yang di-hosting, pilih nama zona yang di-hosting yang sesuai dengan nama domain Anda.
4. Pilih Buat catatan.

5. Pilih Beralih ke wizard.

> **Note**
>
> Jika Anda ingin menggunakan pembuatan cepat untuk membuat catatan alias, lihat Mengonfigurasi Route 53 untuk merutekan lalu lintas ke Bucket S3.

6. Pilih Perutean sederhana, dan pilih Selanjutnya.
7. Pilih Tentukan catatan sederhana.
8. Di Nama catatan, terima nilai defaultnya, yang merupakan nama zona yang di-hosting dan domain Anda.
9. Di Nilai/Rutekan lalu lintas ke, pilih Alias ke titik akhir situs web S3.
10. Pilih Wilayah.
11. Pilih bucket S3.

    Nama bucket harus sesuai dengan nama yang muncul di kotak Nama. Di daftar Pilih bucket S3, nama bucket muncul dengan titik akhir situs web Amazon S3 untuk Wilayah tempat bucket dibuat, misalnya, `s3-website-us-west-1.amazonaws.com (example.com)`.

    Pilih bucket S3 mencantumkan bucket, jika:

    - Anda telah mengonfigurasi bucket sebagai situs web statis.
    - Nama bucket sama dengan nama arsip yang Anda buat.
    - Arus Akun AWS menciptakan ember.

    Jika bucket Anda tidak muncul di daftar Pilih bucket S3, masukkan titik akhir situs web Amazon S3 untuk Wilayah tempat bucket dibuat, misalnya, **`s3-website-us-west-2.amazonaws.com`**. Untuk daftar lengkap titik akhir situs web Amazon S3, lihat Titik akhir Situs Web Amazon S3. Untuk informasi selengkapnya tentang target alias, lihat Nilai/rutekan lalu lintas ke di Panduan Pengembang Amazon Route 53.
12. Dalam jenis Rekam, pilih A - Rute lalu lintas ke IPv4 alamat dan beberapa AWS sumber daya.
13. Untuk Evaluasi kondisi target, pilih Tidak.
14. Pilih Tentukan catatan sederhana.

Untuk menambahkan rekaman alias untuk subdomain Anda (**`www.example.com`**)

1. Di bawah Konfigurasi catatan, pilih Tentukan catatan sederhana.
2. Di Nama catatan untuk subdomain Anda, ketik `www`.
3. Di Nilai/Rutekan lalu lintas ke, pilih Alias ke titik akhir situs web S3.
4. Pilih Wilayah.
5. Pilih bucket S3, misalnya, `s3-website-us-west-2.amazonaws.com (www.example.com)`.

   Jika bucket Anda tidak muncul di daftar Pilih bucket S3, masukkan titik akhir situs web Amazon S3 untuk Wilayah tempat bucket dibuat, misalnya, **`s3-website-us-west-2.amazonaws.com`**. Untuk daftar lengkap titik akhir situs web Amazon S3, lihat Titik akhir Situs Web Amazon S3. Untuk informasi selengkapnya tentang target alias, lihat Nilai/rutekan lalu lintas ke di Panduan Pengembang Amazon Route 53.
6. Dalam jenis Rekam, pilih A - Rute lalu lintas ke IPv4 alamat dan beberapa AWS sumber daya.
7. Untuk Evaluasi kondisi target, pilih Tidak.
8. Pilih Tentukan catatan sederhana.
9. Pada halaman Konfigurasi catatan, pilih Buat catatan.

> **Note**
>
> Perubahan umumnya menyebar ke semua server nama Route 53 dalam waktu 60 detik. Setelah penyebarannya selesai, Anda dapat mengirimkan lalu lintas ke bucket Amazon S3 Anda dengan menggunakan nama rekaman alias yang Anda buat dalam prosedur ini.

Tambahkan rekaman alias untuk domain root dan subdomain Anda (konsol Route 53 yang lama)

Untuk menambahkan rekaman alias untuk domain root Anda (**`example.com`**)

Konsol Route 53 telah dirancang ulang. Di konsol Route 53, Anda dapat menggunakan konsol yang lama untuk sementara waktu. Jika Anda memilih untuk bekerja dengan konsol Route 53 yang lama, gunakan prosedur di bawah ini.

1. Buka konsol Route 53 di https://console.aws.amazon.com/route53/.

> **Note**
>
> Jika Anda belum menggunakan Route 53, lihat Langkah 1: Mendaftarkan domain di Panduan Pengembang Amazon Route 53. Setelah menyelesaikan pengaturan Anda, Anda dapat melanjutkan petunjuknya.

2. Pilih Zona yang Di-Hosting.
3. Dalam daftar zona yang di-hosting, pilih nama zona yang di-hosting yang sesuai dengan nama domain Anda.
4. Pilih Buat Set Catatan.
5. Tentukan nilai-nilai berikut ini:

   Nama

   Terima nilai default, yang merupakan nama zona yang di-hosting dan domain Anda.

   Untuk domain root, Anda tidak perlu memasukkan informasi tambahan apa pun di bidang Nama.

   Tipe

   Pilih A — IPv4 alamat.

   Alias

   Pilih Ya.

   Target Alias

   Di bagian Titik akhir situs web S3 dari daftar tersebut, pilih nama bucket Anda.

   Nama bucket harus sesuai dengan nama yang muncul di kotak Nama. Dalam daftar Target Alias, nama bucket diikuti oleh titik akhir situs web Amazon S3 untuk Wilayah tempat bucket dibuat, misalnya `example.com (s3-website-us-west-2.amazonaws.com)`. Target Alias membuat daftar bucket jika:

   - Anda telah mengonfigurasi bucket sebagai situs web statis.
   - Nama bucket sama dengan nama arsip yang Anda buat.
   - Arus Akun AWS menciptakan ember.

Jika bucket Anda tidak muncul di dalam daftar Target Alias, masukkan titik akhir situs web Amazon S3 untuk Wilayah tempat bucket dibuat, misalnya, `s3-website-us-west-2`. Untuk daftar lengkap titik akhir situs web Amazon S3, lihat Titik akhir Situs Web Amazon S3. Untuk informasi selengkapnya tentang target alias, lihat Nilai/rutekan lalu lintas ke di Panduan Pengembang Amazon Route 53.

Kebijakan Perutean

Terima nilai default Sederhana.

Mengevaluasi Kondisi Target

Terima nilai default Tidak.

6. Pilih Buat.

Untuk menambahkan rekaman alias untuk subdomain Anda (**`www.example.com`**)

1. Di zona yang di-hosting untuk domain root Anda (`example.com`), pilih Buat Set Catatan.
2. Tentukan nilai-nilai berikut ini:

   Nama

   Untuk subdomain, masukkan `www` di dalam kotak.

   Tipe

   Pilih A — IPv4 alamat.

   Alias

   Pilih Ya.

   Target Alias

   Di Titik akhir situs web S3 bagian dari daftar, pilih nama bucket yang sama dengan yang muncul di Nama bidang—misalnya, `www.example.com (s3-website-us-west-2.amazonaws.com)`.

   Kebijakan Perutean

   Terima nilai default Sederhana.

Mengevaluasi Kondisi Target

Terima nilai default Tidak.

3. Pilih Buat.

> **Note**
>
> Perubahan umumnya menyebar ke semua server nama Route 53 dalam waktu 60 detik. Setelah penyebarannya selesai, Anda dapat mengirimkan lalu lintas ke bucket Amazon S3 Anda dengan menggunakan nama rekaman alias yang Anda buat dalam prosedur ini.

Langkah 12: Menguji situs webnya

Verifikasi bahwa situs web dan pengalihannya bekerja dengan benar. Di browser Anda, masukkan file Anda URLs. Dalam contoh ini, Anda dapat mencoba yang berikut URLs:

- Domain (`http://example.com`)–Menampilkan dokumen indeks di bucket `example.com`.
- Subdomain (`http://www.example.com`)–Mengalihkan permintaan Anda ke `http://example.com`. Anda melihat dokumen indeks di bucket `example.com`.

Jika situs web Anda atau tautan pengalihan tidak berfungsi, Anda dapat mencoba hal berikut ini:

- Bersihkan cache–Menghapus cache browser web Anda.
- Periksa server nama–Jika halaman web dan tautan pengalihan tidak berfungsi setelah Anda mengosongkan cache, Anda dapat membandingkan server nama untuk domain Anda dan server nama untuk zona yang di-hosting. Jika server nama tidak cocok, Anda mungkin perlu memperbarui server nama domain Anda agar cocok dengan yang terdaftar di bawah zona yang di-hosting. Untuk informasi lebih lanjut, lihat Menambahkan atau mengubah server nama dan menyimpan glue record untuk domain.

Setelah Anda berhasil menguji domain root dan subdomain Anda, Anda dapat mengatur CloudFront distribusi Amazon untuk meningkatkan kinerja situs web Anda dan menyediakan log yang dapat Anda gunakan untuk meninjau lalu lintas situs web. Untuk informasi selengkapnya, lihat Mempercepat situs web Anda dengan Amazon CloudFront.

Mempercepat situs web Anda dengan Amazon CloudFront

Anda dapat menggunakan Amazon CloudFront untuk meningkatkan kinerja situs web Amazon S3 Anda. CloudFront membuat file situs web Anda (seperti HTML, gambar, dan video) tersedia dari pusat data di seluruh dunia (dikenal sebagai lokasi tepi). Saat pengunjung meminta file dari situs web Anda, CloudFront secara otomatis mengalihkan permintaan ke salinan file di lokasi tepi terdekat. Ini menghasilkan waktu unduh yang lebih cepat daripada jika pengunjung meminta konten dari pusat data yang terletak lebih jauh.

CloudFront menyimpan konten di lokasi tepi untuk jangka waktu yang Anda tentukan. Jika pengunjung meminta konten yang telah di-cache lebih lama dari tanggal kedaluwarsa, CloudFront periksa server asal untuk melihat apakah versi konten yang lebih baru tersedia. Jika versi yang lebih baru tersedia, CloudFront salin versi baru ke lokasi tepi. Perubahan yang Anda lakukan pada konten asli direplikasi ke lokasi edge pada saat pengunjung meminta konten tersebut.

Menggunakan CloudFront tanpa Route 53

Tutorial di halaman ini menggunakan Route 53 untuk menunjuk ke CloudFront distribusi Anda. Namun, jika Anda ingin menayangkan konten yang dihosting di bucket Amazon S3 menggunakan CloudFront tanpa menggunakan Route 53, lihat CloudFrontTutorial Amazon: Menyiapkan Distribusi Konten Dinamis untuk Amazon S3. Saat menayangkan konten yang dihosting di bucket Amazon S3 menggunakan CloudFront, Anda dapat menggunakan nama bucket apa pun, dan HTTP dan HTTPS didukung.

Mengotomatisasi pengaturan dengan template AWS CloudFormation

Untuk informasi selengkapnya tentang menggunakan AWS CloudFormation templat untuk mengonfigurasi situs web statis aman yang membuat CloudFront distribusi untuk melayani situs web Anda, lihat Memulai situs web statis aman di Panduan CloudFront Pengembang Amazon.

Topik

- Langkah 1: Buat CloudFront distribusi
- Langkah 2: Memperbarui kumpulan catatan untuk domain dan subdomain Anda
- (Opsional) Langkah 3: Memeriksa file log

Langkah 1: Buat CloudFront distribusi

Pertama, Anda membuat CloudFront distribusi. Ini membuat situs web Anda tersedia dari pusat data di seluruh dunia.

Untuk membuat distribusi dengan asal Amazon S3

1. Buka CloudFront konsol dihttps://console.aws.amazon.com/cloudfront/v4/home.
2. Pilih Buat Distribusi.
3. Pada halaman Buat Distribusi, di bagian Pengaturan Asal, untuk Nama Domain Asal, masukkan titik akhir situs web Amazon S3 untuk bucket Anda—misalnya, **`example.com.s3-website.us-west-1.amazonaws.com`**.

   CloudFront mengisi ID Asal untuk Anda.
4. Untuk Pengaturan Perilaku Cache Default, pertahankan nilai yang diatur ke default.

   Dengan pengaturan default untuk Kebijakan Protokol Penampil, Anda dapat menggunakan HTTPS untuk situs web statis Anda. Untuk informasi selengkapnya opsi konfigurasi ini, lihat Nilai yang Anda Tentukan Saat Membuat atau Memperbarui Distribusi Web di Panduan CloudFront Pengembang Amazon.
5. Untuk Pengaturan Distribusi, lakukan hal berikut ini:

   a. Biarkan Kelas Harga agar diatur ke Gunakan Semua Lokasi Edge (Performa Terbaik).

   b. Setel Nama Domain Alternatif (CNAMEs) ke domain root dan `www` subdomain. Dalam tutorial ini, berikut merupakan `example.com` dan `www.example.com`.

   > **Important**
   >
   > Sebelum Anda melakukan langkah ini, perhatikan persyaratan untuk menggunakan nama domain alternatif, terutama terkait kebutuhan akan sertifikat SSL/TLS yang valid.

   c. Untuk Sertifikat SSL, pilih Sertifikat SSL Khusus (example.com), dan pilih sertifikat kustom yang mencakup domain dan nama subdomain.

   Untuk informasi selengkapnya, lihat Sertifikat SSL di Panduan CloudFront Pengembang Amazon.

   d. Di Objek Root Default, masukkan nama dokumen indeks Anda, misalnya, `index.html`.

   Jika URL yang digunakan untuk mengakses distribusi tidak berisi nama file, CloudFront distribusi mengembalikan dokumen indeks. Objek Root Default harus sama persis dengan nama dokumen indeks untuk situs web statis Anda. Untuk informasi selengkapnya, lihat Mengonfigurasi dokumen indeks.

e. Tetapkan Pencatatan untuk Nyala.

> **Important**
>
> Saat Anda membuat atau memperbarui distribusi dan mengaktifkan CloudFront pencatatan, CloudFront perbarui daftar kontrol akses bucket (ACL) untuk memberikan `FULL_CONTROL` izin `awslogsdelivery` akun untuk menulis log ke bucket Anda. Untuk informasi selengkapnya, lihat Izin yang diperlukan untuk mengonfigurasi pencatatan standar dan mengakses file log Anda di Panduan CloudFront Pengembang Amazon. Jika bucket yang menyimpan log menggunakan pengaturan yang diberlakukan pemilik Bucket untuk Kepemilikan Objek S3 untuk dinonaktifkan ACLs, CloudFront tidak dapat menulis log ke bucket. Untuk informasi selengkapnya, lihat Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.

f. Untuk Bucket untuk Log, pilih bucket log yang Anda buat.

   Untuk informasi lebih lanjut tentang konfigurasi bucket log, lihat (Opsional) Mencatat lalu lintas web.

g. Jika Anda ingin menyimpan log yang dihasilkan oleh lalu lintas ke CloudFront distribusi dalam folder, di Awalan Log, masukkan nama folder.

h. Simpan semua pengaturan lain pada nilai defaultnya.

6. Pilih Buat Distribusi.
7. Untuk melihat status distribusi, cari distribusi di konsol dan periksa Status kolom.

   Status dari `InProgress` menunjukkan bahwa distribusi belum diterapkan sepenuhnya.

   Setelah distribusi dikerahkan, Anda dapat merujuk konten dengan CloudFront nama domain.
8. Catat nilai Nama Domain yang ditampilkan di CloudFront konsol, misalnya,`dj4p1rv6mvubz.cloudfront.net`.
9. Untuk memverifikasi bahwa CloudFront distribusi Anda berfungsi, masukkan nama domain distribusi di browser web.

   Jika situs web Anda terlihat, CloudFront distribusi berfungsi. Jika situs web Anda memiliki domain khusus yang terdaftar di Amazon Route 53, Anda akan memerlukan nama CloudFront domain untuk memperbarui catatan yang ditetapkan pada langkah berikutnya.

Langkah 2: Memperbarui kumpulan catatan untuk domain dan subdomain Anda

Sekarang setelah Anda berhasil membuat CloudFront distribusi, perbarui catatan alias di Route 53 untuk menunjuk ke CloudFront distribusi baru.

Untuk memperbarui catatan alias untuk menunjuk ke distribusi CloudFront

1. Buka konsol Route 53 di https://console.aws.amazon.com/route53/.
2. Pada navigasi di sebelah kiri, pilih Zona yang di-hosting.
3. Pada halaman Zona yang Di-hosting, pilih zona yang di-hosting yang Anda buat untuk subdomain Anda, misalnya, `www.example.com`.
4. Di bagian bawah Catatan, pilih catatan A yang Anda buat untuk subdomain Anda.
5. Di bagian bawah Detail catatan, pilih Edit catatan.
6. Di bawah Rute lalu lintas ke, pilih Alias untuk CloudFront didistribusikan.
7. Di bawah Pilih distribusi, pilih CloudFront distribusi.
8. Pilih Simpan.
9. Untuk mengarahkan catatan A untuk domain root ke CloudFront distribusi, ulangi prosedur ini untuk domain root, misalnya,`example.com`.

   Pembaruan ke set catatan berlaku dalam 2–48 jam.
10. Untuk melihat apakah catatan A yang baru telah diterapkan, di browser web, masukkan URL subdomain Anda, misalnya, `http://www.example.com`.

    Jika browser tidak lagi mengarahkan Anda ke domain root (misalnya, `http://example.com`), catatan A yang baru tersedia. Ketika catatan A baru telah diterapkan, lalu lintas yang diarahkan oleh catatan A baru ke CloudFront distribusi tidak dialihkan ke domain root. Setiap pengunjung yang merujuk situs dengan menggunakan `http://example.com` atau `http://www.example.com` diarahkan ke lokasi CloudFront tepi terdekat, di mana mereka mendapat manfaat dari waktu pengunduhan yang lebih cepat.

> **Tip**
>
> Browser dapat menyimpan pengaturan pengalihan. Jika menurut Anda pengaturan catatan A yang baru seharusnya sudah diterapkan, tetapi browser Anda masih mengalihkan `http://www.example.com` ke `http://example.com`, coba bersihkan

riwayat dan cache browser Anda, tutup dan buka kembali aplikasi browser Anda, atau gunakan browser web yang berbeda.

(Opsional) Langkah 3: Memeriksa file log

Log akses memberi tahu Anda berapa banyak orang yang mengunjungi situs web. Cookie ini juga berisi data bisnis berharga yang dapat Anda analisis dengan layanan lain, seperti Amazon EMR.

CloudFront log disimpan di bucket dan folder yang Anda pilih saat Anda membuat CloudFront distribusi dan mengaktifkan logging. CloudFront menulis log ke bucket log Anda dalam waktu 24 jam sejak permintaan terkait dibuat.

Untuk melihat file log situs web Anda

1. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Pilih nama bucket pencatatan log untuk situs web Anda.
3. Pilih folder CloudFront log.
4. Unduh `.gzip` file yang ditulis oleh CloudFront sebelum membukanya.

Jika Anda membuat situs web hanya sebagai latihan pembelajaran, Anda dapat menghapus sumber daya yang Anda alokasikan, sehingga Anda tidak lagi dikenakan biaya. Untuk melakukannya, lihat Membersihkan sumber daya contoh Anda. Setelah Anda menghapus sumber daya AWS , situs web Anda tidak lagi tersedia.

Membersihkan sumber daya contoh Anda

Jika Anda membuat situs web statis hanya sebagai latihan pembelajaran, hapus sumber daya AWS yang Anda alokasikan, sehingga Anda tidak lagi dikenakan biaya. Setelah Anda menghapus sumber daya AWS , situs web Anda tidak lagi tersedia.

Tugas

- Langkah 1: Hapus CloudFront distribusi Amazon
- Langkah 2: Menghapus Route 53 zona yang di-hosting
- Langkah 3: Menonaktifkan pencatatan dan hapus bucket S3 Anda

Langkah 1: Hapus CloudFront distribusi Amazon

Sebelum Anda menghapus CloudFront distribusi Amazon, Anda harus menonaktifkannya. Distribusi yang dinonaktifkan tidak lagi berfungsi, dan tidak dikenakan biaya. Anda dapat mengaktifkan distribusi yang dinonaktifkan kapan saja. Setelah Anda menghapus distribusi yang dinonaktifkan, distribusi tersebut tidak lagi tersedia.

Untuk menonaktifkan dan menghapus CloudFront distribusi

1. Buka CloudFront konsol dihttps://console.aws.amazon.com/cloudfront/v4/home.
2. Pilih distribusi yang ingin Anda nonaktifkan, lalu pilih Nonaktifkan.
3. Saat diminta untuk mengonfirmasi, pilih Ya, Nonaktifkan.
4. Pilih distribusi yang dinonaktifkan, lalu pilih Hapus.
5. Saat diminta konfirmasi, pilih Ya, Hapus.

Langkah 2: Menghapus Route 53 zona yang di-hosting

Sebelum menghapus zona yang di-hosting, Anda harus menghapus set rekaman yang Anda buat. Anda tidak perlu menghapus catatan NS dan SOA; catatan ini secara otomatis dihapus ketika Anda menghapus zona yang di-hosting.

Untuk menghapus set catatan

1. Buka konsol Route 53 di https://console.aws.amazon.com/route53/.
2. Di dalam daftar nama domain, pilih nama domain Anda, lalu pilih Masuk ke Set Catatan.
3. Dalam daftar kumpulan rekaman, pilih rekaman A yang Anda buat.

   Jenis setiap set catatan tercantum dalam Jenis kolom.
4. Pilih Hapus Set Catatan.
5. Ketika diminta untuk mengonfirmasi, pilih Konfirmasi.

Untuk menghapus zona Route 53 yang di-hosting

1. Lanjutkan dari prosedur sebelumnya, pilih Kembali ke Zona yang Di-hosting.
2. Pilih nama domain Anda, lalu pilih Hapus Zona yang Di-hosting.
3. Ketika diminta untuk mengonfirmasi, pilih Konfirmasi.

Langkah 3: Menonaktifkan pencatatan dan hapus bucket S3 Anda

Sebelum Anda menghapus bucket S3 Anda, pastikan pembuatan log dinonaktifkan untuk bucket. Jika tidak, AWS terus menulis log ke bucket Anda saat Anda menghapusnya.

Untuk menonaktifkan pembuatan log untuk bucket

1. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di bagian bawah Bucket, pilih nama bucket Anda, lalu pilih Properti.
3. Dari Properti, pilih Pencatatan.
4. Hapus kotak centang Diaktifkan.
5. Pilih Simpan.

Sekarang, Anda dapat menghapus bucket Anda. Untuk informasi selengkapnya, lihat Menghapus bucket tujuan umum.

Menyebarkan situs web statis ke AWS Amplify Hosting dari bucket tujuan umum S3

Kami menyarankan Anda menggunakan AWS Amplify Hosting untuk meng-host konten situs web statis yang disimpan di S3. Amplify Hosting adalah layanan yang dikelola sepenuhnya yang memudahkan penyebaran situs web Anda di jaringan pengiriman konten (CDN) yang tersedia secara global yang didukung oleh Amazon CloudFront, memungkinkan hosting situs web statis yang aman tanpa pengaturan ekstensif. Dengan AWS Amplify Hosting, Anda dapat memilih lokasi objek dalam bucket tujuan umum, menyebarkan konten Anda ke CDN terkelola, dan membuat URL HTTPS publik agar situs web Anda dapat diakses di mana saja. Menyebarkan situs web statis menggunakan Amplify Hosting memberi Anda manfaat dan fitur berikut:

- Penyebaran ke jaringan pengiriman AWS konten (CDN) yang didukung oleh Amazon CloudFront - CloudFront adalah layanan web yang mempercepat distribusi konten web statis dan dinamis Anda kepada pengguna Anda. CloudFront mengirimkan konten Anda melalui jaringan pusat data di seluruh dunia yang disebut lokasi tepi. Saat pengguna meminta konten yang Anda sajikan CloudFront, permintaan akan diarahkan ke lokasi tepi yang memberikan latensi terendah (waktu tunda), sehingga konten dikirimkan dengan kinerja terbaik, peningkatan keandalan, dan ketersediaan. Untuk informasi selengkapnya, lihat Cara CloudFront mengirimkan konten di Panduan CloudFront Pengembang Amazon.

- Dukungan HTTPS - Menyediakan komunikasi dan transfer data yang aman antara situs web Anda dan browser web pengguna.
- Domain khusus - Hubungkan situs web Anda dengan mudah ke URL khusus yang dibeli dari pendaftar domain seperti Amazon Route 53.
- Sertifikat SSL kustom - Saat menyiapkan domain kustom, Anda dapat menggunakan sertifikat terkelola default yang disediakan Amplify untuk Anda atau Anda dapat menggunakan sertifikat kustom Anda sendiri yang dibeli dari otoritas sertifikat pihak ketiga pilihan Anda.
- Dibangun pada metrik dan CloudWatch pemantauan - Memantau lalu lintas, kesalahan, transfer data, dan latensi untuk situs web Anda.
- Perlindungan kata sandi - Batasi akses ke situs web Anda, dengan menyiapkan persyaratan nama pengguna dan kata sandi di konsol Amplify.
- Pengalihan dan penulisan ulang - Buat aturan pengalihan dan tulis ulang di konsol Amplify untuk memungkinkan server web mengalihkan navigasi dari satu URL ke URL lainnya.

Saat Anda menerapkan aplikasi dari bucket tujuan umum Amazon S3 ke Amplify Hosting AWS , biaya didasarkan pada model harga Amplify. Untuk informasi selengkapnya, silakan lihat Harga AWS Amplify.

> **Important**
>
> Amplify Hosting tidak tersedia di semua Wilayah AWS tempat Amazon S3 tersedia. Untuk menerapkan situs web statis ke Amplify Hosting, bucket tujuan umum Amazon S3 yang berisi situs web Anda harus berada di wilayah tempat Amplify tersedia. Untuk daftar wilayah tempat Amplify tersedia, lihat Amplify endpoint di bagian. Referensi Umum Amazon Web Services

Anda dapat memulai proses penerapan dari konsol Amazon S3, konsol Amplify, AWS CLI, atau. AWS SDKs Anda hanya dapat menerapkan ke Amplify dari bucket tujuan umum yang terletak di akun Anda sendiri. Amplify tidak mendukung akses bucket lintas akun.

Gunakan petunjuk berikut untuk menerapkan situs web statis dari bucket tujuan umum Amazon S3 ke Amplify Hosting mulai dari konsol Amazon S3.

Menyebarkan situs web statis ke Amplify dari konsol S3

Untuk menyebarkan situs web statis dari konsol Amazon S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Di panel navigasi kiri, pilih Bucket.
3. Dalam daftar Bucket, pilih bucket tujuan umum yang berisi situs web yang ingin Anda gunakan untuk Amplify Hosting.
4. Pilih tab Properti.
5. Di bawah hosting situs web statis, pilih Buat aplikasi Amplify. Pada langkah ini, proses penyebaran akan pindah ke konsol Amplify.
6. Pada halaman Deploy with S3, lakukan langkah-langkah berikut.
   a. Untuk nama Aplikasi, masukkan nama aplikasi atau situs web Anda.
   b. Untuk nama Branch, masukkan nama backend aplikasi Anda.
   c. Untuk lokasi S3 objek yang akan dihosting, masukkan jalur direktori ke bucket tujuan umum Anda atau pilih Browse S3 untuk mencari dan memilihnya.
7. Pilih Simpan dan deploy.

> **Note**
>
> Jika Anda memperbarui objek apa pun untuk situs web statis di bucket tujuan umum yang dihosting di Amplify, Anda harus menerapkan ulang aplikasi ke Amplify Hosting agar perubahan diterapkan. Amplify Hosting tidak secara otomatis mendeteksi perubahan pada bucket Anda. Untuk informasi selengkapnya, lihat Memperbarui situs web statis yang diterapkan ke Amplify dari bucket S3 di Panduan Pengguna AWS Amplify Hosting.

Untuk memulai langsung dari konsol Amplify, lihat Menerapkan situs web statis dari S3 menggunakan konsol Amplify di Panduan Pengguna Amplify Hosting.AWS

Untuk mulai menggunakan AWS SDKs, lihat Membuat kebijakan bucket untuk menerapkan situs web statis dari S3 menggunakan Panduan Pengguna AWS Amplify Hosting. AWS SDKs

Kuota

Anda Akun AWS memiliki kuota default, sebelumnya disebut sebagai batas, untuk setiap layanan. AWS Kuota Amazon S3 mencakup jumlah bucket tujuan umum, ember direktori, titik akses, dan lainnya. Anda dapat meminta kenaikan untuk beberapa kuota, tetapi tidak semua kuota dapat ditingkatkan. Kenaikan ini tidak diberikan segera, jadi mungkin perlu beberapa hari agar kenaikan Anda menjadi efektif.

Untuk daftar kuota Amazon S3 dan nilai defaultnya, lihat, kuota Amazon S3 di. Referensi Umum AWS

Kuota meningkat

Untuk meminta peningkatan kuota

Anda dapat meminta kenaikan kuota dengan menggunakan salah satu opsi berikut:

- Dari AWS Management Console: Buka konsol Service Quotas. Di panel navigasi, pilih Layanan AWS . Pilih Amazon S3, pilih kuota, dan ikuti petunjuk untuk meminta peningkatan kuota. Untuk instruksi, lihat Meminta peningkatan kuota dalam Panduan Pengguna Service Quotas.
- Dari AWS CLI: Gunakan request-service-quota-increase AWS CLI perintah. Untuk instruksi, lihat Meminta peningkatan kuota dalam Panduan Pengguna Service Quotas.

Referensi

Selain konsol Amazon S3, Anda dapat bekerja dengan Amazon S3 secara terprogram menggunakan Amazon S3 REST,, atau (). APIs AWS SDKs AWS Command Line Interface AWS CLI

- Referensi API Layanan Penyimpanan Sederhana Amazon
- Contoh kode di Referensi API Layanan Penyimpanan Sederhana Amazon
- AWS Antarmuka Baris Perintah di Panduan AWS Command Line Interface Pengguna.
- AWS SDKs

Dokumentasi SDK	Contoh kode
AWS SDK untuk C++	AWS SDK untuk C++ contoh kode
AWS CLI	AWS CLI contoh kode.
AWS SDK untuk Go	AWS SDK untuk Go contoh kode
AWS SDK untuk Java	AWS SDK untuk Java contoh kode
AWS SDK untuk JavaScript	AWS SDK untuk JavaScript contoh kode
AWS SDK untuk Kotlin	AWS SDK untuk Kotlin contoh kode
AWS SDK untuk .NET	AWS SDK untuk .NET contoh kode
AWS SDK untuk PHP	AWS SDK untuk PHP contoh kode
Alat AWS untuk PowerShell	Alat untuk contoh PowerShell kode
AWS SDK untuk Python (Boto3)	AWS SDK untuk Python (Boto3) contoh kode
AWS SDK untuk Ruby	AWS SDK untuk Ruby contoh kode
AWS SDK for Rust	AWS SDK for Rust contoh kode
AWS SDK untuk SAP ABAP	AWS SDK untuk SAP ABAP contoh kode
AWS SDK for Swift	AWS SDK for Swift contoh kode

Riwayat dokumen

- Versi API saat ini:2006-03-01

Tabel berikut menjelaskan perubahan penting dalam setiap rilis Referensi API Amazon Simple Storage Service dan Panduan Pengguna Amazon S3. Untuk notifikasi tentang pembaruan dokumentasi ini, Anda dapat berlangganan ke umpan RSS.

Perubahan	Deskripsi	Tanggal
Amazon S3 Express One Zone sekarang mendukung tag untuk alokasi biaya dan kontrol akses berbasis atribut	Amazon S3 Express One Zone, kelas penyimpanan S3 berkinerja tinggi, sekarang mendukung tag untuk alokasi biaya dan kontrol akses berbasis atribut (ABAC). Anda dapat menambahkan tag ke bucket direktori S3 untuk melacak dan mengatur AWS biaya menggunakan. AWS Manajemen Penagihan dan Biaya Selain itu, dengan dukungan ABAC, Anda dapat menskalakan kontrol akses ke bucket direktori Anda dengan menggunakan tag. S3 Express One Zone mendukung tag pada bucket direktori di semua Wilayah AWS tempat kelas penyimpanan tersedia. Anda dapat memulai dengan menandai menggunakan AWS Management Console, S3 REST API, AWS CLI, atau SDK. AWS Untuk mempelajari lebih lanjut	Juli 2, 2025

	tentang menggunakan tag untuk menyederhanakan alokasi biaya atau ABAC, lihat Menggunakan tag dengan bucket direktori S3.	
Access point sekarang mendukung melampirkan ke volume untuk FSx OpenZFS	Anda dapat membuat dan melampirkan titik akses ke volume FSx untuk OpenZFS menggunakan FSx konsol Amazon, AWS CLI, atau API. Setelah terpasang, Anda dapat menggunakan objek S3 APIs untuk mengakses data file Anda. Data Anda terus berada di sistem FSx file Amazon dan terus dapat diakses secara langsung untuk beban kerja Anda yang ada. Untuk informasi selengkap nya, lihat Mengelola akses ke kumpulan data bersama dengan titik akses.	Juni 25, 2025
Tabel S3 sekarang mendukung strategi pemadatan baru untuk tabel	Tabel S3 sekarang mendukung beberapa strategi pemadatan untuk mengoptim alkan kinerja kueri berdasark an pola akses data yang berbeda. Strategi baru termasuk Sort dan Z-order. Untuk informasi selengkapnya, lihat Strategi Pemadatan untuk tabel.	Juni 18, 2025

Tabel S3 menambahkan kebijakan AWS terkelola baru untuk Tabel S3	Tabel S3 menambahkan kebijakan AWS-managed baru yang disebut. `AmazonS3TablesLakeFormationServiceRole` Kebijakan ini memberikan izin yang memungkinkan akses peran layanan Lake Formation ke tabel Amazon S3 dan bucket tabel. Untuk informasi selengkapnya, lihat kebijakan AWS terkelola untuk Tabel S3.	19 Mei 2025
Tabel S3 sekarang mendukung SSE-KMS menggunakan kunci yang dikelola pelanggan	Tabel S3 sekarang mendukung enkripsi sisi server dengan kunci AWS Key Management Service (AWS KMS) (SSE-KMS) menggunakan kunci yang dikelola pelanggan. Anda dapat menerapkan enkripsi SSE-KMS di bucket meja dan tingkat tabel. Untuk informasi selengkapnya, lihat Menggunakan enkripsi sisi server dengan AWS KMS kunci (SSE-KMS) di bucket tabel.	April 16, 2025

Titik akses untuk bucket direktori tersedia di AWS Local Zones	Bucket direktori sekarang mendukung titik akses untuk menyederhanakan pengelola an akses data dalam skala besar untuk kumpulan data bersama di Amazon S3. Dengan fitur baru ini, Anda dapat membuat ratusan titik akses per ember, masing-masing dengan nama dan izin berbeda yang disesuaik an untuk setiap aplikasi. Untuk informasi selengkap nya, lihat Mengelola akses ke kumpulan data bersama di bucket direktori dengan titik akses.	Maret 31, 2025

Integrasi bucket meja S3 dengan Amazon SageMaker Lakehouse sekarang tersedia secara umum	Anda dapat menginteg rasikan bucket tabel S3 dengan Amazon SageMaker Lakehouse untuk mengakses tabel dari layanan AWS analitik, seperti Amazon Athena, Amazon Redshift, dan. QuickSight Amazon SageMaker Lakehouse menyatukan data Anda di seluruh danau data Amazon S3 dan gudang data Amazon Redshift, sehingga Anda dapat membangun analitik, pembelajaran mesin, dan aplikasi AI generatif pada satu salinan data. Integrasi mengisi AWS Glue Data Catalog dengan sumber daya tabel Anda, dan menggabungkan akses ke sumber daya ini dengan. AWS Lake Formation Integrasi ini memungkin kan kontrol akses berbutir halus melalui Lake Formation untuk memberikan keamanan tambahan. Untuk informasi selengkapnya tentang integrasi, lihat Menggunak an Tabel Amazon S3 AWS dengan layanan analitik. Jika Anda mengatur integrasi dengan rilis pratinjau, Anda dapat terus menggunak an integrasi Anda saat ini. Namun, proses integrasi	Maret 13, 2025

	yang diperbarui memberikan peningkatan kinerja, jadi kami sarankan untuk bermigrasi. Untuk bermigrasi ke integrasi yang diperbarui, lihat Memigrasi ke proses integrasi yang diperbarui.	
Tabel S3 membuat tabel dan dukungan tabel kueri yang ditambahkan ke konsol Amazon S3	Tabel S3 sekarang mendukung pembuatan tabel dan tabel kueri operasi langsung dari konsol Amazon S3 dengan menggunakan Amazon Athena. Dengan fitur baru ini, Anda sekarang dapat membuat tabel, mengisinya dengan data, dan menanyakannya hanya dengan beberapa langkah di konsol Amazon S3. Untuk informasi selengkapnya, lihat Membuat tabel Amazon S3 dan Menanyakan tabel Amazon S3 dengan Athena.	Maret 13, 2025
Tabel S3 menambahkan dukungan untuk menambahkan skema saat membuat tabel	Sekarang Anda dapat menggunakan operasi S3 Tables `CreateTable` API untuk membuat tabel dengan skema dengan menambahkan flag metadata opsional. Untuk informasi selengkapnya, lihat Membuat tabel.	Januari 30, 2025

Tabel S3 menambahkan dukungan untuk menambahkan skema saat membuat tabel	Sekarang Anda dapat menggunakan operasi S3 Tables `CreateTable` API untuk membuat tabel dengan skema dengan menambahkan flag metadata opsional. Untuk informasi selengkapnya, lihat Membuat tabel.	Januari 30, 2025

Metadata S3 sekarang tersedia secara umum	Amazon S3 Metadata sekarang tersedia secara umum. Metadata S3 membantu Anda dengan mudah menemukan dan memahami data S3 Anda dengan metadata otomatis yang dapat dikueri yang diperbarui dalam waktu dekat. Dengan Metadata S3, Anda dapat mengkurasi, mengident ifikasi, dan menggunakan data S3 Anda untuk analisis bisnis, kecerdasan buatan, dan pelatihan model pembelaja ran mesin (AI/ML), dan banyak lagi. Metadata S3 mendukung metadata objek, yang mencakup informasi yang ditentukan sistem seperti ukuran dan sumber objek, dan metadata khusus, seperti tag dengan informasi seperti produk, transaksi, atau peringkat konten. SKUs IDs Untuk informasi selengkapnya, lihat Mempercepat penemuan data dengan Metadata S3.	Januari 27, 2025
AWS kebijakan terkelola — Kebijakan baru	Tabel S3 menambahkan dua kebijakan AWS terkelola baru.	Desember 6, 2024

Tabel S3	Tabel Amazon S3 menyediakan penyimpanan S3 yang dioptimalkan untuk beban kerja analitik, dengan fitur yang meningkatkan kinerja kueri, mengurangi biaya penyimpanan untuk tabel, dan menyederhanakan pengoperasian data lake dalam skala besar. Tabel S3 memperkenalkan jenis bucket baru: bucket tabel, yang dibuat khusus untuk menyimpan tabel sebagai subresource. Apache Iceberg Bucket tabel menyediakan transaksi per detik (TPS) yang lebih tinggi dan throughput kueri yang lebih baik dibandingkan dengan tabel yang dikelola sendiri di bucket tujuan umum S3. Anda dapat secara otomatis mengintegrasikan bucket tabel Anda dengan layanan AWS analitik, seperti Athena, Amazon QuickSight Redshift, dan banyak lagi. Untuk informasi selengkapnya, lihat Bekerja dengan Tabel Amazon S3 dan bucket tabel.	Desember 3, 2024

Pratinjau Metadata S3	Metadata Amazon S3 membantu Anda dengan mudah menemukan dan memahami data S3 Anda dengan metadata otomatis yang dapat dikueri yang diperbarui dalam waktu dekat. Dengan Metadata S3, Anda dapat mengkurasi, mengident ifikasi, dan menggunakan data S3 Anda untuk analisis bisnis, kecerdasan buatan, dan pelatihan model pembelaja ran mesin (AI/ML), dan banyak lagi. Metadata S3 mendukung metadata objek, yang mencakup informasi yang ditentukan sistem seperti ukuran dan sumber objek, dan metadata khusus, seperti tag dengan informasi seperti produk, transaksi, atau peringkat konten. SKUs IDs Untuk informasi selengkapnya, lihat Mempercepat penemuan data dengan Metadata S3.	Desember 3, 2024

Browser Penyimpanan untuk S3	Browser Penyimpanan untuk S3 adalah komponen open source yang dapat Anda tambahkan ke aplikasi web Anda untuk menyediakan pengguna akhir Anda dengan antarmuka sederhana untuk data yang disimpan di S3. Untuk informasi selengkapnya, lihat Browser Penyimpanan untuk S3.	Desember 1, 2024
Algoritma checksum Amazon S3 baru dan fitur integritas checksum yang ditingkatkan	Amazon S3 menambahkan algoritma checksum dan `CRC-64NVME` fitur integritas checksum yang ditingkatkan. Untuk informasi selengkapnya, lihat Memeriksa integritas objek di Amazon S3.	Desember 1, 2024
Beban kerja residensi data	Di Local Zones Khusus, Anda dapat membuat bucket direktori S3 untuk menyimpan data untuk kasus penggunaan residensi dan isolasi data Anda. Untuk informasi selengkapnya, lihat Beban kerja residensi data.	Desember 1, 2024

Kunci kondisi baru untuk menegakkan penulisan bersyarat	Amazon S3 menambahkan kunci kondisi baru `s3:if-match` dan `s3:if-non e-match` digunakan dalam kebijakan bucket untuk memaksa klien menggunak an header `If-None-Match` atau `If-Match` HTTP. Untuk informasi selengkapnya, lihat Menegakkan penulisan bersyarat di bucket Amazon S3.	November 25, 2024
Header HTTP baru untuk penulisan bersyarat untuk memeriksa apakah objek telah berubah	Amazon S3 menambahkan header `If-Match` HTTP untuk memeriksa tag entitas objek (ETag) sebelum menulis objek untuk beberapa operasi API. Dengan header ini, Amazon S3 membandingkan nilai yang diberikan dengan ETag nilai objek di ETag S3. Jika ETag nilainya tidak cocok, operasi gagal. Untuk informasi selengkapnya, lihat Cara mencegah penimpaan objek dengan penulisan bersyarat.	November 25, 2024

Amazon Redshift sekarang terintegrasi dengan S3 Access Grants	Pelanggan Amazon Redshift sekarang dapat menggunakan S3 Access Grants untuk menskalakan dan mengelola izin untuk data S3 mereka. Hal ini memungkinkan pelanggan Amazon Redshift untuk menskalakan izin S3 untuk identitas perusahaan dengan menggunakan AWS IAM Identity Center serta untuk pengguna dan grup IAM. Untuk informasi selengkapnya, lihat Integrasi Amazon Redshift dengan Amazon S3 Access Grants.	November 15, 2024
AWS Organizations akun anggota sekarang dapat mendapatkan kembali akses ke ember Amazon S3 yang terkunci secara tidak sengaja	AWS Organizations akun anggota sekarang dapat menggunakan proses sederhana melalui AWS Identity and Access Management (IAM) untuk mendapatkan kembali akses ke bucket Amazon S3 yang terkunci secara tidak sengaja. Untuk informasi selengkapnya, lihat Melakukan tugas istimewa pada akun AWS Organizations anggota di Panduan AWS Identity and Access Management Pengguna.	November 14, 2024

Kebijakan kontrol sumber daya (RCPs), jenis kebijakan otorisasi baru juga AWS Organizations tersedia untuk bucket Amazon S3	Kebijakan kontrol sumber daya (RCPs), kebijakan otorisasi baru yang dikelola AWS Organizations dapat digunakan untuk menyetel izin maksimum yang tersedia di bucket Amazon S3 di seluruh organisasi Anda. Untuk informasi selengkapnya, lihat Kebijakan kontrol sumber daya (RCPs) di Panduan AWS Organizations Pengguna.	November 13, 2024
Amazon S3 sekarang secara otomatis menyetujui peningkatan kuota bucket hingga 1.000 bucket	Amazon S3 sekarang secara otomatis menyetujui peningkatan kuota bucket hingga 1.000 bucket. Untuk melihat penggunaan bucket Anda atau meminta peningkatan, kunjungi konsol Service Quotas.	September 30, 2024
Perilaku transisi ukuran objek minimum default baru untuk konfigurasi Siklus Hidup Amazon S3	Amazon S3 sekarang menerapkan perilaku default ke konfigurasi Siklus Hidup S3 yang mencegah objek yang lebih kecil dari 128 KB dialihkan ke kelas penyimpanan apa pun. Untuk mempelajari cara mengganti perilaku ini, lihat Mengizinkan objek yang lebih kecil dari 128 KB untuk dialihkan.	September 24, 2024

Bucket direktori sekarang mendukung SSE-KMS menggunakan kunci yang dikelola pelanggan.	Bucket direktori sekarang mendukung enkripsi sisi server dengan kunci AWS Key Management Service (AWS KMS) (SSE-KMS) menggunakan kunci yang dikelola pelanggan. Anda memiliki lebih banyak opsi untuk mengenkripsi dan mengelola keamanan data Anda dalam ember direktori. Untuk informasi selengkapnya, lihat Perlindungan data dan enkripsi di bucket direktori.	September 17, 2024
S3 Access Grants mendukung tindakan API yang mencantumkan hibah akses pemanggil	Identitas IAM dan identitas direktori perusahaan IAM Identity Center sekarang dapat menggunakan `ListCallerAccessGrants` API untuk mencantumkan semua bucket, awalan, dan objek Amazon S3 yang dapat mereka akses, seperti yang ditentukan oleh Hibah Akses S3 mereka. Gunakan API ini untuk menemukan semua data S3 yang dapat diakses oleh IAM atau identitas direktori perusahaan melalui S3 Access Grants, dalam data tertentu. Akun AWS Untuk informasi selengkapnya, lihat Daftar hibah akses pemanggil.	September 5, 2024

Akses yang disempurnakan menolak pesan kesalahan untuk permintaan akun yang sama	Amazon S3 sekarang menyertakan konteks tambahan dalam kesalahan akses ditolak (HTTP `403 Forbidden`) untuk permintaan yang dibuat ke sumber daya dalam hal yang sama. Akun AWS Konteks baru ini mencakup jenis kebijakan yang menolak akses, alasan penolakan, dan informasi tentang pengguna AWS Identity and Access Management (IAM) atau peran yang meminta akses ke sumber daya. Konteks ini membantu Anda memecahkan masalah akses, mengidentifikasi akar penyebab kesalahan akses ditolak, dan memperbaiki kontrol akses yang salah dengan memperbarui kebijakan yang relevan. Konteks tambahan ini juga tersedia di AWS CloudTrail log. Pesan kesalahan yang ditolak akses yang ditingkatkan untuk permintaan akun yang sama sekarang tersedia di semua Wilayah AWS, termasuk AWS GovCloud (US) Regions dan Wilayah Tiongkok. Untuk informasi selengkapnya, lihat Memecahkan masalah	Agustus 21, 2024

	kesalahan akses ditolak (403 Terlarang) di Amazon S3. .	
Amazon S3 mendukung penggunaan penulisan bersyarat untuk dan `PutObject` `CompleteMultipartUpload`	Anda dapat memeriksa keberadaan objek di bucket Anda sebelum membuatnya menggunakan penulisan bersyarat pada operasi unggahan. Ini dapat mencegah penimpaan data yang ada. Penulisan bersyarat akan memvalidasi tidak ada objek yang ada dengan nama kunci yang sama di bucket Anda. Untuk informasi selengkapnya, lihat Permintaan bersyarat.	Agustus 20, 2024
Amazon S3 tidak lagi mengenakan biaya untuk beberapa kode kesalahan HTTP	Amazon S3 telah menyelesaikan perubahan sehingga permintaan tidak sah yang tidak dilakukan pelanggan tidak dikenai biaya. Untuk informasi selengkapnya, lihat Respons kesalahan penagihan Amazon S3.	Agustus 19, 2024
Amazon S3 Select tidak lagi tersedia untuk pelanggan baru	Amazon S3 Select tidak lagi tersedia untuk pelanggan baru. Pelanggan Amazon S3 Select yang sudah ada dapat terus menggunakan fitur seperti biasa. Pelajari selengkapnya.	Juli 25, 2024

Inventaris Amazon S3 mendukung kunci kondisi s3: InventoryAccessibleOptionalFields	Inventaris Amazon S3 mendukung kunci Inventory AccessibleOptionalFields kondisi s3: untuk mengontrol apakah pengguna dapat menyertakan bidang metadata opsional dalam laporan mereka. Untuk informasi selengkapnya, lihat Mengontrol pembuatan konfigurasi laporan Inventaris S3.	Februari 20, 2024
IPv6 dukungan untuk S3 di Outposts	Anda sekarang dapat mengakses S3 di bucket Outposts IPv6 menggunakan via S3 pada titik akhir dual-stack Outposts. IPv6 dukungan untuk S3 di Outposts memungkinkan Anda untuk mengelola S3 Anda pada ember Outposts dan mengontrol sumber daya pesawat melalui jaringan. IPv6	Januari 16, 2024
Kelas penyimpanan Amazon S3 zona tunggal berkinerja tinggi baru–S3 Express One Zone	Amazon S3 Express One Zone adalah kelas penyimpanan Amazon S3 tunggal berkinerja tinggi yang dibuat khusus untuk menghadirkan akses data milidetik satu digit yang konsisten untuk aplikasi Anda yang paling sensitif terhadap latensi. Untuk informasi selengkapnya, lihat S3 Express One Zone.	28 November 2023

Mountpoint untuk Amazon S3 menambahkan dukungan untuk S3 Express One Zone	Anda sekarang dapat memasang bucket direktori S3 Express One Zone dengan Mountpoint.	28 November 2023
Versi skema invokasi Lambda	Operasi Batch Amazon S3 memperkenalkan versi skema invokasi Lambda baru untuk digunakan dengan tugas Operasi Batch yang bekerja pada bucket direktori. Untuk informasi selengkapnya, lihat Menggunakan Lambda dan operasi batch Amazon S3 dengan bucket direktori.	28 November 2023
Tindakan impor untuk bucket direktori	Amazon S3 memperken alkan tindakan impor. Impor adalah metode yang disederha nakan untuk membuat tugas Operasi Batch Amazon S3 guna menyalin objek dari bucket tujuan umum ke bucket direktori. Untuk informasi selengkapnya, lihat Mengimpor objek ke dalam bucket direktori.	28 November 2023

Mengelola akses S3 dengan Pemberian Akses S3	Pemberian Akses Amazon S3 memungkinkan Anda mengelola izin data dalam skala besar untuk pengguna utama (IAM) AWS Identity and Access Management selain identitas direktori dari direktori perusahaan seperti Azure AD. Anda sekarang dapat menerapkan izin S3 dengan hak akses paling rendah dan dengan mudah menskalak an izin tersebut berdasark an kebutuhan bisnis Anda. Untuk informasi selengkapnya, lihat Mengelola akses dengan Pemberian Akses S3.	26 November 2023
Mountpoint untuk Amazon S3 menambahkan fitur caching	Dengan Mountpoint, Anda sekarang dapat mengonfig urasi caching untuk data yang diakses berulang kali.	22 November 2023

Pembuatan manifes Operasi Batch Amazon S3 yang Ditingkatkan	Sekarang Anda dapat mengarahkan Operasi Batch Amazon S3 untuk menghasilkan manifes secara otomatis berdasarkan kriteria filter objek yang Anda tentukan saat membuat tugas. Opsi ini tersedia untuk pekerjaan replikasi batch yang Anda buat di konsol Amazon S3, atau untuk jenis pekerjaan apa pun yang Anda buat dengan menggunakan AWS CLI AWS SDKs,, atau Amazon S3 REST API. Untuk informasi selengkapnya, lihat Membuat tugas Operasi Batch Amazon S3.	22 November 2023
Bucket Amazon S3 yang ada sekarang dapat menambahkan konfigurasi Kunci Objek	Anda kini dapat mengaktifkan Kunci Objek di bucket Amazon S3 yang ada. Anda dapat menetapkan periode retensi dan penahanan hukum untuk bucket baru atau yang sudah ada. Untuk informasi selengkapnya, lihat Menggunakan Kunci Objek.	20 November 2023
Lensa Penyimpanan S3 meminta metrik untuk awalan	Lensa Penyimpanan S3 memperkenalkan metrik permintaan untuk awalan dalam bucket Amazon S3. Untuk informasi selengkapnya, lihat Kategori metrik.	17 November 2023

Grup Lensa Penyimpanan Amazon S3	Lensa Penyimpanan S3 memperkenalkan grup Lensa Penyimpanan, filter yang ditentukan khusus untuk objek berdasarkan metadata objek. Untuk informasi selengkap nya, lihat Bekerja dengan grup Lensa Penyimpanan Amazon S3.	15 November 2023
Kebijakan IAM baru	S3 on Outposts memperken alkan `AWSServic eRoleForS3OnOutpos ts` , peran terkait layanan untuk membantu mengelola sumber daya jaringan untuk Anda. Untuk informasi selengkapnya, lihat Menggunakan peran terkait layanan untuk S3 on Outposts.	3 Oktober 2023
Amazon S3 menyediakan waktu `Last-Modified` untuk menghapus penanda hapus	Amazon S3 menyediakan waktu `Last-Modified` penanda hapus di header respons operasi S3 `Head` dan API `Get`. Untuk informasi selengkapnya, lihat Bekerja dengan penanda hapus.	27 September 2023
Pembaruan Amazon S3 ke AWS kebijakan terkelola	Amazon S3 menambahk an izin `s3:Describe*` ke `AmazonS3ReadOnlyAc cess` . Untuk informasi selengkapnya, lihat kebijakan terkelola AWS untuk Amazon S3.	11 Agustus 2023

Peningkatan waktu mulai untuk permintaan pemulihan Standar yang dilakukan melalui Operasi Batch S3	Pengambilan Standar untuk permintaan pemulihan yang dilakukan melalui Operasi Batch S3 sekarang dapat dimulai dalam beberapa menit. Untuk informasi selengkapnya, lihat Opsi Pengambilan Arsip.	9 Agustus 2023
Menambahkan Mountpoint, klien dengan throughput tinggi untuk memasang bucket Amazon S3 sebagai sistem file lokal.	Dengan Mountpoint, aplikasi Anda dapat mengakses objek yang disimpan di Amazon S3 melalui operasi file, memberi aplikasi Anda akses ke penyimpanan elastis dan throughput Amazon S3 melalui antarmuka file.	9 Agustus 2023
Enkripsi sisi server dua lapis dengan kunci (DSSE-KMS) AWS Key Management Service	Enkripsi sisi server dua lapis dengan kunci AWS Key Management Service (AWS KMS) (DSSE-KMS) menerapkan dua lapisan enkripsi ke objek saat diunggah ke Amazon S3. Untuk informasi selengkapnya, lihat Menggunakan enkripsi sisi server dua lapis dengan kunci. AWS KMS	13 Juni 2023

Amazon S3 mengaktifkan S3 Block Public Access dan menonaktifkan daftar kontrol akses S3 (ACLs) untuk semua bucket baru.	Amazon S3 sekarang secara otomatis mengaktifkan S3 Block Public Access dan menonaktifkan daftar kontrol akses S3 (ACLs) untuk semua bucket S3 baru di semua Wilayah. AWS Untuk informasi selengkapnya, lihat Memblokir akses publik ke penyimpanan Amazon S3 dan Mengontrol kepemilikan objek serta menonaktifkan bucket ACLs Anda.	27 April 2023
Metrik Operasi Replikasi S3 yang Gagal	Amazon S3 menambahkan Amazon CloudWatch metrik baru untuk memantau kegagalan Replikasi S3. Untuk informasi selengkapnya, lihat Memantau progres dengan metrik replikasi.	5 April 2023
DNS privat	AWS PrivateLink untuk Amazon S3 sekarang mendukung DNS Pribadi. Untuk informasi lebih lanjut, lihat DNS Privat.	14 Maret 2023
Dukungan titik akses lintas akun di konsol Amazon S3	Amazon S3 sekarang mendukung pembuatan titik akses lintas akun dengan konsol Amazon S3. Untuk informasi selengkapnya, lihat Membuat titik akses.	14 Maret 2023

Amazon S3 on Outposts mendukung Replikasi S3 di Outposts	Dengan Replikasi S3 lokal, Anda dapat secara otomatis mereplikasi objek ke satu bucket tujuan Outposts atau ke beberapa bucket tujuan. Bucket tujuan dapat berada di Outposts yang berbeda AWS Outposts atau dalam Outposts yang sama dengan bucket sumber. Untuk informasi selengkapnya, lihat Mereplikasi objek untuk S3 on Outposts.	14 Maret 2023
Alias Titik Akses Lambda Objek Amazon S3	Saat Anda membuat Titik Akses Lambda Objek, Amazon S3 secara otomatis menghasilkan alias unik untuk Titik Akses Lambda Objek Anda. Anda dapat menggunakan alias ini alih-alih nama bucket Amazon S3 atau Amazon Resource Name (ARN) Titik Akses Lamda Objek dalam permintaan untuk operasi bidang data titik akses. Untuk informasi selengkapnya, lihat Cara menggunakan alias bergaya bucket untuk Titik Akses Object Lambda.	14 Maret 2023

Dukungan lintas akun Titik Akses Multiwilayah Amazon S3	Amazon S3 sekarang mendukung pembuatan Titik Akses Multiwilayah lintas akun dengan konsol Amazon S3. Untuk informasi selengkap nya, lihat Membuat Titik Akses Multiwilayah.	14 Maret 2023
Titik akses lintas akun	Amazon S3 mendukung pembuatan titik akses lintas akun. Anda dapat membuat titik akses lintas akun dengan menggunakan AWS Command Line Interface (AWS CLI) atau operasi `CreateAccessPoint` API REST. Untuk informasi selengkapnya, lihat Membuat titik akses.	30 November 2022
Amazon S3 mendukung kontrol failover untuk Titik Akses Multiwilayah Amazon S3	Amazon S3 memperken alkan kontrol failover untuk Titik Akses Multiwilayah. Kontrol ini memungkinkan Anda mengalihkan lalu lintas permintaan akses data S3 yang dirutekan melalui Titik Akses Multiwilayah Amazon S3 ke Wilayah AWS alternati f dalam beberapa menit untuk menguji dan membangun aplikasi yang sangat tersedia. Untuk informasi selengkapnya, lihat kontrol failover Titik Akses Multiwilayah Amazon S3.	28 November 2022

Lensa Penyimpanan Amazon S3 meningkatkan visibilitas seluruh organisasi dengan 34 metrik baru	Lensa Penyimpanan S3 memperkenalkan 34 metrik tambahan untuk mengungkap peluang pengoptimalan biaya yang lebih dalam, mengidentifikasi praktik terbaik perlindungan data, dan meningkatkan kinerja alur kerja aplikasi. Untuk informasi selengkapnya, lihat metrik Lensa Penyimpanan S3.	17 November 2022
Amazon S3 mendukung tingkat permintaan pemulihan yang lebih tinggi untuk S3 Glacier Flexible Retrieval dan S3 Glacier Deep Archive	Amazon S3 mendukung permintaan pemulihan dengan kecepatan hingga 1.000 transaksi per detik, per Akun AWS untuk kelas penyimpanan S3 Glacier Flexible Retrieval dan S3 Glacier Deep Archive.	15 November 2022

Amazon S3 on Outposts mendukung tindakan dan filter Siklus Hidup S3	S3 on Outposts mendukung aturan Siklus Hidup S3 tambahan untuk mengoptim alkan manajemen kapasitas. Anda dapat mengakhiri objek seiring bertambahnya usia atau diganti dengan versi yang lebih baru. Anda dapat membuat aturan siklus hidup untuk seluruh bucket atau subset objek dalam bucket dengan melakukan filter dengan awalan, tag objek, atau ukuran objek. Untuk informasi selengkapnya, lihat Membuat dan mengelola konfigurasi siklus hidup.	2 November 2022
Dukungan Replikasi S3 untuk objek SSE-C	Anda dapat mereplikasi objek yang dibuat menggunak an enkripsi di sisi server dengan kunci yang disediaka n pelanggan. Untuk informasi selengkapnya tentang mereplikasi objek terenkripsi, lihat Mereplikasi objek yang dibuat dengan enkripsi di sisi server (SSE-C, SSE-S3, SSE-KMS).	24 Oktober 2022

Amazon S3 on Outposts mendukung alias titik akses	Dengan S3 di Outposts, Anda harus menggunakan titik akses untuk mengakses objek apa pun dalam bucket Outposts. Setiap kali Anda membuat titik akses untuk bucket, S3 di Outposts secara otomatis menghasil kan alias titik akses. Anda dapat menggunakan alias titik akses ini alih-alih ARN titik akses untuk operasi bidang data apa pun. Untuk informasi selengkapnya, lihat Menggunakan alias bergaya bucket untuk titik akses bucket S3 on Outposts.	21 Oktober 2022
S3 Object Lambda mendukung operasi `HeadObject` , `ListObjects` , dan `ListObjectsV2`	Anda dapat menggunakan kode kustom untuk memodifik asi data yang dikembalikan oleh permintaan S3 `GET`, `LIST`, atau `HEAD` standar untuk melakukan filter baris, mengubah ukuran citra secara dinamis, menyamarkan data rahasia, dan banyak lagi. Untuk informasi selengkapnya, lihat Mengubah objek dengan S3 Object Lambda.	4 Oktober 2022

Amazon S3 on Outposts mendukung Penentuan Versi S3	Saat diaktifkan, S3 Penentuan Versi menyimpan beberapa salinan objek yang berbeda dalam bucket yang sama. Anda dapat menggunakan penentuan versi S3 untuk menyimpan, mengambil, dan memulihkan setiap versi dari setiap objek yang disimpan dalam bucket Outposts Anda. Penentuan Versi S3 membantu Anda memulihkan dari tindakan pengguna yang tidak diinginkan dan kegagalan aplikasi. Untuk informasi selengkapnya, lihat Mengelola Penentuan Versi S3 untuk bucket S3 on Outposts.	21 September 2022
AWS Backup untuk Amazon S3	AWS Backup adalah layanan berbasis kebijakan yang dikelola sepenuhnya yang dapat Anda gunakan untuk menentukan kebijakan pencadangan pusat untuk melindungi data Amazon S3 Anda. Untuk informasi selengkapnya, lihat Menggunakan AWS Backup untuk Amazon S3.	18 Februari 2022

Gunakan Replikasi Batch S3 untuk mereplikasi objek yang ada	Dengan Replikasi Batch S3, Anda dapat mereplikasi objek yang ada sebelum konfigurasi replikasi dilakukan. Mereplikasi objek yang ada dilakukan melalui penggunaan tugas Operasi Batch. Replikasi Batch S3 berbeda dengan replikasi langsung, yang secara terus menerus dan otomatis menyalin objek baru di seluruh bucket Amazon S3. Untuk informasi selengkapnya, lihat Mereplikasi objek yang ada dengan Replikasi Batch S3.	8 Februari 2022
Mengganti nama S3 Glacier Instant Retrieval	Kelas penyimpanan Glacier telah diubah namanya menjadi S3 Glacier Flexible Retrieval. Perubahan ini tidak memengaruhi API.	30 November 2021

Pengaturan Kepemilikan Objek S3 baru untuk menonaktifkan ACLs	Anda dapat menerapkan pengaturan yang diberlakukan pemilik bucket untuk Object Ownership ACLs untuk menonaktifkan bucket dan objek di dalamnya dan mengambil kepemilikan setiap objek di bucket Anda. Pengaturan yang diberlakukan pemilik bucket menyederhanakan manajemen akses untuk data yang disimpan di Amazon S3. Untuk informasi selengkapnya, lihat Mengontrol kepemilikan objek dan menonaktifkan bucket ACLs Anda.	30 November 2021
Kelas penyimpanan S3 Intelligent-Tiering	S3 Intelligent-Tiering Archive Instant Access adalah kelas penyimpanan tambahan dalam S3 Intelligent-Tiering. Untuk informasi selengkapnya, lihat Cara kerja S3 Intelligent-Tiering.	30 November 2021
Kelas penyimpanan S3 Glacier Instant Retrieval	Anda sekarang dapat menempatkan objek dalam kelas penyimpanan S3 Glacier Instant Retrieval. Untuk informasi selengkapnya tentang kelas penyimpanan ini, lihat Menggunakan kelas penyimpanan Amazon S3.	30 November 2021

AWS Backup untuk Pratinjau Amazon S3	AWS Backup adalah layanan berbasis kebijakan yang dikelola sepenuhnya yang dapat Anda gunakan untuk menentukan kebijakan pencadangan pusat untuk melindungi data Amazon S3 Anda. Untuk informasi selengkapnya, lihat Menggunakan AWS Backup untuk Amazon S3.	30 November 2021
AWS Identity and Access Management Access Analyzer untuk Amazon S3	Penganalisis Akses IAM menjalankan pemeriksaan kebijakan untuk memvalida si kebijakan Anda terhadap tata bahasa kebijakan IAM dan praktik terbaik. Untuk mempelajari validasi kebijakan menggunakan IAM Access Analyzer lebih lanjut, lihat Validasi kebijakan IAM Access Analyzer di Panduan Pengguna IAM.	30 November 2021
Tipe peristiwa baru	Tipe peristiwa baru yang ditambahkan ke Notifikasi Peristiwa Amazon S3, lihat Notifikasi Peristiwa Amazon S3.	29 November 2021
Aktifkan Amazon EventBridge di ember	Anda dapat mengaktifkan bucket EventBridge Amazon S3 untuk mengirim acara ke Amazon EventBridge, lihat Menggunakan. EventBridge	29 November 2021

Filter Siklus Hidup S3 baru	Anda dapat membuat aturan siklus hidup berdasarkan ukuran objek atau menentukan berapa banyak versi objek lama yang akan disimpan. Untuk informasi selengkapnya, lihat Contoh konfigurasi Siklus Hidup S3.	23 November 2021
Publikasikan metrik Lensa Penyimpanan Amazon S3 ke Amazon CloudWatch	Anda dapat memublikasikan metrik penggunaan dan aktivitas Lensa Penyimpanan S3 CloudWatch ke Amazon untuk membuat tampilan terpadu kesehatan operasional Anda di dasbor. CloudWatch Anda juga dapat menggunakan CloudWatch fitur, seperti alarm dan tindakan yang dipicu, matematika metrik, dan deteksi anomali, untuk memantau dan mengambil tindakan pada metrik Lensa Penyimpanan S3. Selain itu, CloudWatch APIs aktifkan aplikasi, termasuk penyedia pihak ketiga, untuk mengakses metrik Lensa Penyimpanan S3 Anda. Untuk informasi selengkapnya, lihat metrik Lensa Penyimpanan Monitor S3 di. CloudWatch	22 November 2021

Titik Akses Multiwilayah	Anda dapat menggunakan Titik Akses Multiwilayah untuk membuat titik akhir global yang dapat digunakan aplikasi untuk memenuhi permintaan dari bucket Amazon S3 yang terletak di beberapa Wilayah AWS. Anda dapat menggunakan Titik Akses Multiwilayah ini untuk merutekan data ke bucket dengan latensi terendah. Untuk informasi selengkapnya tentang Titik Akses Multiwilayah dan cara menggunakannya, lihat Titik Akses Multiwilayah di Amazon S3.	2 September 2021
Amazon S3 on Outposts menambahkan akses lokal langsung untuk aplikasi	Jalankan aplikasi Anda di luar AWS Outposts virtual private cloud (VPC) dan akses data S3 Anda di Outposts. Anda juga dapat mengakses objek S3 on Outposts langsung dari jaringan on-premise Anda. Untuk informasi selengkapnya tentang mengonfigurasi S3 pada titik akhir Outposts menggunakan alamat IP (CoIP) milik pelanggan dan mengakses objek Anda dengan membuat gateway lokal dari jaringan on-premise, lihat Mengakses Amazon S3 on Outposts menggunakan titik akses khusus VPC.	29 Juli 2021

Alias titik akses Amazon S3	Saat Anda membuat titik akses, Amazon S3 secara otomatis membuat alias yang dapat Anda gunakan alih-alih nama bucket untuk akses data. Anda dapat menggunakan alias titik akses ini alih-alih Amazon Resource Name (ARN) untuk operasi bidang data titik akses apa pun. Untuk informasi selengkapnya, lihat Menggunakan alias bergaya bucket untuk titik akses Anda.	26 Juli 2021
Inventaris Amazon S3 dan Operasi Batch S3 mendukung status Kunci Bucket S3	Inventaris Amazon S3 dan Operasi Batch mendukung pengidentifikasian dan penyalinan objek yang ada dengan Kunci Bucket S3. Kunci Bucket S3 mempercepat pengurangan biaya enkripsi di sisi server untuk objek yang sudah ada. Untuk informasi selengkapnya, lihat Inventaris Amazon S3 dan Objek Penyalinan Operasi Batch.	3 Juni 2021

Snapshot akun metrik Lensa Penyimpanan Amazon S3	Snapshot akun Lensa Penyimpanan S3 menampilkan total penyimpanan, jumlah objek, dan ukuran objek rata-rata Anda di halaman beranda konsol S3 (Bucket) dengan meringkas metrik dari dasbor default Anda. Untuk informasi selengkapnya, lihat Snapshot akun metrik Lensa Penyimpanan S3.	5 Mei 2021
Peningkatan dukungan titik akhir Amazon S3 on Outposts	S3 on Outposts sekarang mendukung hingga 100 titik akhir per Outpost. Untuk informasi selengkapnya, lihat Pembatasan jaringan S3 on Outposts.	29 April 2021
Amazon S3 tentang pemberitahuan acara Outposts di Acara Amazon CloudWatch	Anda dapat menggunakan CloudWatch Events untuk membuat aturan untuk menangkap S3 pada peristiwa Outposts API dan mendapatkan pemberitahuan melalui semua target yang didukung. CloudWatch Untuk informasi selengkapnya, lihat Menerima pemberitahuan CloudWatch acara S3 di Outposts menggunakan Acara.	19 April 2021

S3 Object Lambda	Dengan S3 Object Lambda, Anda dapat menambahkan kode Anda sendiri ke permintaan GET Amazon S3 untuk memodifikasi dan memproses data saat dikembalikan ke aplikasi. Anda dapat menggunakan kode khusus untuk memodifikasi data yang dikembalikan oleh permintaan GET S3 standar untuk melakukan filter baris, mengubah ukuran citra secara dinamis, menyamarkan data rahasia, dan banyak lagi. Untuk informasi selengkapnya, lihat Mengubah objek.	18 Maret 2021
AWS PrivateLink	Dengan AWS PrivateLink Amazon S3, Anda dapat terhubung langsung ke S3 dengan menggunakan titik akhir antarmuka di cloud pribadi virtual (VPC) Anda alih-alih terhubung melalui internet. Titik akhir antarmuka dapat diakses langsung dari aplikasi yang ada di on-premise atau di Wilayah AWS yang berbeda. Untuk informasi selengkapnya lihat, AWS PrivateLink untuk Amazon S3.	2 Februari 2021

Mengelola Amazon S3 pada kapasitas Outposts dengan AWS CloudTrail	S3 pada acara manajemen Outposts tersedia CloudTrail melalui log. Untuk informasi selengkapnya, lihat Mengelola S3 pada kapasitas Outposts dengan. CloudTrail	21 Desember 2020
Konsistensi kuat	Amazon S3 memberikan read-after-write konsistensi yang kuat untuk PUT dan DELETE permintaan objek di bucket S3 Anda secara keseluruhan. Wilayah AWS Selain itu, operasi baca di Amazon S3 Select, daftar kontrol akses Amazon S3, Amazon S3 Object Tag, dan metadata objek (misalnya, objek HEAD) sangat konsisten. Untuk informasi selengkapnya, lihat Model konsistensi data Amazon S3.	1 Desember 2020

Sinkronisasi modifikasi replika Amazon S3	Sinkronisasi modifikasi replika Amazon S3 membuat metadata objek, seperti tag, dan pengaturan Kunci Objek ACLs, sinkron antara objek sumber dan replika. Saat fitur ini diaktifkan, Amazon S3 mereplikasi perubahan metadata yang dibuat ke objek sumber atau salinan replika. Untuk informasi selengkapnya, lihat Mereplikasi perubahan metadata dengan sinkronisasi modifikasi replika.	1 Desember 2020
Kunci Bucket Amazon S3	Kunci Bucket Amazon S3 mengurangi biaya enkripsi di sisi server Amazon S3 dengan (SSE-KMS) AWS Key Management Service . Kunci level bucket baru untuk enkripsi di sisi server ini dapat mengurangi biaya permintaan AWS KMS hingga 99 persen dengan mengurangi lalu lintas permintaan dari Amazon S3 ke AWS KMS. Untuk informasi selengkapnya, lihat Mengurangi biaya SSE-KMS menggunakan Kunci Bucket S3.	1 Desember 2020

Lensa Penyimpanan Amazon S3	Lensa Penyimpanan S3 menggabungkan metrik Anda dan menampilkan informasi di bagian Snapshot akun di halaman Bucket konsol Amazon S3. Lensa Penyimpanan S3 juga menyediakan dasbor interaktif yang dapat Anda gunakan untuk memvisualisasikan wawasan dan tren, menandai outlier, serta menerima rekomendasi untuk mengoptimalkan biaya penyimpanan dan menerapkan praktik terbaik perlindungan data. Dasbor Anda memiliki opsi perincian untuk menghasilkan dan memvisualisasikan wawasan di tingkat organisasi, akun, Wilayah AWS, kelas penyimpanan, bucket, prefiks, atau grup Lensa Penyimpanan. Anda juga dapat mengirimkan ekspor metrik harian dalam format CSV atau Parquet ke bucket S3. Untuk informasi selengkapnya, lihat Menilai aktivitas penyimpanan dan penggunaan Anda dengan Lensa Penyimpanan S3.	18 November 2020

Menelusuri permintaan S3 menggunakan AWS X-Ray	Amazon S3 terintegrasi dengan X-Ray untuk menyebarkan konteks pelacakan dan memberi Anda satu rantai permintaan dengan simpul hulu dan hilir. Untuk informasi selengkapnya, lihat Melacak permintaan menggunakan X-Ray.	16 November 2020
Metrik Replikasi S3	Metrik Replikasi S3 memberikan metrik terperinci untuk aturan replikasi dalam konfigurasi replikasi Anda. Untuk informasi selengkapnya, lihat Metrik replikasi dan pemberitahuan peristiwa Amazon S3.	9 November 2020
S3 Intelligent-Tiering Archive Access dan Deep Archive Access	S3 Intelligent-Tiering Archive Access dan Deep Archive Access adalah tingkat penyimpanan tambahan dalam S3 Intelligent-Tiering. Untuk informasi selengkapnya, lihat Kelas penyimpanan untuk secara otomatis mengoptimalkan objek yang sering dan jarang diakses .	9 November 2020

Replikasi penanda hapus	Dengan replikasi penanda hapus, Anda dapat memastikan bahwa penanda hapus disalin ke bucket tujuan untuk aturan replikasi Anda. Untuk informasi selengkapnya, lihat Menggunakan replikasi penanda hapus.	9 November 2020
Kepemilikan Objek S3	Kepemilikan Objek adalah pengaturan bucket S3 yang dapat Anda gunakan untuk mengendalikan kepemilikan objek baru yang diunggah ke bucket Anda. Untuk informasi selegnkapnya, lihat Menggunakan Kepemilikan Objek S3.	2 Oktober 2020
Amazon S3 di Outposts	Dengan Amazon S3 di Outposts, Anda dapat membuat bucket S3 pada AWS Outposts sumber daya Anda dan dengan mudah menyimpan serta mengambil objek lokal untuk aplikasi yang memerlukan akses data lokal, pemrosesan data lokal, dan residensi data. Anda dapat menggunakan S3 di Outposts melalui AWS Management Console AWS CLI,, AWS SDKs, atau REST API. Untuk informasi selengkapnya, lihat Penggunaan Amazon S3 on Outposts.	30 September 2020

Kondisi pemilik bucket	Anda dapat menggunakan kondisi pemilik bucket Amazon S3 untuk memastikan bahwa bucket yang Anda gunakan dalam operasi S3 milik yang Anda harapkan. Akun AWS Untuk informasi selengkapnya, lihat Kondisi pemilik bucket.	11 September 2020
Dukungan Operasi Batch S3 untuk Retensi Kunci Objek	Sekarang, Anda dapat menggunakan Operasi Batch dengan Kunci Objek S3 untuk menerapkan pengaturan penyimpanan pada banyak objek Amazon S3 sekaligus . Untuk informasi selengkap nya, lihat Menetapkan tanggal Retensi Kunci Objek S3 dengan Operasi Batch S3.	4 Mei 2020
Dukungan Operasi Batch S3 untuk Penahanan Hukum Kunci Objek	Sekarang Anda dapat menggunakan Operasi Batch dengan Kunci Objek S3 untuk menambahkan penahanan hukum ke banyak objek Amazon S3 sekaligus. Untuk informasi selengkapnya, lihat Menggunakan Operasi Batch S3 untuk mengatur Penahanan Hukum Kunci Objek S3.	4 Mei 2020

Tag tugas untuk Operasi Batch S3	Anda dapat menambahkan tag ke tugas Operasi Batch S3 Anda untuk mengontrol dan memberi label tugas tersebut. Untuk informasi selengkapnya, lihat Tag untuk tugas Operasi Batch S3.	16 Maret 2020
Titik akses Amazon S3	Titik akses Amazon S3 menyederhanakan pengelola an akses data dalam skala besar untuk set data bersama di S3. Titik akses diberi nama titik akhir jaringan yang melekat pada bucket yang dapat Anda gunakan untuk melakukan operasi objek S3. Untuk informasi selengkapnya, lihat Mengelola akses data dengan titik akses Amazon S3.	2 Desember 2019
Access Analyzer untuk Amazon S3	Access Analyzer untuk Amazon S3 memberi tahu Anda tentang bucket S3 yang dikonfigurasi untuk memungkinkan akses ke siapa pun di internet atau Akun AWS lainnya, termasuk akun di luar organisasi Anda. Untuk informasi selengkapnya, lihat Menggunakan Access Analyzer untuk Amazon S3.	2 Desember 2019

Kontrol Waktu Replikasi S3 (S3 RTC)	Kontrol Waktu Replikasi S3 (S3 RTC) mereplikasi sebagian besar objek yang Anda unggah ke Amazon S3 dalam hitungan detik, dan 99,99 persen dari objek tersebut dalam waktu 15 menit. Untuk informasi selengkapnya, lihat Mereplikasi Objek menggunakan Kontrol Waktu Replikasi S3 (S3 RTC).	20 November 2019
Replikasi Wilayah yang Sama	Replikasi Wilayah yang Sama (SRR) digunakan untuk menyalin objek di seluruh bucket Amazon S3 dalam Wilayah AWS yang sama. Untuk informasi tentang Replikasi Lintas-Wilayah (CRR) dan Replikasi Wilayah yang Sama, lihat Replikasi.	18 September 2019
Dukungan Replikasi Lintas-Wilayah untuk Kunci Objek S3	Replikasi Lintas-Wilayah sekarang mendukung Kunci Objek. Untuk informasi selengkapnya, lihat Apa yang Direplikasi Amazon S3?.	28 Mei 2019

Operasi Batch S3	Dengan menggunakan Operasi Batch S3, Anda dapat melakukan Operasi Batch berskala besar pada objek Amazon S3. Operasi Batch S3 dapat menjalankan satu operasi pada daftar objek yang Anda tentukan. Satu tugas dapat melakukan operasi tertentu pada miliaran objek yang berisi data sebesar exabyte. Untuk informasi selengkapnya, lihat Melakukan Operasi Batch S3.	30 April 2019
Wilayah Asia Pasifik (Hong Kong)	Amazon S3 kini tersedia di Wilayah Asia Pasifik (Hong Kong). Untuk informasi selengkapnya tentang Wilayah dan titik akhir Amazon S3, lihat Wilayah dan titik akhir di Referensi Umum AWS.	24 April 2019
Menambahkan bidang baru ke log akses server	Amazon S3 menambahkan bidang baru berikut ke log akses server: versi Keamanan Lapisan Pengangkutan (TLS). Untuk informasi selengkapnya, lihat Format log akses server.	28 Maret 2019

Kelas penyimpanan arsip baru	Amazon S3 kini menawarkan kelas penyimpanan arsip baru, S3 Glacier Deep Archive (`DEEP_ARCHIVE`), untuk menyimpan objek yang jarang diakses. Untuk informasi selengkapnya, lihat Kelas Penyimpanan.	27 Maret 2019
Menambahkan bidang baru ke log akses server	Amazon S3 menambahkan bidang baru berikut ke log akses server: Host Id, Signature Version, Cipher Suite, Authentication Type, dan Host Header. Untuk informasi selengkapnya, lihat Format log akses server.	5 Maret 2019
Dukungan untuk berkas Inventaris Amazon S3 yang diformat oleh Parquet	Amazon S3 sekarang mendukung format Apache Parquet (Parquet) selain format file kolom baris yang dioptimalkan (ORC) Apache dan nilai yang dipisahkan koma (CSV) untuk file output persediaan. Untuk informasi selengkapnya, lihat Inventaris.	4 Desember 2018
Kunci Objek S3	Amazon S3 kini menawarkan fungsionalitas Kunci Objek yang menyediakan perlindungan Tulis Sekali Baca Banyak (WORM) untuk objek Amazon S3. Untuk informasi selengkapnya, lihat Mengunci Objek.	26 November 2018

Peningkatan kecepatan pemulihan	Dengan peningkatan kecepatan pemulihan Amazon S3, Anda dapat mengubah kecepatan pemulihan dari kelas penyimpanan S3 Glacier Flexible Retrieval menjadi lebih cepat saat pemulihan sedang berlangsung. Untuk informasi selengkapnya, lihat Memulihkan Objek yang Diarsipkan.	26 November 2018
Notifkasi Peristiwa Pemulihan	Notifikasi Peristiwa Amazon S3 kini mendukung peristiwa inisiasi dan penyelesaian saat memulihkan objek dari kelas penyimpanan S3 Glacier Flexible Retrieval. Untuk informasi selengkapnya, lihat Notifikasi Peristiwa.	26 November 2018

PUT langsung ke kelas penyimpanan S3 Glacier Flexible Retrieval	Operasi PUT Amazon S3 kini mendukung penentuan S3 Glacier Flexible Retrieval sebagai kelas penyimpanan saat membuat objek. Sebelumnya, Anda harus melakukan transisi objek ke kelas penyimpanan S3 Glacier Flexible Retrieval dari kelas penyimpanan Amazon S3 lainnya. Selain itu, saat menggunakan Replikasi Lintas-Wilayah (CRR) S3, Anda sekarang dapat menentukan S3 Glacier Flexible Retrieval sebagai kelas penyimpanan untuk objek yang direplikasi. Untuk informasi selengkapnya tentang kelas penyimpanan S3 Glacier Flexible Retrieval, lihat Kelas Penyimpanan. Untuk informasi selengkapnya tentang menentukan kelas penyimpanan untuk objek yang direplikasi, Ikhtisar Konfigurasi Replikasi. Untuk informasi selengkapnya tentang PUT langsung ke perubahan API REST S3 Glacier Flexible Retrieval, lihat Riwayat Dokumen: PUT langsung ke S3 Glacier Flexible Retrieval.	26 November 2018

Kelas penyimpanan baru	Amazon S3 kini menawarkan kelas penyimpanan baru bernama S3 Intelligent-Tiering (`INTELLIGENT_TIERING`) yang dirancang untuk data jangka panjang dengan pola akses yang berubah atau tidak diketahui. Untuk informasi selengkapnya, lihat Kelas Penyimpanan.	26 November 2018
Blokir Akses Publik Amazon S3	Amazon S3 kini menyertakan kemampuan untuk memblokir akses publik ke bucket dan objek dengan basis per bucket atau per akun. Untuk informasi selengkapnya, lihat Menggunakan Blokir Akses Publik Amazon S3.	15 November 2018
Peningkatan pemfilteran dalam aturan Replikasi Lintas-Wilayah (CRR)	Dalam konfigurasi aturan CRR, Anda dapat menentukan filter objek untuk memilih subset objek yang akan menerapkan aturan tersebut. Sebelumnya, Anda hanya dapat memfilter pada awalan kunci objek. Dalam rilis ini, Anda dapat memfilter awalan kunci objek, satu atau beberapa tag objek, atau keduanya. Untuk informasi selengkapnya, lihat Pengaturan CRR: Ikhtisar Konfigurasi Replikasi.	19 September 2018

Fitur baru Amazon S3 Select	Amazon S3 Select sekarang mendukung input Apache Parquet, kueri pada objek JSON bersarang, dan dua metrik pemantauan Amazon baru (dan). CloudWatch `SelectScannedBytes` `SelectReturnedBytes`	5 September 2018
Pembaruan kini tersedia melalui RSS	Anda sekarang dapat berlangganan umpan RSS untuk menerima pemberitahuan tentang pembaruan pada Panduan Pengguna Amazon S3.	19 Juni 2018

Pembaruan sebelumnya

Tabel berikut menjelaskan perubahan penting dalam setiap rilis Panduan Pengguna Amazon S3 sebelum 19 Juni 2018.

Perubahan	Deskripsi	Tanggal
Pembaruan contoh kode	Contoh kode diperbarui: • C#—Memperbarui semua contoh untuk menggunakan pola asinkron berbasis tugas. Untuk informasi selengkapnya, lihat Amazon Web Services Asynchronous APIs for .NET di Panduan Pengembang. AWS SDK untuk .NET Contoh kode sekarang sesuai dengan versi 3 dari AWS SDK untuk .NET. • Java—Memperbarui semua contoh untuk menggunakan model pembuat klien. Untuk informasi selengkapnya tentang model pembuat klien, lihat Membuat Klien Layanan.	30 April 2018

Perubahan	Deskripsi	Tanggal
	• PHP—Memperbarui semua contoh untuk menggunakan AWS SDK untuk PHP 3.0. Untuk informasi lebih lanjut tentang AWS SDK untuk PHP 3.0, lihat AWS SDK untuk PHP. • Ruby—Kode contoh yang diperbarui sehingga contoh bekerja dengan versi 3. AWS SDK untuk Ruby	

Amazon S3 sekarang melaporkan S3 Glacier Flexible Retrieval dan `ONEZONE_IA` kelas penyimpanan ke metrik penyimpanan Amazon Logs CloudWatch	Selain melaporkan byte aktual, metrik penyimpanan ini mencakup byte overhead per objek untuk kelas penyimpanan yang berlaku (`ONEZONE_IA` , `STANDARD_IA` , dan S3 Glacier Flexible Retrieval):
	• Untuk objek kelas penyimpanan `ONEZONE_IA` dan `STANDARD_IA` , Amazon S3 melaporkan objek lebih kecil dari 128 KB sebagai 128 KB. Untuk informasi selengkapnya, lihat Memahami dan mengelola kelas penyimpanan Amazon S3.
	• Untuk objek kelas penyimpanan S3 Glacier Flexible Retrieval, metrik penyimpanan melaporkan overhead berikut:
	  • Overhead 32 KB per objek, dikenakan harga kelas penyimpanan S3 Glacier Flexible Retrieval
	  • Overhead 8 KB per objek, dikenakan harga kelas penyimpanan `STANDARD`
	  Untuk informasi selengkapnya, lihat Transisi objek menggunakan Siklus Hidup Amazon S3.
	Untuk informasi selengkapnya tentang metrik penyimpanan, lihat Memantau metrik dengan Amazon CloudWatch.
		30 April 2018

Perubahan	Deskripsi	Tanggal
Kelas penyimpanan baru	Amazon S3 sekarang menawarkan kelas penyimpanan baru, `STANDARD_IA` (IA, untuk akses yang jarang) guna menyimpan objek. Kelas penyimpanan ini dioptimalkan untuk data yang tahan lama dan tidak banyak diakses. Untuk informasi selengkapnya, lihat Memahami dan mengelola kelas penyimpanan Amazon S3.	4 April 2018
Amazon S3 Select	A> Amazon S3 sekarang mendukung pengambilan konten objek berdasarkan ekspresi SQL. Untuk informasi selengkapnya, lihat Kueri data di tempat dengan Amazon S3 Select.	4 April 2018
Wilayah Asia Pasifik (Osaka-Lokal)	Amazon S3 kini tersedia di Wilayah Asia Pasifik (Osaka-Lokal). Untuk informasi selengkapnya tentang Wilayah dan titik akhir Amazon S3, lihat Wilayah dan titik akhir di Referensi Umum AWS. **Important** Anda dapat menggunakan Wilayah Asia Pasifik (Osaka-Lokal) hanya dalam hubungannya dengan Wilayah Asia Pasifik (Tokyo). Untuk meminta akses ke Wilayah Asia Pasifik (Osaka-Lokal), hubungi perwakilan penjualan Anda.	12 Februari 2018
Stempel waktu pembuatan Inventaris Amazon S3	Inventaris Amazon S3 kini mencakup stempel waktu tanggal dan waktu mulai pembuatan laporan Inventaris Amazon S3. Anda dapat menggunakan stempel waktu untuk menentukan perubahan dalam penyimpanan Amazon S3 Anda sejak waktu mulai pembuatan laporan inventaris.	16 Januari 2018

Perubahan	Deskripsi	Tanggal
Wilayah Eropa (Paris)	Amazon S3 kini tersedia di Wilayah Eropa (Paris). Untuk informasi selengkapnya tentang Wilayah dan titik akhir Amazon S3, lihat Wilayah dan titik akhir di Referensi Umum AWS.	18 Desember 2017
Wilayah Tiongkok (Ningxia)	Amazon S3 kini tersedia di Wilayah Tiongkok (Ningxia). Untuk informasi selengkapnya tentang Wilayah dan titik akhir Amazon S3, lihat Wilayah dan titik akhir di Referensi Umum AWS.	29 November 2017
Dukungan untuk berkas Inventaris Amazon S3 yang diformat oleh ORC	Amazon S3 kini mendukung format kolom baris yang dioptimalkan (ORC) Apache selain format file nilai yang dipisahkan koma (CSV) untuk file output inventaris. Selain itu, Anda kini dapat melakukan kueri Inventaris Amazon S3 menggunakan SQL standar dengan menggunakan Amazon Athena, Amazon Redshift Spectrum, dan alat lainnya seperti Prasto, Apache Hive, dan Apache Spark. Untuk informasi selengkapnya, lihat Membuat katalog dan menganalisis data Anda dengan S3 Inventory.	17 November 2017
Enkripsi default untuk bucket S3	Enkripsi default Amazon S3 menyediakan cara untuk menetapkan perilaku enkripsi default untuk bucket S3. Anda dapat menetapkan enkripsi default pada bucket sehingga semua objek dienkripsi saat mereka disimpan dalam bucket. Objek dienkripsi menggunakan enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3) atau kunci terkelola (SSE-KMS). AWS Untuk informasi selengkapnya, lihat Mengatur perilaku enkripsi sisi server default untuk bucket Amazon S3.	6 November 2017

Perubahan	Deskripsi	Tanggal
Status enkripsi dalam Inventaris Amazon S3	Amazon S3 sekarang mendukung penyertaan status enkripsi dalam Inventaris Amazon S3 sehingga Anda dapat melihat bagaimana objek Anda dienkripsi saat diam untuk audit kepatuhan atau tujuan lain. Anda juga dapat mengonfigurasi untuk mengenkripsi Inventaris Amazon S3 dengan enkripsi di sisi server (SSE) atau SSE-KMS sehingga semua file inventaris dienkripsi dengan semestiny a. Untuk informasi selengkapnya, lihat Membuat katalog dan menganalisis data Anda dengan S3 Inventory.	6 November 2017
Peningkatan Replikasi Lintas-Wilayah (CRR)	Replikasi Lintas-Wilayah sekarang mendukung hal berikut: • Dalam skenario lintas akun, Anda dapat menambahkan konfigurasi CRR untuk mengubah kepemilikan replika ke Akun AWS yang menjadi pemilik bucket tujuan. Untuk informasi selengkapnya, lihat Mengubah pemilik replika. • Secara default, Amazon S3 tidak mereplikasi objek di bucket sumber Anda yang dibuat menggunakan enkripsi sisi server menggunakan kunci yang disimpan dalam AWS KMS Dalam konfigurasi CRR Anda, Anda sekarang dapat mengarahkan Amazon S3 untuk mereplikasi objek ini. Untuk informasi selengkapnya, lihat Mereplikasi objek terenkripsi (SSE-S3, SSE-KMS, DSSE-KMS, SSE-C).	6 November 2017
Wilayah Eropa (London)	Amazon S3 kini tersedia di Wilayah Eropa (London). Untuk informasi selengkapnya tentang Wilayah dan titik akhir Amazon S3, lihat Wilayah dan titik akhir di Referensi Umum AWS.	13 Desember 2016
Wilayah Kanada (Pusat)	Amazon S3 kini tersedia di Wilayah Kanada (Pusat). Untuk informasi selengkapnya tentang Wilayah dan titik akhir Amazon S3, lihat Wilayah dan titik akhir di Referensi Umum AWS.	8 Desember 2016

Perubahan	Deskripsi	Tanggal
Penandaan objek	Amazon S3 kini mendukung penandaan objek. Penandaan objek memungkinkan Anda mengategorikan penyimpanan. Awalan nama kunci objek juga memungkinkan Anda untuk mengategorikan penyimpanan, penandaan objek menambah dimensi lain. Ada penawaran penandaan keuntungan tambahan. Ini termasuk: • Tag objek memungkinkan kontrol akses izin terperinci (misalnya, Anda dapat memberikan izin pengguna IAM ke objek hanya baca dengan tag tertentu). • Kontrol terperinci dalam menentukan konfigurasi siklus hidup. Anda dapat menetapkan tag untuk memilih subset objek yang menerapkan aturan siklus hidup. • Jika Replikasi Lintas-Wilayah (CRR) telah dikonfigurasi, Amazon S3 dapat mereplikasi tag. Anda harus memberikan izin yang diperlukan kepada peran IAM yang dibuat untuk Amazon S3 guna mengasumsikan objek yang direplikasi atas nama Anda. • Anda juga dapat menyesuaikan CloudWatch metrik dan CloudTrail peristiwa untuk menampilkan informasi berdasarkan filter tag tertentu. Untuk informasi selengkapnya, lihat Mengategorikan penyimpanan Anda menggunakan tag.	29 November 2016

Perubahan	Deskripsi	Tanggal
Siklus Hidup Amazon S3 kini mendukung filter berbasis tag	Amazon S3 kini mendukung pemfilteran berbasis tag dalam konfigurasi siklus hidup. Anda sekarang dapat menentukan aturan siklus hidup, yaitu Anda bisa menentukan awalan kunci, satu atau beberapa tag objek, atau kombinasi keduanya untuk memilih subset objek yang menerapkan aturan siklus hidup. Untuk informasi selengkapnya, lihat Mengelola siklus hidup objek.	29 November 2016
CloudWatch minta metrik untuk bucket	Amazon S3 sekarang mendukung CloudWatch metrik untuk permintaan yang dibuat di bucket. Saat Anda mengaktifkan metrik ini untuk satu bucket, metrik melaporkan pada interval 1 menit. Anda juga dapat mengonfigurasi objek mana dalam bucket yang akan melaporkan metrik permintaan ini. Untuk informasi selengkapnya, lihat Memantau metrik dengan Amazon CloudWatch.	29 November 2016
Inventaris Amazon S3	Amazon S3 kini mendukung inventaris penyimpanan. Inventaris Amazon S3 menyediakan output berkas datar dari objek Anda dan metadata yang sesuai setiap hari atau setiap minggu untuk bucket S3 atau awalan bersama (yaitu, objek yang memiliki nama yang dimulai dengan string umum). Untuk informasi selengkapnya, lihat Membuat katalog dan menganalisis data Anda dengan S3 Inventory.	29 November 2016

Perubahan	Deskripsi	Tanggal
Analitik Amazon S3–Analisis Kelas Penyimpanan	Analitik Amazon S3 yang baru–fitur analisis kelas penyimpanan mengamati pola akses data untuk membantu Anda menentukan kapan mengalihkan penyimpanan `STANDARD` yang jarang diakses ke kelas penyimpanan `STANDARD_IA` (IA, untuk akses yang jarang). Setelah analisis kelas penyimpanan mengamati pola akses yang jarang dari kumpulan data yang difilter selama periode waktu tertentu, Anda dapat menggunakan hasil analisis untuk membantu Anda meningkatkan konfigurasi siklus hidup. Fitur ini juga mencakup analisis terperinci harian mengenai penggunaan penyimpanan Anda di tingkat bucket, awalan, atau tag tertentu yang dapat Anda ekspor ke bucket S3.	29 November 2016
Pengambilan data baru yang Dipercepat dan Massal saat memulihkan objek yang diarsipkan dari S3 Glacier	Amazon S3 kini mendukung pengambilan data yang Dipercepat dan Massal di samping pengambilan Standar saat memulihkan objek yang diarsipkan ke S3 Glacier. Untuk informasi selengkapnya, lihat Memulihkan objek yang diarsipkan.	21 November 2016
CloudTrail pencatatan objek	CloudTrail mendukung pencatatan operasi API tingkat objek Amazon S3 seperti`GetObject` ,`PutObject` , dan. `DeleteObject` Anda dapat mengonfigurasi penyeleksi peristiwa Anda untuk log operasi API tingkat objek. Untuk informasi selengkapnya, lihat Mencatat panggilan API Amazon S3 menggunakan AWS CloudTrail.	21 November 2016
Wilayah AS Timur (Ohio)	Amazon S3 kini tersedia di Wilayah AS Timur (Ohio). Untuk informasi selengkapnya tentang Wilayah dan titik akhir Amazon S3, lihat Wilayah dan titik akhir di Referensi Umum AWS.	17 Oktober 2016

Perubahan	Deskripsi	Tanggal
IPv6 dukungan untuk Amazon S3 Transfer Acceleration	Amazon S3 sekarang mendukung Internet Protocol versi 6 (IPv6) untuk Amazon S3 Transfer Acceleration. Anda dapat terhubung ke Amazon S3 IPv6 melalui menggunak an dual-stack baru untuk titik akhir Transfer Acceleration. Untuk informasi selengkapnya, lihat Memulai Amazon S3 Transfer Acceleration.	6 Oktober 2016
IPv6 dukungan	Amazon S3 sekarang mendukung Internet Protocol versi 6 ()IPv6. Anda dapat mengakses Amazon S3 melalui IPv6 dengan menggunakan titik akhir dual-stack. Untuk informasi selengkapnya, lihat Membuat permintaan ke Amazon S3 IPv6 di Referensi API Amazon S3.	11 Agustus, 2016
Wilayah Asia Pasifik (Mumbai)	Amazon S3 kini tersedia di Wilayah Asia Pasifik (Mumbai). Untuk informasi selengkapnya tentang Wilayah dan titik akhir Amazon S3, lihat Wilayah dan titik akhir di Referensi Umum AWS.	27 Juni 2016
Amazon S3 Transfer Acceleration	Amazon S3 Transfer Acceleration memungkinkan transfer berkas yang cepat, mudah, dan aman melalui jarak jauh antara klien Anda dan bucket S3. Transfer Acceleration memanfaatkan lokasi edge Amazon yang didistribusikan CloudFront secara global. Untuk informasi selengkapnya, lihat Mengonfigurasi transfer file yang cepat dan aman menggunakan Amazon S3 Transfer Acceleration.	19 April 2016
Dukungan siklus hidup untuk menghapus penanda hapus objek kedaluwarsa	Tindakan `Expiration` konfigurasi siklus hidup kini memungkinkan Anda mengarahkan Amazon S3 untuk menghapus penanda hapus objek yang kedaluwarsa di bucket berversi. Untuk informasi selengkapnya, lihat Elemen untuk mendeskripsikan tindakan siklus hidup.	16 Maret 2016

Perubahan	Deskripsi	Tanggal
Konfigurasi siklus hidup bucket kini mendukung tindakan untuk menghenti kan pengunggahan multibagian yang tidak lengkap	Konfigurasi siklus hidup bucket kini mendukung tindakan `AbortIncompleteMultipartUpload` yang dapat Anda gunakan untuk mengarahkan Amazon S3 untuk menghentikan pengunggahan multibagian yang tidak selesai dalam jumlah hari yang ditentukan setelah diinisias i. Saat pengunggahan multibagian memenuhi syarat untuk operasi penghentian, Amazon S3 menghapus setiap bagian yang diunggah dan menghentikan pengunggahan multibagian. Untuk informasi konseptual, lihat topik berikut di Panduan Pengguna Amazon S3: • Membatalkan unggahan multibagian • Elemen untuk mendeskripsikan tindakan siklus hidup Operasi API berikut telah diperbarui untuk mendukung tindakan baru: • PUT Bucket lifecycle–Konfigurasi XML sekarang memungkinkan Anda menentukan tindakan `AbortInco mpleteMultipartUpload` dalam aturan konfigura si siklus hidup. • List Parts dan Initiate Multipart Upload–Kedua operasi API ini sekarang mengembalikan dua header respons tambahan (`x-amz-abort-date` , dan `x-amz-abort-rule-id`) jika bucket memiliki aturan siklus hidup yang menentukan tindakan `AbortInco mpleteMultipartUpload` . Header dalam respons ini menunjukkan saat unggahan multibagian yang dimulai memenuhi syarat untuk operasi penghentian dan aturan siklus hidup mana yang berlaku.	16 Maret 2016

Perubahan	Deskripsi	Tanggal
Wilayah Asia Pasifik (Seoul)	Amazon S3 kini tersedia di Wilayah Asia Pasifik (Seoul). Untuk informasi selengkapnya tentang Wilayah dan titik akhir Amazon S3, lihat Wilayah dan titik akhir di Referensi Umum AWS.	6 Januari 2016
Kunci kondisi baru dan perubahan pengunggahan multibagian	Kebijakan IAM kini mendukung kunci kondisi `s3:x-amz-storage-class` Amazon S3. Untuk informasi selengkapnya, lihat Contoh kebijakan bucket menggunakan tombol kondisi. Anda tidak perlu lagi menjadi inisiator pengunggahan multibagian untuk mengunggah bagian dan menyelesaikan pengunggahan. Untuk informasi selengkapnya, lihat API dan izin unggahan multibagian.	14 Desember 2015
Mengganti nama Wilayah Standar AS	Mengubah string nama Wilayah dari "Standar AS” menjadi "AS Timur (Virginia Utara).” Ini hanya pembaruan nama Wilayah, tidak ada perubahan dalam fungsionalitas.	11 Desember 2015

Perubahan	Deskripsi	Tanggal
Kelas penyimpanan baru	Amazon S3 sekarang menawarkan kelas penyimpanan baru, STANDARD_IA (IA, untuk akses yang jarang) guna menyimpan objek. Kelas penyimpanan ini dioptimalkan untuk data yang tahan lama dan tidak banyak diakses. Untuk informasi selengkapnya, lihat Memahami dan mengelola kelas penyimpanan Amazon S3. Pembaruan fitur konfigurasi siklus hidup sekarang memungkinkan Anda melakukan transisi objek ke kelas penyimpanan STANDARD_IA. Untuk informasi selengkap nya, lihat Mengelola siklus hidup objek. Sebelumnya, fitur Replikasi Lintas-Wilayah menggunak an kelas penyimpanan objek sumber untuk replika objek. Sekarang, saat Anda mengonfigurasi Replikasi Lintas-Wilayah, Anda dapat menentukan kelas penyimpanan untuk replika objek yang dibuat dalam bucket tujuan. Untuk informasi selengkapnya, lihat Mereplikasi objek di dalam dan di seluruh Wilayah.	16 September 2015
AWS CloudTrail integrasi	AWS CloudTrail Integrasi baru memungkinkan Anda merekam aktivitas API Amazon S3 di bucket S3 Anda. Anda dapat menggunakan CloudTrail untuk melacak pembuatan atau penghapusan bucket S3, modifikasi kontrol akses, atau perubahan konfigurasi siklus hidup. Untuk informasi selengkapnya, lihat Mencatat panggilan API Amazon S3 menggunakan AWS CloudTrail.	1 September 2015

Perubahan	Deskripsi	Tanggal
Peningkatan batas bucket	Amazon S3 kini mendukung peningkatan batas bucket. Secara default, pelanggan dapat membuat hingga 100 ember di dalamnya. Akun AWS Pelanggan yang memerlukan bucket tambahan dapat meningkatkan batas tersebut dengan mengajukan peningkatan batas layanan. Untuk informasi tentang cara meningkatkan batas bucket Anda, buka Layanan AWS kuota di AWS Referensi Umum. Untuk informasi selengkapnya, lihat Menggunakan AWS SDKs dan Kuota, batasan, dan batasan bucket tujuan umum.	4 Agustus 2015
Pembaruan model konsistensi	Amazon S3 sekarang mendukung read-after-write konsistensi untuk objek baru yang ditambahkan ke Amazon S3 di Wilayah AS Timur (Virginia N.). Sebelum pembaruan ini, semua Wilayah kecuali Wilayah AS Timur (Virginia N.) mendukung read-after-write konsistensi untuk objek baru yang diunggah ke Amazon S3. Dengan peningkat an ini, Amazon S3 sekarang read-after-write mendukung konsistensi di semua Wilayah untuk objek baru yang ditambahkan ke Amazon S3. Read-after-writekonsistensi memungkinkan Anda untuk mengambil objek segera setelah pembuatan di Amazon S3. Untuk informasi selengkapnya, lihat Wilayah.	4 Agustus 2015
Pemberitahuan peristiwa	Notifikasi Peristiwa Amazon S3 telah diperbarui untuk menambahkan pemberitahuan saat objek dihapus dan untuk menambahkan pemfilteran pada nama objek dengan awalan dan suffix yang cocok. Untuk informasi selengkap nya, lihat Notifikasi Peristiwa Amazon S3.	28 Juli 2015

Perubahan	Deskripsi	Tanggal
CloudWatch Integrasi Amazon	CloudWatch Integrasi Amazon baru memungkinkan Anda memantau dan mengatur alarm pada penggunaan Amazon S3 Anda CloudWatch melalui metrik untuk Amazon S3. Metrik yang didukung meliputi total byte penyimpanan Standar, byte total untuk Reduced-Redundancy Storage, dan jumlah total objek untuk bucket S3 tertentu. Untuk informasi selengkapnya, lihat Memantau metrik dengan Amazon CloudWatch.	28 Juli 2015
Dukungan untuk menghapus dan mengosongkan bucket berisi	Amazon S3 kini mendukung penghapusan dan pengosong an bucket berisi. Untuk informasi selengkapnya, lihat Mengosongkan ember tujuan umum.	16 Juli 2015
Kebijakan bucket untuk titik akhir VPC Amazon	Amazon S3 telah menambahkan dukungan kebijakan bucket untuk titik akhir cloud privat virtual (VPC). Anda dapat menggunakan kebijakan bucket S3 untuk mengontro l akses ke bucket dari titik akhir VPC tertentu, atau spesifik. VPCs Titik akhir VPC mudah dikonfigurasi, sangat andal, dan menyediakan koneksi aman ke Amazon S3 tanpa memerlukan gateway atau instans NAT. Untuk informasi selengkapnya, lihat Megendalikan akses dari titik akhir VPC dengan kebijakan bucket.	29 April 2015
Pemberitahuan peristiwa	Pemberitahuan Acara Amazon S3 telah diperbarui untuk mendukung peralihan ke izin berbasis sumber daya untuk fungsi. AWS Lambda Untuk informasi selengkapnya, lihat Notifikasi Peristiwa Amazon S3.	9 April 2015
Replikasi Lintas-Wi layah	Amazon S3 kini mendukung Replikasi Lintas-Wilayah. Replikasi Lintas Wilayah adalah penyalinan objek secara otomatis dan asinkron di seluruh ember dengan cara yang berbeda. Wilayah AWS Untuk informasi selengkapnya, lihat Mereplikasi objek di dalam dan di seluruh Wilayah.	24 Maret 2015

Perubahan	Deskripsi	Tanggal
Pemberitahuan peristiwa	Amazon S3 kini mendukung tipe dan tujuan peristiwa baru dalam konfigurasi pemberitahuan bucket. Sebelum rilis ini, Amazon S3 hanya mendukung s3: jenis ReducedRedundancyLostObject acara dan topik Amazon SNS sebagai tujuan. Untuk informasi selengkapnya tentang tipe peristiwa baru, lihat Notifikasi Peristiwa Amazon S3.	13 November 2014
Enkripsi sisi server dengan kunci enkripsi yang disediakan pelanggan	Enkripsi sisi server dengan kunci AWS Key Management Service (AWS KMS) (SSE-KMS) Amazon S3 sekarang mendukung enkripsi sisi server menggunakan. AWS KMS Fitur ini memungkinkan Anda mengelola kunci amplop AWS KMS, dan AWS KMS panggilan Amazon S3 untuk mengakses kunci amplop dalam izin yang Anda tetapkan. Untuk informasi selengkapnya tentang enkripsi sisi server dengan AWS KMS, lihat Melindungi Data Menggunakan Enkripsi Sisi Server dengan. AWS Key Management Service	12 November 2014
Wilayah Eropa (Frankfurt)	Amazon S3 kini tersedia di Wilayah Eropa (Frankfurt).	23 Oktober 2014

Perubahan	Deskripsi	Tanggal
Enkripsi di sisi server dengan kunci enkripsi yang disediakan pelanggan	Amazon S3 kini mendukung enkripsi di sisi server menggunakan kunci enkripsi yang disediakan pelanggan (SSE-C). Enkripsi di sisi server memungkinkan Anda meminta Amazon S3 mengenkripsi data diam Anda. Saat menggunakan SSE-C, Amazon S3 mengenkripsi objek Anda dengan kunci enkripsi khusus yang Anda sediakan. Karena Amazon S3 melakukan enkripsi untuk Anda, Anda mendapatkan manfaat menggunakan kunci enkripsi Anda sendiri tanpa perlu biaya penulisan atau mengeksekusi kode enkripsi Anda sendiri. Untuk informasi selengkapnya tentang SSE-C, lihat enkripsi di sisi server (Menggunakan Kunci Enkripsi yang Disediakan Pelanggan).	12 Juni 2014
Dukungan siklus hidup untuk Penentuan Versi	Sebelum rilis ini, konfigurasi siklus hidup hanya didukung pada bucket tanpa versi. Sekarang Anda dapat mengonfig urasi siklus hidup di bucket tanpa versi dan dengan dukungan Penentuan Versi. Untuk informasi selengkapnya, lihat Mengelola siklus hidup objek.	20 Mei 2014
Topik kontrol akses direvisi	Revisi dokumentasi kontrol akses Amazon S3. Untuk informasi selengkapnya, lihat Identity and Access Management untuk Amazon S3.	15 April 2014
Topik pencatatan log akses server direvisi	Revisi dokumentasi pencatatan akses server. Untuk informasi selengkapnya, lihat Pencatatan permintaan dengan pencatatan akses server.	26 November 2013
Sampel .NET SDK diperbarui ke versi 2.0	Sampel .NET SDK dalam panduan ini sekarang sesuai dengan versi 2.0.	26 November 2013

Perubahan	Deskripsi	Tanggal
Dukungan SOAP Melalui HTTP Tidak Berlaku Lagi	Dukungan SOAP melalui HTTP dihilangkan, tetapi masih tersedia melalui HTTPS. Fitur Amazon S3 baru tidak mendukung SOAP. Kami menyarankan Anda menggunakan REST API atau AWS SDKs.	20 September 2013
Dukungan variabel kebijakan IAM	Bahasa kebijakan IAM kini mendukung variabel. Ketika suatu kebijakan dievaluasi, setiap variabel kebijakan diganti dengan nilai yang dipasok oleh informasi berbasis konteks dari sesi pengguna yang diautentikasi. Anda dapat menggunakan variabel kebijakan untuk menentukan kebijakan tujuan umum tanpa mencantumkan semua komponen kebijakan secara eksplisit. Untuk informasi selengkapnya tentang variabel kebijakan, lihat Ikhtisar Variabel Kebijakan IAM dalam Panduan Pengguna IAM. Untuk contoh variabel kebijakan di Amazon S3, lihat Contoh kebijakan berbasis identitas untuk Amazon S3.	3 April 2013
Dukungan konsol untuk Pembayaran oleh Pemohon	Sekarang Anda dapat mengonfigurasi bucket untuk Pembayaran oleh Pemohon dengan menggunakan konsol Amazon S3. Untuk informasi selengkapnya, lihat Menggunakan Requester Pays ember tujuan umum untuk transfer dan penggunaan penyimpanan.	31 Desember 2012

Perubahan	Deskripsi	Tanggal
Dukungan domain root untuk hosting situs web	Amazon S3 kini mendukung hosting situs web statis pada domain root. Pengunjung situs web Anda dapat mengakses situs Anda dari peramban mereka tanpa menyebutkan www di alamat web (misalnya, mereka dapat menggunakan example.com, alih-alih www.example.com). Banyak pelanggan sudah melakukan hosting situs web statis di Amazon S3 yang dapat diakses dari subdomain www (misalnya, www.example.com). Sebelumnya, untuk mendukung akses domain root, Anda perlu menjalank an server web Anda sendiri untuk memproksi permintaa n domain root dari peramban ke situs web Anda di Amazon S3. Menjalankan server web ke permintaan proksi menimbulkan biaya tambahan, beban operasion al, dan potensi titik kegagalan lainnya. Sekarang, Anda dapat memanfaatkan ketersediaan dan daya tahan tinggi Amazon S3 untuk alamat domain www dan root. Untuk informasi selengkapnya, lihat Hosting situs web statis menggunakan Amazon S3.	27 Desember 2012
Revisi konsol	Konsol Amazon S3 telah diperbarui. Topik dokumentasi yang mengacu pada konsol telah direvisi sebagaimana mestinya.	14 Desember 2012

Perubahan	Deskripsi	Tanggal
Dukungan untuk Mengarsipkan Data ke S3 Glacier	Amazon S3 kini mendukung opsi penyimpanan yang memungkinkan Anda memanfaatkan layanan penyimpanan berbiaya rendah S3 Glacier untuk pengarsipan data. Untuk mengarsipkan objek, Anda menentukan aturan pengarsip an yang mengidentifikasi objek dan kerangka waktu saat Anda ingin Amazon S3 mengarsipkan objek tersebut ke S3 Glacier. Anda dapat dengan mudah menetapkan aturan pada bucket menggunakan konsol Amazon S3 atau secara terprogram menggunakan Amazon S3 API atau. AWS SDKs Untuk informasi selengkapnya, lihat Mengelola siklus hidup objek.	13 November 2012
Dukungan untuk Pengalihan Halaman Situs Web	Untuk bucket yang dikonfigurasi sebagai situs web, Amazon S3 kini mendukung pengalihan permintaan objek ke objek lain dalam bucket yang sama atau ke URL eksternal. Untuk informasi selengkapnya, lihat (Opsional) Mengonfigurasi pengalihan halaman web. Untuk informasi tentang hosting situs web, lihat Hosting situs web statis menggunakan Amazon S3.	4 Oktober 2012
Dukungan untuk Berbagi Sumber Daya Lintas Asal (CORS)	Amazon S3 kini mendukung Berbagi Sumber Daya Lintas Asal (CORS). CORS mendefinisikan cara aplikasi web klien yang dimuat dalam satu domain yang dapat berintera ksi dengan atau mengakses sumber daya dalam domain yang berbeda. Dengan dukungan CORS di Amazon S3, Anda dapat membangun aplikasi web sisi klien yang kaya di atas Amazon S3 dan secara selektif memungkinkan akses lintas domain ke sumber daya Amazon S3 Anda. Untuk informasi selengkapnya, lihat Berbagi sumber daya lintas asal (CORS).	31 Agustus 2012

Perubahan	Deskripsi	Tanggal
Dukungan untuk Tanda Alokasi Biaya	Amazon S3 kini mendukung penandaan alokasi biaya, yang memungkinkan Anda memberi label pada bucket S3 sehingga Anda dapat melacak biayanya terhadap proyek atau kriteria lainnya dengan lebih mudah. Untuk informasi selengkapnya tentang penandaan untuk bucket, lihat Menggunakan tanda alokasi biaya bucket S3.	21 Agustus 2012
Dukungan untuk akses API yang dilindungi MFA dalam kebijakan bucket	Amazon S3 sekarang mendukung akses API yang dilindungi MFA, fitur yang dapat menerapkan Otentikasi AWS Multi-Faktor untuk tingkat keamanan ekstra saat mengakses sumber daya Amazon S3 Anda. Ini adalah fitur keamanan yang mengharuskan pengguna untuk membuktikan kepemilikan fisik perangkat MFA dengan menyediakan kode MFA yang valid. Untuk informasi selengkapnya, kunjungi Autentikasi Multi-Faktor AWS. Sekarang Anda dapat meminta autentikasi MFA untuk permintaan akses sumber daya Amazon S3. Untuk menerapkan autentikasi MFA, Amazon S3 kini mendukung kunci `aws:MultiFactorAuthAge` dalam kebijakan bucket. Untuk contoh kebijakan bucket, lihat Membutuhkan MFA.	10 Juli 2012
Dukungan Kedaluwarsa Objek	Anda dapat menggunakan Kedaluwarsa Objek untuk menjadwalkan penghapusan data otomatis setelah periode waktu yang dikonfigurasi. Anda mengatur kedaluwarsa objek dengan menambahkan konfigurasi siklus hidup ke bucket.	27 Desember 2011
Wilayah baru yang didukung	Amazon S3 kini mendukung Wilayah Amerika Selatan (Sao Paulo). Untuk informasi selengkapnya, lihat Mengakses bucket tujuan umum Amazon S3.	14 Desember 2011

Perubahan	Deskripsi	Tanggal
Penghapusan Multi-Objek	Amazon S3 kini mendukung API Penghapusan Multi-Objek yang memungkinkan Anda menghapus beberapa objek dalam satu permintaan. Dengan fitur ini, Anda dapat menghapus banyak objek dari Amazon S3 lebih cepat dibandingkan dengan menggunakan beberapa permintaan DELETE individual. Untuk informasi selengkapnya, lihat Menghapus objek Amazon S3.	7 Desember 2011
Wilayah baru yang didukung	Amazon S3 kini mendukung Wilayah AS Barat (Oregon). Untuk informasi selengkapnya, lihat Bucket dan Wilayah.	8 November 2011
Pembaruan Dokumentasi	Perbaikan bug dokumentasi.	8 November 2011
Pembaruan Dokumentasi	Selain perbaikan bug dokumentasi, rilis ini mencakup beberapa peningkatan berikut: • Bagian enkripsi sisi server baru menggunakan AWS SDK untuk PHP dan AWS SDK untuk Ruby (lihat). Menentukan enkripsi di sisi server dengan kunci terkelola Amazon S3 (SSE-S3)	17 Oktober 2011
Dukungan enkripsi di sisi server	Amazon S3 kini mendukung enkripsi di sisi server. Dengan ini, Anda dapat meminta Amazon S3 untuk mengenkripsi data diam, yaitu mengenkripsi data objek Anda ketika Amazon S3 menulis data Anda ke disk di pusat datanya. Selain pembaruan REST API, AWS SDK untuk Java dan .NET menyediakan fungsionalitas yang diperlukan untuk meminta enkripsi sisi server. Anda juga dapat meminta enkripsi di sisi server saat mengunggah objek menggunakan file AWS Management Console. Untuk mempelajari enkripsi data lebih lanjut, kunjungi Menggunakan Enkripsi Data.	4 Oktober 2011

Perubahan	Deskripsi	Tanggal
Pembaruan Dokumentasi	Selain perbaikan bug dokumentasi, rilis ini mencakup beberapa peningkatan berikut: • Menambahkan sampel Ruby dan PHP ke Making request di bagian Referensi API Amazon S3. • Ditambahkan bagian yang menjelaskan bagaimana untuk menghasilkan dan menggunakan URLs presigned . Untuk informasi selengkapnya, lihat Berbagi objek dengan presigned URLs dan Berbagi objek dengan presigned URLs. • Memperbarui bagian yang ada untuk memperkenalkan AWS Explorers for Eclipse dan Visual Studio. Untuk informasi selengkapnya, lihat Mengembangkan dengan Amazon S3 menggunakan Referensi API Amazon S3 AWS SDKs di Amazon S3.	22 September 2011

Perubahan	Deskripsi	Tanggal
Dukungan untuk mengirim permintaan menggunakan kredensial keamanan sementara	Selain menggunakan kredensil keamanan pengguna Anda Akun AWS dan IAM untuk mengirim permintaan yang diautentikasi ke Amazon S3, Anda sekarang dapat mengirim permintaan menggunakan kredensil keamanan sementara yang Anda peroleh dari (IAM). AWS Identity and Access Management Anda dapat menggunakan AWS Security Token Service API atau pustaka pembungkus AWS SDK untuk meminta kredensil sementara ini dari IAM. Anda dapat meminta kredensial keamanan sementara ini untuk penggunaan Anda sendiri atau menyerahkannya kepada pengguna gabungan dan aplikasi. Fitur ini memungkinkan Anda untuk mengelola pengguna Anda di luar AWS dan memberi mereka kredensil keamanan sementara untuk mengakses sumber daya Anda AWS . Untuk informasi selengkapnya, lihat Membuat permintaan di Referensi API Amazon S3. Untuk informasi selengkapnya tentang dukungan IAM untuk kredensial keamanan sementara, lihat Kredensial Keamanan Sementara dalam Panduan Pengguna IAM.	3 Agustus 2011
API Unggahan Multibagian diperluas untuk memungkinkan penyalinan objek hingga 5 TB	Sebelum rilis ini, API Amazon S3 mendukung penyalinan objek dengan ukuran hingga 5 GB. Untuk mengaktifkan penyalinan objek yang lebih besar dari 5 GB, Amazon S3 kini memperluas API unggahan multibagian dengan operasi baru, `Upload Part (Copy)` . Anda dapat menggunakan operasi pengunggahan multibagian ini untuk menyalin objek berukuran hingga 5 TB. Untuk informasi selengkapnya, lihat Menyalin, memindahkan, dan mengganti nama objek. Untuk informasi konseptual tentang API unggahan multibagian, lihat Mengunggah dan menyalin objek menggunakan unggahan multipart di Amazon S3.	21 Juni 2011

Perubahan	Deskripsi	Tanggal
Panggilan API SOAP melalui HTTP dinonaktifkan	Untuk meningkatkan keamanan, panggilan API SOAP melalui HTTP dinonaktifkan. Permintaan SOAP yang diautentikasi dan anonim harus dikirim ke Amazon S3 menggunakan SSL.	6 Juni 2011
IAM memungkinkan delegasi lintas akun	Sebelumnya, untuk mengakses sumber daya Amazon S3, pengguna IAM memerlukan izin dari induk dan pemilik sumber daya Akun AWS Amazon S3. Dengan akses lintas akun, pengguna IAM sekarang hanya perlu izin dari akun pemilik. Artinya, Jika pemilik sumber daya memberikan akses ke Akun AWS, sekarang Akun AWS dapat memberikan pengguna IAM-nya akses ke sumber daya ini. Untuk informasi selengkapnya, lihat Membuat peran untuk mendelegasikan izin ke pengguna IAM dalam Panduan pengguna IAM. Untuk informasi selengkapnya tentang menentukan pengguna utama dalam kebijakan bucket, lihat Prinsipal untuk kebijakan bucket.	6 Juni 2011
Tautan baru	Informasi titik akhir layanan ini sekarang terletak di Referensi Umum AWS . Untuk informasi selengkapnya, buka Wilayah dan titik akhir dalam Referensi Umum AWS.	1 Maret 2011

Perubahan	Deskripsi	Tanggal
Dukungan untuk hosting situs web statis di Amazon S3	Amazon S3 memperkenalkan dukungan yang ditingkatkan untuk hosting situs web statis. Ini termasuk dukungan untuk dokumen indeks dan dokumen kesalahan kustom. Saat menggunakan fitur ini, permintaan ke root bucket atau subfolder (misalnya, `http://mywebsite.com/subfolder`) akan menampilkan dokumen indeks, alih-alih daftar objek di bucket Anda. Jika terjadi kesalahan, Amazon S3 akan menampilkan pesan kesalahan khusus Anda, alih-alih pesan kesalahan Amazon S3. Untuk informasi selengkapnya, lihat Hosting situs web statis menggunakan Amazon S3.	6 Juni 2011
Informasi titik akhir layanan ini sekarang terletak di Referensi Umum AWS . Untuk informasi selengkapnya, buka Wilayah dan titik akhir dalam Referensi Umum AWS.	1 Maret 2011	
Dukungan untuk hosting situs web statis di Amazon S3	Amazon S3 memperkenalkan dukungan yang ditingkatkan untuk hosting situs web statis. Ini termasuk dukungan untuk dokumen indeks dan dokumen kesalahan kustom. Saat menggunakan fitur ini, permintaan ke root bucket atau subfolder (misalnya, `http://mywebsite.com/subfolder`) akan menampilkan dokumen indeks, alih-alih daftar objek di bucket Anda. Jika terjadi kesalahan, Amazon S3 akan menampilkan pesan kesalahan khusus Anda, alih-alih pesan kesalahan Amazon S3. Untuk informasi selengkapnya, lihat Hosting situs web statis menggunakan Amazon S3.	17 Februari 2011

Perubahan	Deskripsi	Tanggal
Dukungan API Header Respons	API REST GET Object kini memungkinkan Anda mengubah header respons permintaan REST GET Object untuk setiap permintaan. Artinya, Anda dapat mengubah metadata objek dalam respons, tanpa mengubah objek itu sendiri. Untuk informasi selengkapnya, lihat Mengunggah objek.	14 Januari 2011
Dukungan objek besar	Amazon S3 telah meningkatkan ukuran maksimum objek yang dapat Anda simpan dalam bucket S3 dari 5 GB menjadi 5 TB. Jika Anda menggunakan API REST, Anda dapat mengunggah objek hingga 5 GB dalam satu operasi PUT. Untuk objek yang lebih besar, Anda harus menggunakan API REST Pengunggahan Multibagian untuk mengunggah objek secara terpisah. Untuk informasi selengkapnya, lihat Mengunggah dan menyalin objek menggunakan unggahan multipart di Amazon S3.	9 Desember 2010
Pengunggahan multibagian	Pengunggahan multibagian memungkinkan unggahan yang lebih cepat dan lebih fleksibel ke Amazon S3. Dengan ini, Anda dapat mengunggah satu objek sebagai sekumpulan bagian. Untuk informasi selengkapnya, lihat Mengunggah dan menyalin objek menggunakan unggahan multipart di Amazon S3.	10 November 2010
Dukungan ID kanonis dalam kebijakan bucket	Anda sekarang dapat menentukan kanonik IDs dalam kebijakan bucket. Untuk informasi selengkapnya, lihat Prinsipal untuk kebijakan bucket	17 September 2010
Amazon S3 bekerja dengan IAM	Layanan ini sekarang terintegrasi dengan AWS Identity and Access Management (IAM). Untuk informasi selengkapnya, kunjungi Layanan AWS yang bekerja dengan IAM di Panduan Pengguna IAM.	2 September 2010

Perubahan	Deskripsi	Tanggal
Pemberitahuan	Fitur pemberitahuan Amazon S3 memungkinkan Anda untuk mengonfigurasikan bucket sehingga Amazon S3 menerbitkan pesan ke topik Amazon Simple Notification Service (Amazon SNS) ketika Amazon S3 mendeteksi peristiwa penting pada bucket. Untuk informasi selengkapnya, lihat Mengatur Pemberitahuan Peristiwa Bucket.	14 Juli 2010
Kebijakan bucket	Kebijakan bucket adalah sistem manajemen akses yang Anda gunakan untuk mengatur izin akses di seluruh bucket, objek, dan kumpulan objek. Fungsi ini melengkapi dan dalam banyak kasus menggantikan daftar kontrol akses. Untuk informasi selengkapnya, lihat Kebijakan bucket untuk Amazon S3.	6 Juli 2010
Sintaksis gaya jalur tersedia di semua Wilayah	Amazon S3 kini mendukung sintaksis gaya jalur untuk bucket mana pun di Wilayah Klasik AS, atau jika bucket berada di Wilayah yang sama dengan titik akhir permintaan. Untuk informasi selengkapnya, lihat Hosting Virtual.	9 Juni 2010
Titik akhir baru untuk Eropa (Irlandia)	Amazon S3 kini menyediakan titik akhir untuk Eropa (Irlandia): `http://s3-eu-west-1.amazonaws.com` .	9 Juni 2010
Konsol	Anda kini dapat menggunakan Amazon S3 melalui AWS Management Console. Anda dapat membaca tentang semua fungsionalitas Amazon S3 di konsol dalam Panduan Pengguna Amazon Simple Storage Service.	9 Juni 2010
Pengurangan Redundansi	Amazon S3 kini memungkinkan Anda mengurangi biaya penyimpanan dengan menyimpan objek di Amazon S3 dengan pengurangan redundansi. Untuk informasi selengkapnya, lihat Reduced Redundancy Storage.	12 Mei 2010
Wilayah baru yang didukung	Amazon S3 kini mendukung Wilayah Asia Pasifik (Singapura). Untuk informasi selengkapnya, lihat Bucket dan Wilayah.	28 April 2010

Perubahan	Deskripsi	Tanggal
Penentuan Versi Objek	Rilis ini memperkenalkan Penentuan Versi objek. Semua objek sekarang dapat memiliki kunci dan versi. Jika Anda mengaktifkan Penentuan Versi untuk bucket, Amazon S3 memberikan ID versi unik kepada semua objek yang ditambahkan ke bucket. Fitur ini memungkinkan Anda memulihkan dari penimpaan dan penghapusan yang tidak diinginkan. Untuk informasi selengkapnya, lihat Penentuan Versi dan Menggunakan Penentuan Versi.	8 Februari 2010
Wilayah baru yang didukung	Amazon S3 kini mendukung Wilayah AS Barat (Californ ia Utara). Titik akhir baru untuk permintaan pada Wilayah ini adalah `s3-us-west-1.amazonaws.com` . Untuk informasi selengkapnya, lihat Bucket dan Wilayah.	2 Desember 2009
AWS SDK untuk .NET	AWS sekarang menyediakan pustaka, kode sampel, tutorial, dan sumber daya lainnya untuk pengembang perangkat lunak yang lebih suka membangun aplikasi menggunakan operasi API khusus bahasa.NET daripada REST atau SOAP. Pustaka ini menyediakan fungsi dasar (tidak termasuk APIs dalam REST atau SOAP), seperti autentikasi permintaan, percobaan ulang permintaan, dan penanganan kesalahan sehingga lebih mudah untuk memulai. Untuk informasi selengkapnya tentang pustaka dan sumber daya khusus bahasa, lihat Mengembangkan dengan Amazon S3 menggunakan Referensi API Amazon S3 di AWS SDKs Amazon S3.	11 November 2009

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