aws

Panduan Developerr

Amazon Glacier

Versi API 2012-06-01

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

Amazon Glacier: Panduan Developerr

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

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

Table of Contents

..... x
Apa itu Amazon Glacier? 1
Apakah Anda Saat Ini Menggunakan Amazon Glacier? 1
Model Data 3
Vault 3
Arsip 4
Pekerjaan 5
Konfigurasi notifikasi 6
Operasi yang Didukung 6
Operasi Vault 7
Operasi Arsip 7
Lowongan 7
Mengakses Amazon Glacier 7
Wilayah dan titik akhir 8
Memulai 9
Langkah 1: Sebelum Anda Mulai 10
Mengatur sebuah Akun AWS 10
Unduh AWS SDK yang Sesuai 12
Langkah 2: Buat Vault 13
Langkah 3: Unggah Arsip ke Vault 14
Unggah Arsip dengan Menggunakan Java 15
Unggah Arsip dengan Menggunakan .NET 21
Langkah 4: Unduh Arsip dari Vault 22
Unduh Arsip dengan Menggunakan Java 24
Unduh Arsip dengan Menggunakan .NET 25
Langkah 5: Hapus Arsip dari Vault 27
Bagian Terkait 28
Menghapus Arsip dengan Menggunakan Java 28
Menghapus Arsip dengan Menggunakan .NET 30
Menghapus Arsip dengan Menggunakan AWS CLI 31
Langkah 6: Hapus Vault 34
Dari Sini, Ke Mana Lagi? 35
Bekerja dengan Vault 36
Operasi Vault di Amazon Glacier 37

Membuat dan Menghapus Vault 37
Mengambil Metadata Vault 37
Mengunduh Inventaris Vault 37
Mengonfigurasi Notifikasi Vault 38
Membuat Vault 39
Membuat Vault Menggunakan Java 39
Membuat Vault Menggunakan .NET 43
Membuat Vault Menggunakan REST 47
Membuat Vault Menggunakan Konsol 47
Membuat Vault Menggunakan AWS CLI 48
Mengambil Metadata Vault 49
Mengambil Metadata Vault Menggunakan Java 50
Mengambil Metadata Vault Menggunakan .NET 52
Mengambil Metadata Vault Menggunakan REST 55
Mengambil Metadata Vault Menggunakan AWS CLI 55
Mengunduh Inventaris Vault 56
Tentang Inventaris 58
Mengunduh Inventaris Vault Menggunakan Java 59
Mengunduh Inventaris Vault Menggunakan .NET 66
Mengunduh Inventaris Vault Menggunakan REST 74
Mengunduh Inventaris Vault Menggunakan AWS CLI 74
Mengonfigurasi Notifikasi Vault 77
Konsep Umum 78
Mengonfigurasi Notifikasi Vault Menggunakan Java 79
Mengonfigurasi Notifikasi Vault Menggunakan .NET 83
Mengonfigurasi Notifikasi Vault Menggunakan REST API 86
Mengkonfigurasi Pemberitahuan Vault dengan Menggunakan Konsol 86
Mengkonfigurasi Pemberitahuan Vault Menggunakan CLI 89
Menghapus Vault 90
Menghapus Vault Menggunakan Java 91
Menghapus Vault Menggunakan .NET 92
Menghapus Vault Menggunakan REST 94
Menghapus Vault Kosong dengan Menggunakan Konsol 94
Menghapus Vault Menggunakan AWS CLI 95
Menandai Vault 99
Menandai Vaults dengan Menggunakan Konsol Amazon Glacier 99

Menandai Vaults dengan Menggunakan AWS CLI 101
Menandai Vaults dengan Menggunakan Amazon Glacier API 101
Bagian Terkait 102
Kunci Vault 102
Gambaran Umum Penguncian Vault 102
Penguncian Vault dengan Menggunakan API 104
Penguncian Vault Menggunakan CLI 105
Penguncian Vault dengan Menggunakan Konsol 107
Bekerja dengan Arsip 110
Operasi Arsip 111
Mengunggah Arsip 111
Mencari Arsip 111
Mengunduh Arsip 111
Menghapus Arsip 112
Memperbarui Arsip 112
Mempertahankan Metadata Arsip Sisi Klien 112
Mengunggah Arsip 113
Opsi untuk Mengunggah Arsip 113
Mengunggah Arsip dalam Satu Operasi 114
Mengunggah Arsip Besar dalam Beberapa Bagian 124
Mengunduh Arsip 143
Mengambil Arsip 143
Mengunduh Arsip Menggunakan Java 148
Mengunduh Arsip Menggunakan .NET 165
Mengunduh Arsip Besar Menggunakan Python 181
Mengunduh Arsip dengan Menggunakan REST 189
Mengunduh Arsip Menggunakan AWS CLI 190
Menghapus Arsip 193
Menghapus Arsip Menggunakan Java 194
Menghapus Arsip Menggunakan .NET 196
Menghapus Arsip Menggunakan REST 199
Menghapus Arsip Menggunakan AWS CLI 200
Menggunakan AWS SDKs 203
AWS Perpustakaan SDK untuk Java dan .NET 203
Apa itu API Tingkat Rendah? 203
Apa itu API Tingkat Tinggi? 204

Kapan Menggunakan API Tingkat Tinggi dan Tingkat Rendah 204
Bekerja dengan AWS SDKs 205
Menggunakan AWS SDK for Java 206
Menggunakan API Tingkat Rendah 206
Menggunakan API Tingkat Tinggi 207
Menjalankan Contoh Java Menggunakan Eclipse 208
Mengatur Titik Akhir 209
Menggunakan AWS SDK for .NET 210
Menggunakan API Tingkat Rendah 210
Menggunakan API Tingkat Tinggi 211
Menjalankan Contoh .NET 212
Mengatur Titik Akhir 213
Contoh kode 214
Hal-hal mendasar 215
Halo Amazon Glacier 216
Tindakan 217
Skenario 282
Mengarsipkan file, mendapatkan notifikasi, dan memulai pekerjaan 283
Dapatkan konten arsip dan hapus arsip 289
Keamanan 295
Perlindungan Data 296
Enkripsi data 296
Manajemen kunci 297
Privasi Lalu Lintas Kerja Internet 297
Identity and Access Management 298
Audiens 298
Mengautentikasi dengan identitas 299
Mengelola akses menggunakan kebijakan 300
Bagaimana Amazon Glacier bekerja dengan IAM 302
Contoh kebijakan berbasis identitas 309
Contoh kebijakan berbasis sumber daya 315
Pemecahan masalah 318
Referensi Izin API Amazon Glacier 320
Pembuatan Log dan Pemantauan 328
Validasi Kepatuhan 329
Ketahanan 331

Keamanan Infrastruktur 332
VPC Endpoint 332
Kebijakan Pengambilan Data 333
Memilih Kebijakan Pengambilan Data Amazon Glacier 333
Kebijakan Hanya Tingkat Gratis 334
Kebijakan Tingkat Pengambilan Maksimal 335
Kebijakan Tidak Ada Batas Pengambilan 335
Menggunakan Konsol Amazon Glacier untuk Mengatur Kebijakan Pengambilan Data 335
Menggunakan Amazon Glacier API untuk Mengatur Kebijakan Pengambilan Data 336
Menggunakan Amazon Glacier REST API untuk Mengatur Kebijakan Pengambilan Data 336
Menggunakan AWS SDKs untuk Mengatur Kebijakan Pengambilan Data 337
Memberi Tag pada Sumber Daya 338
Dasar Penandaan 338
Pembatasan Tag 339
Melacak Biaya Menggunakan Penandaan 339
Mengelola Kontrol Akses dengan Penandaan 340
Bagian Terkait 340
Audit Logging dengan AWS CloudTrail 341
Informasi Amazon Glacier di CloudTrail 341
Memahami Entri File Log Amazon Glacier 342
Referensi API 346
Header Permintaan Umum 347
Header Respons Umum 350
Menandatangani Permintaan 351
Contoh Perhitungan Tanda Tangan 352
Menghitung Tanda Tangan untuk Operasi Streaming 354
Checksum Komputasi 356
Contoh Hash Pohon 1: Mengunggah arsip dalam satu permintaan 358
Contih Hash Pohon 2: Mengunggah arsip menggunakan unggahan multipart 359
Menghitung Hash Pohon File 360
Menerima Checksum Saat Mengunduh Data 369
Respons Kesalahan 372
Contoh 1: Menjelaskan permintaan Tugas dengan ID tugas yang tidak ada 375
Contoh 2: Mencantumkan permintaan Tugas dengan nilai yang tidak valid untuk parameter permintaan 376
Operasi Vault 377

Membatalkan Kunci Vault 378
Menambahkan Tanda ke Vault 381
Membuat Vault 384
Menyelesaikan Kunci Vault 387
Menghapus Vault 390
Menghapus Kebijakan Akses Vault 393
Menghapus Notifikasi Vault 396
Menjelaskan Vault 399
Mendapatkan Kebijakan Akses Vault 403
Mendapatkan Kunci Vault 406
Mendapatkan Notifikasi Vault 411
Memulai Kunci Vault 414
Mencantumkan Tanda untuk Vault 419
Mencantumkan Vault 422
Hapus Tag Dari Vault 429
Mendapatkan Kebijakan Akses Vault 432
Menetapkan Konfigurasi Notifikasi Vault 436
Operasi Arsip 440
Menghapus Arsip 440
Mengunggah Arsip 443
Operasi Unggahan Multipart 448
Membatalkan Unggahan Multipart 449
Selesaikan Unggahan Multibagian 451
Mulai Unggahan Multibagian 456
Mencantumkan Bagian 462
Mencantumkan Unggahan Multipart 469
Mengunggah Bagian 476
Operasi Tugas 483
Mendeskripsikan Tugas 483
Mendapatkan Output Tugas 494
Memulai Tugas 505
Mencantumkan Tugas 516
Tipe Data yang Digunakan dalam Operasi Tugas 526
CSVInput 527
CSVOutput 528
Enkripsi 530

GlacierJobDescription 531
Pemberian Izin 534
Penerima izin 535
InputSerialization 536
InventoryRetrievalJobInput 537
jobParameters 538
OutputLocation 541
OutputSerialization 541
S3Location 542
SelectParameters 544
Operasi Pengambilan Data 545
Mendapatkan Kebijakan Pengambilan Data 545
Mencantumkan Kapasitas yang Disediakan 549
Membeli Kapasitas yang Disediakan. 552
Mengatur Kebijakan Pengambilan Data 556
Riwayat Dokumen 562
Pembaruan Sebelumnya 563
AWS Glosarium 566

Halaman ini hanya untuk pelanggan lama layanan Amazon Glacier menggunakan Vaults dan REST API asli dari 2012.

Jika Anda mencari solusi penyimpanan arsip, sebaiknya gunakan kelas penyimpanan Amazon Glacier di Amazon S3, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval, dan S3 Glacier Deep Archive. Untuk mempelajari lebih lanjut tentang opsi penyimpanan ini, lihat kelas penyimpanan Amazon Glacier.

Amazon Glacier (layanan berbasis brankas mandiri asli) tidak lagi menerima pelanggan baru. Amazon Glacier adalah layanan mandiri dengan miliknya APIs sendiri yang menyimpan data di brankas dan berbeda dari Amazon S3 dan kelas penyimpanan Amazon S3 Glacier. Data Anda yang ada akan tetap aman dan dapat diakses di Amazon Glacier tanpa batas waktu. Tidak diperlukan migrasi. Untuk penyimpanan arsip jangka panjang berbiaya rendah, AWS rekomendasikan kelas penyimpanan Amazon S3 Glacier, yang memberikan pengalaman pelanggan yang unggul dengan APIs berbasis ember S3, ketersediaan penuh, biaya lebih rendah, AWS Region dan integrasi layanan. AWS Jika Anda ingin meningkatkan kemampuan, pertimbangkan untuk bermigrasi ke kelas penyimpanan Amazon S3 Glacier dengan menggunakan Panduan Solusi AWS kami untuk mentransfer data dari kubah Amazon Glacier ke kelas penyimpanan Amazon S3 Glacier.

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

Apa itu Amazon Glacier?

Jika saat ini Anda menggunakan layanan Amazon Glacier dan ingin mempelajari lebih lanjut, Anda akan menemukan informasi yang Anda butuhkan dalam panduan ini. Amazon Glacier adalah layanan yang aman dan tahan lama untuk pengarsipan data berbiaya rendah dan pencadangan jangka panjang menggunakan brankas. Untuk informasi selengkapnya tentang harga layanan Amazon Glacier, lihat harga Amazon Glacier.

Topik

- Apakah Anda Saat Ini Menggunakan Amazon Glacier?
- Model Data Amazon Glacier
- Operasi yang Didukung di Amazon Glacier
- Mengakses Amazon Glacier

Apakah Anda Saat Ini Menggunakan Amazon Glacier?

> **Note**
>
> Bagian ini adalah tentang layanan Amazon Glacier. Jika saat ini Anda menggunakan kelas penyimpanan Amazon S3 Glacier (S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval, dan S3 Glacier Deep Archive), lihat Kelas penyimpanan untuk mengarsipkan objek di Panduan Pengguna Amazon S3.

Jika saat ini Anda menggunakan layanan Amazon Glacier dan ingin mempelajari lebih lanjut, kami sarankan Anda mulai dengan membaca bagian berikut:

- Apa itu Amazon Glacier - Bagian lainnya menjelaskan model data yang mendasarinya, operasi yang didukungnya, dan AWS SDKs yang dapat Anda gunakan untuk berinteraksi dengan layanan.
- Memulai — Memulai dengan Amazon Glacier Bagian ini memandu Anda melalui proses pembuatan lemari besi, mengunggah arsip, membuat pekerjaan untuk mengunduh arsip, mengambil hasil pekerjaan, dan menghapus arsip.

> **Important**
>
> Amazon Glacier memang menyediakan konsol. Namun, setiap operasi arsip, seperti mengunggah, mengunduh, atau menghapus, mengharuskan Anda untuk menggunakan AWS Command Line Interface (AWS CLI) atau menulis kode. Tidak ada dukungan konsol untuk operasi arsip. Misalnya, untuk mengunggah data, seperti foto, video, dan dokumen lainnya, Anda harus menggunakan AWS CLI atau menulis kode untuk membuat permintaan, dengan menggunakan REST API secara langsung atau dengan menggunakan AWS SDKs.
>
> Untuk menginstal AWS CLI, lihat AWS Command Line Interface. Untuk informasi selengkapnya tentang penggunaan Amazon Glacier dengan, lihat AWS CLIAWS CLI Referensi untuk Amazon Glacier. Untuk contoh penggunaan untuk mengunggah arsip AWS CLI ke Amazon Glacier, lihat Menggunakan Amazon Glacier with the. AWS Command Line Interface

Di luar bagian memulai, Anda mungkin ingin mempelajari lebih lanjut tentang operasi Amazon Glacier. Bagian berikut memberikan informasi rinci tentang bekerja dengan Amazon Glacier dengan menggunakan REST API dan untuk Java dan Microsoft AWS SDKs .NET:

- Menggunakan AWS SDKs dengan Amazon Glacier

  Bagian ini memberikan gambaran umum tentang yang AWS SDKs digunakan dalam berbagai contoh kode dalam panduan ini. Ulasan bagian ini akan membantu saat membaca bagian berikut. Ini mencakup ikhtisar tingkat tinggi dan tingkat rendah APIs yang SDKs ditawarkan ini, kapan menggunakannya, dan langkah-langkah umum untuk menjalankan contoh kode yang disediakan dalam panduan ini.

- Bekerja dengan Vaults di Amazon Glacier

  Bagian ini memberikan detail berbagai operasi vault, seperti membuat vault, mengambil metadata vault, menggunakan pekerjaan untuk mengambil inventaris vault, dan mengonfigurasi pemberitahuan vault. Selain menggunakan konsol Amazon Glacier, Anda dapat menggunakan untuk berbagai operasi AWS SDKs brankas. Bagian ini menjelaskan API dan menyediakan sampel kerja dengan menggunakan AWS SDK for Java dan AWS SDK for .NET.

- Bekerja dengan Arsip di Amazon Glacier

Bagian ini memberikan rincian operasi arsip, seperti mengunggah arsip dalam satu permintaan atau menggunakan operasi unggahan multibagian untuk mengunggah arsip besar di beberapa bagian. Bagian ini juga menjelaskan cara membuat pekerjaan untuk mengunduh arsip secara asinkron. Bagian ini memberikan contoh dengan menggunakan AWS SDK for Java dan AWS SDK for .NET.

- Referensi API untuk Amazon Glacier

  Amazon Glacier adalah layanan. RESTful Bagian ini menjelaskan operasi REST, termasuk sintaks, serta contoh permintaan dan respons untuk semua operasi. Pustaka AWS SDK membungkus API ini, menyederhanakan tugas pemrograman Anda.

Model Data Amazon Glacier

Komponen inti model data Amazon Glacier termasuk brankas dan arsip. Amazon Glacier adalah layanan web berbasis REST. Dalam hal REST, vault dan arsip adalah sumber daya. Selain itu, model data Amazon Glacier mencakup sumber daya konfigurasi pekerjaan dan pemberitahuan. Sumber daya ini melengkapi sumber daya inti.

Topik

- Vault
- Arsip
- Pekerjaan
- Konfigurasi notifikasi

Vault

Di Amazon Glacier, lemari besi adalah wadah untuk menyimpan arsip. Vault mirip dengan ember Amazon S3. Saat membuat vault, Anda menentukan nama dan memilih AWS Region tempat Anda ingin membuat brankas.

Setiap sumber daya vault memiliki alamat yang unik. Bentuk umumnya adalah:

```
https://region-specific-endpoint/account-id/vaults/vault-name
```

Misalnya, Anda membuat vault (`examplevault`) di Wilayah AS Barat (Oregon) di akun Anda dengan ID 111122223333. Anda dapat mengatasi vault ini dengan menggunakan URI berikut:

```
https://glacier.us-west-2.amazonaws.com/111122223333/vaults/examplevault
```

Inilah arti berbagai komponen URI:

- `glacier.us-west-2.amazonaws.com` mengidentifikasi Wilayah US West (Oregon).
- `111122223333`adalah Akun AWS ID yang memiliki brankas.
- `vaults`mengacu pada koleksi brankas yang dimiliki oleh. Akun AWS
- `examplevault` mengidentifikasi vault tertentu dalam kumpulan vault.

An Akun AWS dapat membuat brankas dalam apa pun yang didukung. AWS Region Untuk daftar yang didukung Wilayah AWS, lihatMengakses Amazon Glacier. Dalam Wilayah, akun harus menggunakan nama vault yang unik. An Akun AWS dapat membuat kubah dengan nama yang sama di Wilayah yang berbeda.

Anda dapat menyimpan arsip dalam jumlah yang tidak terbatas di vault. Bergantung pada kebutuhan bisnis atau aplikasi Anda, Anda dapat menyimpan arsip ini dalam satu vault atau beberapa vault.

Amazon Glacier mendukung berbagai operasi lemari besi. Operasi Vault bersifat spesifik Wilayah. Misalnya, ketika Anda membuat vault, Anda membuatnya di Wilayah tertentu. Saat Anda meminta daftar vault, Anda memintanya dari daftar tertentu AWS Region, dan daftar yang dihasilkan hanya menyertakan vault yang dibuat di Wilayah tertentu tersebut.

Arsip

Arsip dapat berupa data apa pun, seperti foto, video, atau dokumen. Arsip mirip dengan objek Amazon S3, dan merupakan unit dasar penyimpanan di Amazon Glacier. Setiap arsip memiliki ID unik dan deskripsi opsional. Anda dapat menentukan deskripsi opsional ini hanya selama pengunggahan arsip. Amazon Glacier memberikan arsip ID, yang unik di AWS Region mana arsip disimpan.

Setiap arsip memiliki alamat yang unik. Format yang umum adalah sebagai berikut:

```
https://region-specific-endpoint/account-id/vaults/vault-name/archives/archive-id
```

Berikut ini adalah contoh URI dari arsip yang disimpan di `examplevault` brankas di Wilayah Barat AS (Oregon) di akun 111122223333:

```
https://glacier.us-west-2.amazonaws.com/111122223333/vaults/
examplevault/archives/NkbByEejwEggmBz2fTHgJrg0XBoDfjP4q6iu87-
TjhqG6eGoOY9Z8i1_AUyUsuhPAdTqLHy8pTl5nfCFJmDl2yEZONi5L26Omw12vcs01MNGntHEQL8MBfGlqrEXAMPLEArchi
```

Anda dapat menyimpan arsip dalam jumlah yang tidak terbatas di vault.

Pekerjaan

Pekerjaan Amazon Glacier dapat mengambil arsip, atau mendapatkan inventaris lemari besi.

Mengambil arsip dan inventaris brankas (daftar arsip) adalah operasi asinkron di Amazon Glacier, di mana Anda pertama kali memulai pekerjaan, dan kemudian mengunduh hasil pekerjaan setelah Amazon Glacier menyelesaikan pekerjaan.

> **Note**
>
> Amazon Glacier menawarkan solusi arsip data penyimpanan dingin. Jika aplikasi Anda memerlukan solusi penyimpanan yang memerlukan pengambilan data waktu nyata, Anda dapat mempertimbangkan untuk menggunakan Amazon S3. Untuk informasi selengkapnya, lihat Amazon Simple Storage Service (Amazon S3).

Untuk memulai tugas inventaris vault, Anda memberi nama vault. Pekerjaan pengambilan arsip memerlukan nama vault dan ID arsip. Anda juga dapat memberikan deskripsi tugas opsional untuk membantu mengidentifikasi tugas.

Pekerjaan pengambilan arsip dan inventaris vault dikaitkan dengan brankas. Vault dapat memiliki beberapa tugas yang berlangsung pada titik waktu apa pun. Saat Anda mengirim permintaan pekerjaan (memulai pekerjaan), Amazon Glacier mengembalikan ID pekerjaan kepada Anda untuk melacak pekerjaan tersebut. Setiap tugas diidentifikasi secara unik oleh URI formulir:

```
https://region-specific-endpoint/account-id/vaults/vault-name/jobs/job-id
```

Berikut ini adalah contoh pekerjaan yang terkait dengan `examplevault` lemari besi di Wilayah Barat AS (Oregon) di akun 111122223333.

```
https://glacier.us-west-2.amazonaws.com/111122223333/vaults/examplevault/jobs/
HkF9p6o7yjhFx-
K3CGl6fuSm6VzW9T7esGQfco8nUXVYwS0jlb5gq1JZ55yHgt5vP54ZShjoQzQVVh7vEXAMPLEjobID
```

Untuk setiap pekerjaan, Amazon Glacier menyimpan informasi, seperti jenis pekerjaan, deskripsi, tanggal pembuatan, tanggal penyelesaian, dan status pekerjaan. Anda dapat memperoleh informasi tentang tugas spesifik atau mendapatkan daftar semua tugas Anda yang terkait dengan vault. Daftar pekerjaan yang dikembalikan Amazon Glacier mencakup semua pekerjaan yang sedang berjalan dan baru saja selesai.

Konfigurasi notifikasi

Karena pekerjaan membutuhkan waktu untuk dijalankan, Amazon Glacier mendukung mekanisme notifikasi untuk memberi tahu Anda saat pekerjaan selesai. Anda dapat mengonfigurasi vault untuk mengirim notifikasi ke topik Amazon Simple Notification Service (Amazon SNS) saat tugas selesai. Anda dapat menentukan satu topik Amazon SNS per vault dalam konfigurasi notifikasi.

Amazon Glacier menyimpan konfigurasi notifikasi sebagai dokumen JSON. Berikut adalah contoh konfigurasi notifikasi vault:

```
{
   "Topic": "arn:aws:sns:us-west-2:111122223333:mytopic",
   "Events": ["ArchiveRetrievalCompleted", "InventoryRetrievalCompleted"]
}
```

Konfigurasi notifikasi dikaitkan dengan brankas; Anda dapat memilikinya untuk setiap brankas. Setiap sumber daya konfigurasi notifikasi secara unik diidentifikasi oleh URI formulir:

```
https://region-specific-endpoint/account-id/vaults/vault-name/notification-
configuration
```

Amazon Glacier mendukung operasi untuk mengatur, mendapatkan, dan menghapus konfigurasi notifikasi. Saat Anda menghapus konfigurasi notifikasi, tidak ada notifikasi yang dikirim saat operasi pengambilan data apa pun di vault selesai.

Operasi yang Didukung di Amazon Glacier

Untuk bekerja dengan brankas dan arsip (lihatModel Data Amazon Glacier), Amazon Glacier mendukung serangkaian operasi. Di antara semua operasi yang didukung, hanya operasi berikut yang asinkron:

- Mengambil arsip
- Mengambil inventaris vault (daftar arsip)

Operasi ini mengharuskan Anda memulai tugas terlebih dahulu, lalu mengunduh output tugas. Bagian berikut merangkum operasi Amazon Glacier.

Operasi Vault

Amazon Glacier menyediakan operasi untuk membuat dan menghapus brankas. Anda dapat memperoleh deskripsi brankas untuk brankas tertentu atau untuk semua brankas di file. AWS Region Deskripsi brankas memberikan informasi, seperti tanggal pembuatan, jumlah arsip di lemari besi, ukuran total dalam byte yang digunakan oleh semua arsip di lemari besi, dan tanggal Amazon Glacier menghasilkan inventaris brankas. Amazon Glacier juga menyediakan operasi untuk mengatur, mengambil, dan menghapus konfigurasi notifikasi di brankas. Untuk informasi selengkapnya, lihat Bekerja dengan Vaults di Amazon Glacier.

Operasi Arsip

Amazon Glacier menyediakan operasi bagi Anda untuk mengunggah dan menghapus arsip. Anda tidak dapat memperbarui arsip yang ada; Anda harus menghapus arsip yang ada dan mengunggah arsip baru. Setiap kali Anda mengunggah arsip, Amazon Glacier menghasilkan ID arsip baru. Untuk informasi selengkapnya, lihat Bekerja dengan Arsip di Amazon Glacier.

Lowongan

Anda dapat memulai pekerjaan Amazon Glacier untuk melakukan pengambilan di arsip atau mendapatkan inventaris brankas.

Berikut ini adalah jenis pekerjaan Amazon Glacier:

- `archive-retrieval`— Ambil arsip.

  Untuk informasi selengkapnya, lihat Mengunduh Arsip di Amazon Glacier.
- `inventory-retrieval`— Inventarisasi lemari besi.

  Untuk informasi selengkapnya, lihat Mengunduh Inventaris Vault di Amazon Glacier.

Mengakses Amazon Glacier

Amazon Glacier adalah layanan web RESTful yang menggunakan HTTP dan HTTPS sebagai protokol transportasi JavaScript dan Object Notation (JSON) sebagai format serialisasi pesan. Kode aplikasi Anda dapat membuat permintaan langsung ke API layanan web Amazon Glacier.

Ketika menggunakan API REST secara langsung, Anda harus menulis kode yang diperlukan untuk menandatangani dan mengautentikasi permintaan Anda. Untuk informasi selengkapnya tentang API, lihat Referensi API untuk Amazon Glacier.

Atau, Anda dapat menyederhanakan pengembangan aplikasi dengan menggunakan AWS SDKs yang membungkus panggilan Amazon Glacier REST API. Anda menentukan kredensial Anda, dan pustaka ini menangani autentikasi serta meminta penandatanganan. Untuk informasi selengkapnya tentang menggunakan AWS SDKs, lihat Menggunakan AWS SDKs dengan Amazon Glacier.

Amazon Glacier juga menyediakan konsol. Namun, semua operasi arsip dan pekerjaan mengharuskan Anda untuk menulis kode dan membuat permintaan dengan menggunakan REST API secara langsung atau pustaka pembungkus AWS SDK. Untuk mengakses konsol Amazon Glacier, buka Konsol Amazon Glacier.

Wilayah dan titik akhir

Anda membuat lemari besi di tempat tertentu AWS Region. Anda selalu mengirim permintaan Amazon Glacier Anda ke titik akhir khusus untuk file. AWS RegionUntuk daftar yang Wilayah AWS didukung oleh Amazon Glacier, lihat titik akhir dan kuota Amazon Glacier di Referensi Umum.AWS

Memulai dengan Amazon Glacier

Anda dapat memulai dengan Amazon Glacier (Amazon Glacier) dengan bekerja dengan brankas dan arsip. Vault adalah wadah untuk menyimpan arsip, dan arsip adalah objek apa pun, seperti foto, video, atau dokumen, yang Anda simpan di lemari besi. Arsip adalah unit dasar penyimpanan di Amazon Glacier. Latihan memulai ini memberikan instruksi bagi Anda untuk menjelajahi operasi dasar Amazon Glacier di brankas dan arsip. Untuk informasi lebih lanjut tentang sumber daya ini, lihat Model Data Amazon Glacier bagian.

Dalam latihan memulai, Anda akan membuat brankas, mengunggah dan mengunduh arsip, lalu menghapus arsip dan brankas. Anda dapat melakukan semua operasi ini secara terprogram. Namun, latihan memulai menggunakan konsol manajemen Amazon Glacier untuk membuat dan menghapus brankas. Untuk mengunggah dan mengunduh arsip, bagian memulai ini menggunakan API tingkat tinggi untuk AWS SDK for Java dan file. AWS SDK for .NET API tingkat tinggi memberikan pengalaman pemrograman yang disederhanakan saat bekerja dengan Amazon Glacier. Untuk informasi selengkapnya tentang penggunaan API tingkat tinggi dengan API AWS SDKs, lihatMenggunakan AWS SDKs dengan Amazon Glacier.

> **Important**
>
> Amazon Glacier memang menyediakan konsol. Namun, setiap operasi arsip, seperti mengunggah, mengunduh, atau menghapus, mengharuskan Anda untuk menggunakan (AWS Command Line Interface CLI) atau menulis kode. Tidak ada dukungan konsol untuk operasi arsip. Misalnya, untuk mengunggah data, seperti foto, video, dan dokumen lainnya, Anda harus menggunakan AWS CLI atau menulis kode untuk membuat permintaan, dengan menggunakan REST API secara langsung atau dengan menggunakan AWS SDKs. Untuk menginstal AWS CLI, lihat AWS Command Line Interface. Untuk informasi selengkapnya tentang penggunaan Amazon Glacier dengan, lihat AWS CLIAWS CLI Referensi untuk Amazon Glacier. Untuk contoh penggunaan untuk mengunggah arsip AWS CLI ke Amazon Glacier, lihat Menggunakan Amazon Glacier with the. AWS Command Line Interface

Latihan memulai ini memberikan contoh kode di Java dan C# bagi Anda untuk mengunggah dan mengunduh arsip. Bagian terakhir dari latihan memulai menyediakan langkah-langkah di mana Anda dapat mempelajari lebih lanjut tentang pengalaman pengembang dengan Amazon Glacier.

Topik

- Langkah 1: Sebelum Anda Memulai dengan Amazon Glacier
- Langkah 2: Buat Vault di Amazon Glacier
- Langkah 3: Unggah Arsip ke Vault di Amazon Glacier
- Langkah 4: Unduh Arsip dari Vault di Amazon Glacier
- Langkah 5: Hapus Arsip dari Vault di Amazon Glacier
- Langkah 6: Hapus Vault di Amazon Glacier
- Dari Sini, Ke Mana Lagi?

Langkah 1: Sebelum Anda Memulai dengan Amazon Glacier

Sebelum Anda dapat memulai dengan latihan ini, Anda harus mendaftar untuk Akun AWS (jika Anda belum memilikinya), dan kemudian unduh salah satunya AWS SDKs. Lihat bagian berikut untuk instruksi.

Topik

- Mengatur Akun AWS dan Pengguna Administrator
- Unduh AWS SDK yang Sesuai

Mengatur Akun AWS dan Pengguna Administrator

Jika Anda belum melakukannya, Anda harus mendaftar untuk Akun AWS dan membuat pengguna administrator di akun.

Untuk menyelesaikan pengaturan, ikuti instruksi dalam topik berikut.

Mengatur Akun AWS dan Membuat Pengguna Administrator

Mendaftar untuk AWS

Saat Anda mendaftar ke Amazon Web Services (AWS), Anda Akun AWS secara otomatis mendaftar untuk semua layanan AWS, termasuk Amazon Glacier. Anda hanya membayar biaya layanan yang Anda gunakan. Untuk informasi selengkapnya tentang tarif penggunaan Amazon Glacier, lihat halaman harga Amazon Glacier.

Jika Anda sudah memiliki Akun AWS, lompat keUnduh AWS SDK yang Sesuai. Jika Anda tidak memiliki Akun AWS, gunakan prosedur berikut untuk membuatnya.

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.

Untuk membuat pengguna administrator, pilih salah satu opsi berikut.

Pilih salah satu cara untuk mengelola administrator Anda	Untuk	Oleh	Anda juga bisa
Di Pusat Identitas IAM (Direkomendasikan)	Gunakan kredensi jangka pendek untuk mengakses. AWS Ini sejalan dengan praktik terbaik keamanan. Untuk informasi tentang	Mengikuti petunjuk di Memulai di Panduan AWS IAM Identity Center Pengguna.	Konfigurasikan akses terprogram dengan Mengonfigurasi AWS CLI yang akan digunakan AWS IAM Identity Center dalam AWS Command Line Interface Panduan Pengguna.

Pilih salah satu cara untuk mengelola administrator Anda	Untuk	Oleh	Anda juga bisa
	praktik terbaik, lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.		
Di IAM (Tidak direkomendasikan)	Gunakan kredensi jangka panjang untuk mengakses. AWS	Mengikuti petunjuk di Buat pengguna IAM untuk akses darurat di Panduan Pengguna IAM.	Konfigurasikan akses terprogram dengan Mengelola kunci akses untuk pengguna IAM di Panduan Pengguna IAM.

Unduh AWS SDK yang Sesuai

Untuk mencoba latihan memulai, Anda harus memutuskan bahasa pemrograman mana yang ingin Anda gunakan, dan kemudian mengunduh AWS SDK yang sesuai untuk platform pengembangan Anda.

Latihan memulai memberikan contoh di Java dan C#.

Mengunduh AWS SDK for Java

Untuk menguji contoh Java dalam panduan developer ini, Anda memerlukan AWS SDK for Java. Anda memiliki opsi unduhan berikut:

- Jika Anda menggunakan Eclipse, Anda dapat mengunduh dan menginstal AWS Toolkit for Eclipse dengan menggunakan situs pembaruan http://aws.amazon.com/eclipse/. Untuk informasi selengkapnya, lihat AWS Toolkit for Eclipse.

- Jika Anda menggunakan IDE lain untuk membuat aplikasi Anda, unduh AWS SDK for Java.

Mengunduh AWS SDK for .NET

Untuk menguji contoh C# dalam panduan developer ini, Anda memerlukan AWS SDK for .NET. Anda memiliki opsi unduhan berikut:

- Jika Anda menggunakan Visual Studio, Anda dapat menginstal kedua AWS SDK for .NET dan AWS Toolkit for Visual Studio. Toolkit menyediakan AWS Explorer untuk Visual Studio dan template proyek yang dapat Anda gunakan untuk pengembangan. Untuk mengunduh AWS SDK for .NET, buka http://aws.amazon.com/sdkfornet. Secara default, skrip instalasi menginstal AWS SDK dan file. AWS Toolkit for Visual Studio Untuk mempelajari selengkapnya tentang toolkit, lihat Panduan AWS Toolkit for Visual Studio Pengguna.
- Jika Anda menggunakan IDE lain untuk membuat aplikasi Anda, Anda dapat menggunakan tautan yang sama disediakan dalam langkah sebelumnya dan cukup menginstal AWS SDK for .NET.

Langkah 2: Buat Vault di Amazon Glacier

Vault adalah kontainer untuk menyimpan arsip. Langkah pertama Anda adalah membuat brankas di salah satu yang didukung Wilayah AWS. Untuk daftar yang didukung oleh Amazon Glacier, lihat titik akhir dan kuota Amazon Glacier di Referensi Umum. Wilayah AWS AWS

Anda dapat membuat brankas secara terprogram atau dengan menggunakan konsol Amazon Glacier. Bagian ini menggunakan konsol untuk membuat vault.

Untuk membuat vault

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Glacier di rumah. https://console.aws.amazon.com/glacier/
2. Di panel navigasi kiri, pilih Vaults.
3. Pilih Buat lemari besi.

   Halaman Create vault terbuka.
4. Di bawah Pilih Wilayah, pilih AWS Region dari pemilih Wilayah. Brankas Anda akan berlokasi di Wilayah yang Anda pilih.
5. Untuk nama Vault, masukkan nama untuk brankas Anda.

Berikut ini adalah persyaratan penamaan brankas:

- Nama brankas harus unik di dalam Akun AWS dan AWS Region di mana lemari besi dibuat.
- Nama vault harus memiliki panjang antara 1 dan 255 karakter.
- Nama vault hanya dapat berisi karakter berikut: a—z, A-Z, 0—9, _ (garis bawah), - (tanda hubung), dan. (periode).

6. Di bawah Pemberitahuan acara, untuk mengaktifkan atau menonaktifkan notifikasi pada vault saat pekerjaan selesai, pilih salah satu pengaturan berikut:

   - Nonaktifkan notifikasi — Pemberitahuan dimatikan, dan notifikasi tidak dikirim ke topik Simple Notification Service Amazon (Amazon SNS) saat pekerjaan tertentu selesai.

   - Aktifkan notifikasi — Pemberitahuan diaktifkan, dan notifikasi dikirim ke topik Amazon SNS yang disediakan saat pekerjaan tertentu selesai.

     Jika Anda memilih Mengaktifkan notifikasi, lihat Mengonfigurasi Pemberitahuan Vault dengan Menggunakan Konsol Amazon Glacier.

7. Jika nama AWS Region dan vault sudah benar, pilih Create vault.

Vault baru Anda sekarang terdaftar di halaman Vaults di konsol Amazon Glacier.

Langkah 3: Unggah Arsip ke Vault di Amazon Glacier

Pada langkah ini, Anda akan mengunggah arsip sampel ke vault yang Anda buat pada langkah sebelumnya (lihat). Langkah 2: Buat Vault di Amazon Glacier Bergantung pada platform pengembangan yang Anda gunakan, pilih salah satu tautan di akhir bagian ini.

> **Important**
>
> Setiap operasi arsip, seperti mengunggah, mengunduh, atau menghapus, mengharuskan Anda untuk menggunakan (AWS Command Line Interface CLI) atau menulis kode. Tidak ada dukungan konsol untuk operasi arsip. Misalnya, untuk mengunggah data, seperti foto, video, dan dokumen lainnya, Anda harus menggunakan AWS CLI atau menulis kode untuk membuat permintaan, dengan menggunakan REST API secara langsung atau dengan menggunakan AWS SDKs.

Untuk menginstal AWS CLI, lihat AWS Command Line Interface. Untuk informasi selengkapnya tentang penggunaan Amazon Glacier dengan, AWS CLI lihat Referensi AWS CLI untuk Amazon Glacier. Untuk contoh penggunaan untuk mengunggah arsip AWS CLI ke Amazon Glacier, lihat Menggunakan Amazon Glacier with the. AWS Command Line Interface

Arsip adalah objek apa pun, seperti foto, video, atau dokumen yang Anda simpan di vault. Arsip adalah unit dasar penyimpanan di Amazon Glacier. Anda dapat mengunggah arsip dalam satu permintaan. Untuk arsip besar, Amazon Glacier menyediakan operasi API unggahan multibagian yang memungkinkan Anda mengunggah arsip dalam beberapa bagian.

Pada bagian memulai ini, Anda mengunggah arsip sampel dalam satu permintaan. Untuk latihan ini, Anda menentukan file yang berukuran lebih kecil. Untuk file yang lebih besar, unggahan multipart cocok untuk digunakan. Untuk informasi selengkapnya, lihat Mengunggah Arsip Besar dalam Beberapa Bagian (Unggahan Multipart).

Topik

- Unggah Arsip ke Vault di Amazon Glacier dengan Menggunakan AWS SDK for Java
- Unggah Arsip ke Vault di Amazon Glacier dengan Menggunakan AWS SDK for .NET

Unggah Arsip ke Vault di Amazon Glacier dengan Menggunakan AWS SDK for Java

Contoh kode Java berikut menggunakan API tingkat tinggi AWS SDK for Java untuk mengunggah arsip sampel ke vault. Dalam contoh kode, perhatikan hal berikut:

- Contoh membuat instans dari kelas `AmazonGlacierClient`.
- Contoh menggunakan operasi `upload` API `ArchiveTransferManager` kelas dari API tingkat tinggi. AWS SDK for Java
- Contohnya menggunakan Wilayah Barat AS (Oregon) (`us-west-2`).

Untuk step-by-step petunjuk tentang cara menjalankan contoh ini, lihatMenjalankan Contoh Java untuk Amazon Glacier Menggunakan Eclipse. Anda harus memperbarui kode seperti yang ditunjukkan dengan nama file arsip yang ingin Anda unggah.

> **Note**
>
> Amazon Glacier menyimpan inventaris semua arsip di brankas Anda. Ketika Anda mengunggah arsip dalam contoh berikut, arsip tidak akan muncul di vault di konsol manajemen hingga inventaris vault telah diperbarui. Pembaruan ini biasanya terjadi sekali sehari.

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.glacier.GlacierClient;
import software.amazon.awssdk.services.glacier.model.UploadArchiveRequest;
import software.amazon.awssdk.services.glacier.model.UploadArchiveResponse;
import software.amazon.awssdk.services.glacier.model.GlacierException;
import java.io.File;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.io.FileInputStream;
import java.io.IOException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class UploadArchive {

    static final int ONE_MB = 1024 * 1024;
```

```
    public static void main(String[] args) {
        final String usage = """

                Usage:   <strPath> <vaultName>\s

                Where:
                   strPath - The path to the archive to upload (for example, C:\\AWS
\\test.pdf).
                   vaultName - The name of the vault.
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String strPath = args[0];
        String vaultName = args[1];
        File myFile = new File(strPath);
        Path path = Paths.get(strPath);
        GlacierClient glacier = GlacierClient.builder()
                .region(Region.US_EAST_1)
                .build();

        String archiveId = uploadContent(glacier, path, vaultName, myFile);
        System.out.println("The ID of the archived item is " + archiveId);
        glacier.close();
    }

    public static String uploadContent(GlacierClient glacier, Path path, String
 vaultName, File myFile) {
        // Get an SHA-256 tree hash value.
        String checkVal = computeSHA256(myFile);
        try {
            UploadArchiveRequest uploadRequest = UploadArchiveRequest.builder()
                    .vaultName(vaultName)
                    .checksum(checkVal)
                    .build();

            UploadArchiveResponse res = glacier.uploadArchive(uploadRequest, path);
            return res.archiveId();

        } catch (GlacierException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
```

```
            System.exit(1);
        }
        return "";
    }

    private static String computeSHA256(File inputFile) {
        try {
            byte[] treeHash = computeSHA256TreeHash(inputFile);
            System.out.printf("SHA-256 tree hash = %s\n", toHex(treeHash));
            return toHex(treeHash);

        } catch (IOException ioe) {
            System.err.format("Exception when reading from file %s: %s", inputFile,
ioe.getMessage());
            System.exit(-1);

        } catch (NoSuchAlgorithmException nsae) {
            System.err.format("Cannot locate MessageDigest algorithm for SHA-256:
%s", nsae.getMessage());
            System.exit(-1);
        }
        return "";
    }

    public static byte[] computeSHA256TreeHash(File inputFile) throws IOException,
            NoSuchAlgorithmException {

        byte[][] chunkSHA256Hashes = getChunkSHA256Hashes(inputFile);
        return computeSHA256TreeHash(chunkSHA256Hashes);
    }

    /**
     * Computes an SHA256 checksum for each 1 MB chunk of the input file. This
     * includes the checksum for the last chunk, even if it's smaller than 1 MB.
     */
    public static byte[][] getChunkSHA256Hashes(File file) throws IOException,
            NoSuchAlgorithmException {

        MessageDigest md = MessageDigest.getInstance("SHA-256");
        long numChunks = file.length() / ONE_MB;
        if (file.length() % ONE_MB > 0) {
            numChunks++;
        }
```

```
        if (numChunks == 0) {
            return new byte[][] { md.digest() };
        }

        byte[][] chunkSHA256Hashes = new byte[(int) numChunks][];
        FileInputStream fileStream = null;

        try {
            fileStream = new FileInputStream(file);
            byte[] buff = new byte[ONE_MB];

            int bytesRead;
            int idx = 0;

            while ((bytesRead = fileStream.read(buff, 0, ONE_MB)) > 0) {
                md.reset();
                md.update(buff, 0, bytesRead);
                chunkSHA256Hashes[idx++] = md.digest();
            }

            return chunkSHA256Hashes;

        } finally {
            if (fileStream != null) {
                try {
                    fileStream.close();
                } catch (IOException ioe) {
                    System.err.printf("Exception while closing %s.\n %s",
file.getName(),
                            ioe.getMessage());
                }
            }
        }
    }

    /**
     * Computes the SHA-256 tree hash for the passed array of 1 MB chunk
     * checksums.
     */
    public static byte[] computeSHA256TreeHash(byte[][] chunkSHA256Hashes)
            throws NoSuchAlgorithmException {

        MessageDigest md = MessageDigest.getInstance("SHA-256");
        byte[][] prevLvlHashes = chunkSHA256Hashes;
```

```
        while (prevLvlHashes.length > 1) {
            int len = prevLvlHashes.length / 2;
            if (prevLvlHashes.length % 2 != 0) {
                len++;
            }

            byte[][] currLvlHashes = new byte[len][];
            int j = 0;
            for (int i = 0; i < prevLvlHashes.length; i = i + 2, j++) {

                // If there are at least two elements remaining.
                if (prevLvlHashes.length - i > 1) {

                    // Calculate a digest of the concatenated nodes.
                    md.reset();
                    md.update(prevLvlHashes[i]);
                    md.update(prevLvlHashes[i + 1]);
                    currLvlHashes[j] = md.digest();

                } else { // Take care of the remaining odd chunk
                    currLvlHashes[j] = prevLvlHashes[i];
                }
            }

            prevLvlHashes = currLvlHashes;
        }

        return prevLvlHashes[0];
    }

    /**
     * Returns the hexadecimal representation of the input byte array
     */
    public static String toHex(byte[] data) {
        StringBuilder sb = new StringBuilder(data.length * 2);
        for (byte datum : data) {
            String hex = Integer.toHexString(datum & 0xFF);

            if (hex.length() == 1) {
                // Append leading zero.
                sb.append("0");
            }
            sb.append(hex);
        }
```

```
        return sb.toString().toLowerCase();
    }
}
```

- Untuk detail API, lihat UploadArchivedi Referensi AWS SDK for Java 2.x API.

Unggah Arsip ke Vault di Amazon Glacier dengan Menggunakan AWS SDK for .NET

Contoh kode C# berikut menggunakan API tingkat tinggi AWS SDK for .NET untuk mengunggah arsip sampel ke vault. Dalam contoh kode, perhatikan hal berikut:

- Contoh ini membuat instance `ArchiveTransferManager` kelas untuk titik akhir Wilayah Amazon Glacier yang ditentukan.
- Contoh kode menggunakan Wilayah AS Barat (Oregon) (`us-west-2`).
- Contoh menggunakan operasi `Upload` API `ArchiveTransferManager` kelas untuk mengunggah arsip Anda. Untuk arsip kecil, operasi ini mengunggah arsip langsung ke Amazon Glacier. Untuk arsip yang lebih besar, operasi ini menggunakan operasi API unggahan multibagian di Amazon Glacier untuk membagi unggahan menjadi beberapa bagian untuk pemulihan kesalahan yang lebih baik, jika ada kesalahan yang ditemui saat streaming data ke Amazon Glacier.

Untuk step-by-step petunjuk tentang cara menjalankan contoh berikut, lihatMenjalankan Contoh Kode. Anda harus memperbarui kode seperti yang ditunjukkan dengan nama brankas Anda dan nama file arsip yang akan diunggah.

> **Note**
>
> Amazon Glacier menyimpan inventaris semua arsip di brankas Anda. Saat Anda mengunggah arsip dalam contoh berikut, arsip tidak akan muncul di vault di konsol manajemen hingga inventaris vault diperbarui. Pembaruan ini biasanya terjadi sekali sehari.

Example— Mengunggah Arsip dengan Menggunakan API Tingkat Tinggi AWS SDK for .NET

```
using System;
```

```
using Amazon.Glacier;
using Amazon.Glacier.Transfer;
using Amazon.Runtime;

namespace glacier.amazon.com.docsamples
{
    class ArchiveUploadHighLevel_GettingStarted
    {
        static string vaultName = "examplevault";
        static string archiveToUpload = "*** Provide file name (with full path) to
 upload ***";

        public static void Main(string[] args)
        {
            try
            {
                var manager = new
 ArchiveTransferManager(Amazon.RegionEndpoint.USWest2);
                // Upload an archive.
                string archiveId = manager.Upload(vaultName, "getting started archive
 test", archiveToUpload).ArchiveId;
                Console.WriteLine("Copy and save the following Archive ID for the next
 step.");
                Console.WriteLine("Archive ID: {0}", archiveId);
                Console.WriteLine("To continue, press Enter");
                Console.ReadKey();
            }
            catch (AmazonGlacierException e) { Console.WriteLine(e.Message); }
            catch (AmazonServiceException e) { Console.WriteLine(e.Message); }
            catch (Exception e) { Console.WriteLine(e.Message); }
            Console.WriteLine("To continue, press Enter");
            Console.ReadKey();
        }
    }
}
```

Langkah 4: Unduh Arsip dari Vault di Amazon Glacier

Pada langkah ini, Anda akan mengunduh arsip contoh yang Anda unggah sebelumnya. Langkah 3: Unggah Arsip ke Vault di Amazon Glacier

> **Important**
>
> Amazon Glacier memang menyediakan konsol. Namun, setiap operasi arsip, seperti mengunggah, mengunduh, atau menghapus, mengharuskan Anda untuk menggunakan (AWS Command Line Interface CLI) atau menulis kode. Tidak ada dukungan konsol untuk operasi arsip. Misalnya, untuk mengunggah data, seperti foto, video, dan dokumen lainnya, Anda harus menggunakan AWS CLI atau menulis kode untuk membuat permintaan, dengan menggunakan REST API secara langsung atau dengan menggunakan AWS SDKs. Untuk menginstal AWS CLI, lihat AWS Command Line Interface. Untuk informasi selengkapnya tentang penggunaan Amazon Glacier dengan, AWS CLI lihat Referensi AWS CLI untuk Amazon Glacier. Untuk contoh penggunaan untuk mengunggah arsip AWS CLI ke Amazon Glacier, lihat Menggunakan Amazon Glacier with the. AWS Command Line Interface

Secara umum, mengambil data Anda dari Amazon Glacier adalah proses dua langkah:

1. Mulai tugas pengambilan.
2. Setelah pekerjaan selesai, unduh byte data.

Untuk mengambil arsip dari Amazon Glacier, Anda pertama kali memulai pekerjaan. Setelah pekerjaan selesai, Anda mengunduh data. Untuk informasi selengkapnya tentang pengambilan arsip, lihat Mengambil Arsip Amazon Glacier.

Waktu akses permintaan Anda bergantung pada opsi pengambilan yang Anda pilih: Pengambilan Dipercepat, Standar, atau Massal. Untuk semua kecuali arsip terbesar (250 MB+), arsip yang diakses dengan menggunakan Expedited retrievals biasanya tersedia dalam waktu 1-5 menit. Arsip yang diambil dengan menggunakan pengambilan Standar biasanya tersedia antara 3-5 jam. Pengambilan massal biasanya tersedia dalam waktu 5-12 jam. Untuk informasi selengkapnya tentang berbagai opsi pengambilan, lihat FAQ Amazon Glacier. Untuk informasi tentang biaya pengambilan data, lihat halaman harga Amazon Glacier.

Contoh kode yang ditampilkan dalam topik berikut memulai pekerjaan, menunggu sampai selesai, dan kemudian mengunduh data arsip.

Topik

- Unduh Arsip dari Vault di Amazon Glacier dengan Menggunakan AWS SDK for Java
- Unduh Arsip dari Vault di Amazon Glacier dengan Menggunakan AWS SDK for .NET

Unduh Arsip dari Vault di Amazon Glacier dengan Menggunakan AWS SDK for Java

Contoh kode Java berikut menggunakan API tingkat tinggi AWS SDK for Java untuk mengunduh arsip yang Anda unggah di langkah sebelumnya. Dalam contoh kode, perhatikan hal berikut:

- Contoh membuat instans dari kelas `AmazonGlacierClient`.
- Kode ini menggunakan Wilayah US West (Oregon) (`us-west-2`) untuk mencocokkan lokasi tempat Anda membuat vault di Langkah 2: Buat Vault di Amazon Glacier.
- Contoh menggunakan operasi `download` API `ArchiveTransferManager` kelas dari API tingkat tinggi. AWS SDK for Java Contoh ini membuat topik Amazon Simple Notification Service (Amazon SNS), dan antrean Amazon Simple Queue Service (Amazon SQS) yang berlangganan topik tersebut. Jika Anda membuat pengguna admin AWS Identity and Access Management (IAM) seperti yang diinstruksikanLangkah 1: Sebelum Anda Memulai dengan Amazon Glacier, pengguna Anda memiliki izin IAM yang diperlukan untuk pembuatan dan penggunaan topik Amazon SNS dan antrean Amazon SQS.

Untuk step-by-step petunjuk tentang cara menjalankan contoh ini, lihatMenjalankan Contoh Java untuk Amazon Glacier Menggunakan Eclipse. Anda harus memperbarui kode seperti yang ditunjukkan dengan ID arsip file yang Anda unggah. Langkah 3: Unggah Arsip ke Vault di Amazon Glacier

Example— Mengunduh Arsip dengan Menggunakan AWS SDK for Java

```
import java.io.File;
import java.io.IOException;

import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.glacier.AmazonGlacierClient;
import com.amazonaws.services.glacier.transfer.ArchiveTransferManager;
import com.amazonaws.services.sns.AmazonSNSClient;
import com.amazonaws.services.sqs.AmazonSQSClient;

public class AmazonGlacierDownloadArchive_GettingStarted {
    public static String vaultName = "examplevault";
    public static String archiveId = "*** provide archive ID ***";
    public static String downloadFilePath  = "*** provide location to download archive
 ***";
```

```
    public static AmazonGlacierClient glacierClient;
    public static AmazonSQSClient sqsClient;
    public static AmazonSNSClient snsClient;

    public static void main(String[] args) throws IOException {


     ProfileCredentialsProvider credentials = new ProfileCredentialsProvider();

        glacierClient = new AmazonGlacierClient(credentials);
        sqsClient = new AmazonSQSClient(credentials);
        snsClient = new AmazonSNSClient(credentials);

        glacierClient.setEndpoint("glacier.us-west-2.amazonaws.com");
        sqsClient.setEndpoint("sqs.us-west-2.amazonaws.com");
        snsClient.setEndpoint("sns.us-west-2.amazonaws.com");

        try {
            ArchiveTransferManager atm = new ArchiveTransferManager(glacierClient,
 sqsClient, snsClient);

            atm.download(vaultName, archiveId, new File(downloadFilePath));

        } catch (Exception e)
        {
            System.err.println(e);
        }
    }
}
```

Unduh Arsip dari Vault di Amazon Glacier dengan Menggunakan AWS SDK for .NET

Contoh kode C # berikut menggunakan API tingkat tinggi AWS SDK for .NET untuk mengunduh arsip yang Anda unggah sebelumnya. Unggah Arsip ke Vault di Amazon Glacier dengan Menggunakan AWS SDK for .NET Dalam contoh kode, perhatikan hal berikut:

- Contoh ini membuat instance `ArchiveTransferManager` kelas untuk titik akhir Wilayah Amazon Glacier yang ditentukan.
- Contoh kode ini menggunakan Wilayah US West (Oregon) (`us-west-2`) untuk mencocokkan lokasi tempat Anda membuat vault sebelumnya di Langkah 2: Buat Vault di Amazon Glacier.

- Contoh menggunakan operasi `Download` API `ArchiveTransferManager` kelas untuk mengunduh arsip Anda. Contoh ini membuat topik Amazon Simple Notification Service (Amazon SNS), dan antrean Amazon Simple Queue Service (Amazon SQS) yang berlangganan topik tersebut. Jika Anda membuat pengguna admin AWS Identity and Access Management (IAM) seperti yang diinstruksikanLangkah 1: Sebelum Anda Memulai dengan Amazon Glacier, pengguna Anda memiliki izin IAM yang diperlukan untuk pembuatan dan penggunaan topik Amazon SNS dan antrean Amazon SQS.
- Contoh tersebut selanjutnya memulai tugas pengambilan arsip dan melakukan polling antrean untuk arsip yang akan tersedia. Ketika arsip tersedia, unduhan dimulai. Untuk informasi tentang waktu pengambilan, lihat Opsi Pengambilan Arsip.

Untuk step-by-step petunjuk tentang cara menjalankan contoh ini, lihatMenjalankan Contoh Kode. Anda harus memperbarui kode seperti yang ditunjukkan dengan ID arsip file yang Anda unggah. Langkah 3: Unggah Arsip ke Vault di Amazon Glacier

Example— Unduh Arsip dengan Menggunakan API Tingkat Tinggi dari AWS SDK for .NET

```
using System;
using Amazon.Glacier;
using Amazon.Glacier.Transfer;
using Amazon.Runtime;

namespace glacier.amazon.com.docsamples
{
    class ArchiveDownloadHighLevel_GettingStarted
    {
        static string vaultName = "examplevault";
        static string archiveId = "*** Provide archive ID ***";
        static string downloadFilePath = "*** Provide the file name and path to where
 to store the download ***";

        public static void Main(string[] args)
        {
            try
            {
                var manager = new
 ArchiveTransferManager(Amazon.RegionEndpoint.USWest2);

                var options = new DownloadOptions();
```

```
                options.StreamTransferProgress +=
 ArchiveDownloadHighLevel_GettingStarted.progress;
                // Download an archive.
                Console.WriteLine("Intiating the archive retrieval job and then polling
 SQS queue for the archive to be available.");
                Console.WriteLine("Once the archive is available, downloading will
 begin.");
                manager.Download(vaultName, archiveId, downloadFilePath, options);
                Console.WriteLine("To continue, press Enter");
                Console.ReadKey();
            }
            catch (AmazonGlacierException e) { Console.WriteLine(e.Message); }
            catch (AmazonServiceException e) { Console.WriteLine(e.Message); }
            catch (Exception e) { Console.WriteLine(e.Message); }
            Console.WriteLine("To continue, press Enter");
            Console.ReadKey();
        }

        static int currentPercentage = -1;
        static void progress(object sender, StreamTransferProgressArgs args)
        {
            if (args.PercentDone != currentPercentage)
            {
                currentPercentage = args.PercentDone;
                Console.WriteLine("Downloaded {0}%", args.PercentDone);
            }
        }
    }
}
```

Langkah 5: Hapus Arsip dari Vault di Amazon Glacier

Pada langkah ini, Anda akan menghapus arsip sampel yang Anda unggah. Langkah 3: Unggah Arsip ke Vault di Amazon Glacier

> **Important**
>
> Anda tidak dapat menghapus arsip dengan menggunakan konsol Amazon Glacier. Setiap operasi arsip, seperti mengunggah, mengunduh, atau menghapus, mengharuskan Anda untuk menggunakan (AWS Command Line Interface CLI) atau menulis kode. Untuk mengunggah data, seperti foto, video, dan dokumen lainnya, Anda harus menggunakan AWS

CLI atau menulis kode untuk membuat permintaan, dengan menggunakan REST API secara langsung atau dengan menggunakan AWS SDKs.
Untuk menginstal AWS CLI, lihat AWS Command Line Interface. Untuk informasi selengkapnya tentang penggunaan Amazon Glacier dengan, AWS CLI lihat Referensi AWS CLI untuk Amazon Glacier. Untuk contoh penggunaan untuk mengunggah arsip AWS CLI ke Amazon Glacier, lihat Menggunakan Amazon Glacier with the. AWS Command Line Interface

Hapus arsip sampel dengan mengikuti salah satu dari ini SDKs atau AWS CLI:

- Menghapus Arsip dari Vault di Amazon Glacier dengan Menggunakan AWS SDK for Java
- Menghapus Arsip dari Vault di Amazon Glacier dengan Menggunakan AWS SDK for .NET
- Hapus Arsip di Amazon Glacier dengan Menggunakan AWS CLI

Bagian Terkait

- Langkah 3: Unggah Arsip ke Vault di Amazon Glacier
- Menghapus Arsip di Amazon Glacier

Menghapus Arsip dari Vault di Amazon Glacier dengan Menggunakan AWS SDK for Java

Contoh kode berikut menggunakan AWS SDK for Java untuk menghapus arsip. Dalam kode, perhatikan hal berikut:

- Objek `DeleteArchiveRequest` menjelaskan permintaan hapus, termasuk nama vault lokasi arsip dan ID arsip.
- Operasi `deleteArchive` API mengirimkan permintaan ke Amazon Glacier untuk menghapus arsip.
- Contohnya menggunakan Wilayah Barat AS (Oregon) (`us-west-2`).

Untuk step-by-step petunjuk tentang cara menjalankan contoh ini, lihatMenjalankan Contoh Java untuk Amazon Glacier Menggunakan Eclipse. Anda harus memperbarui kode seperti yang

ditunjukkan dengan ID arsip file yang Anda unggah. Langkah 3: Unggah Arsip ke Vault di Amazon Glacier

Example— Menghapus Arsip dengan Menggunakan AWS SDK for Java

```
import java.io.IOException;

import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.glacier.AmazonGlacierClient;
import com.amazonaws.services.glacier.model.DeleteArchiveRequest;

public class AmazonGlacierDeleteArchive_GettingStarted {

    public static String vaultName = "examplevault";
    public static String archiveId = "*** provide archive ID***";
    public static AmazonGlacierClient client;

    public static void main(String[] args) throws IOException {


     ProfileCredentialsProvider credentials = new ProfileCredentialsProvider();

        client = new AmazonGlacierClient(credentials);
        client.setEndpoint("https://glacier.us-west-2.amazonaws.com/");

        try {

            // Delete the archive.
            client.deleteArchive(new DeleteArchiveRequest()
                .withVaultName(vaultName)
                .withArchiveId(archiveId));

            System.out.println("Deleted archive successfully.");

        } catch (Exception e) {
            System.err.println("Archive not deleted.");
            System.err.println(e);
        }
    }
}
```

Menghapus Arsip dari Vault di Amazon Glacier dengan Menggunakan AWS SDK for .NET

Contoh kode C # berikut menggunakan API tingkat tinggi AWS SDK for .NET untuk menghapus arsip yang Anda unggah di langkah sebelumnya. Dalam contoh kode, perhatikan hal berikut:

- Contoh ini membuat instance `ArchiveTransferManager` kelas untuk titik akhir Wilayah Amazon Glacier yang ditentukan.
- Contoh kode menggunakan Wilayah AS Barat (Oregon) (`us-west-2`).
- Contoh menggunakan operasi `Delete` API dari `ArchiveTransferManager` kelas yang disediakan sebagai bagian dari API tingkat tinggi. AWS SDK for .NET

Untuk step-by-step petunjuk tentang cara menjalankan contoh ini, lihatMenjalankan Contoh Kode. Anda harus memperbarui kode seperti yang ditunjukkan dengan ID arsip file yang Anda unggah. Langkah 3: Unggah Arsip ke Vault di Amazon Glacier

Example— Menghapus Arsip dengan Menggunakan API Tingkat Tinggi AWS SDK for .NET

```
using System;
using Amazon.Glacier;
using Amazon.Glacier.Transfer;
using Amazon.Runtime;

namespace glacier.amazon.com.docsamples
{
  class ArchiveDeleteHighLevel_GettingStarted
  {
    static string vaultName = "examplevault";
    static string archiveId = "*** Provide archive ID ***";

    public static void Main(string[] args)
    {
      try
      {
        var manager = new ArchiveTransferManager(Amazon.RegionEndpoint.USWest2);
        manager.DeleteArchive(vaultName, archiveId);
      }
      catch (AmazonGlacierException e) { Console.WriteLine(e.Message); }
      catch (AmazonServiceException e) { Console.WriteLine(e.Message); }
      catch (Exception e) { Console.WriteLine(e.Message); }
```

```
      Console.WriteLine("To continue, press Enter");
      Console.ReadKey();
    }
  }
}
```

Hapus Arsip di Amazon Glacier dengan Menggunakan AWS CLI

Anda dapat menghapus arsip di Amazon Glacier dengan menggunakan AWS Command Line Interface ().AWS CLI

Topik

- (Prasyarat) Menyiapkan AWS CLI
- Contoh: Menghapus Arsip dengan Menggunakan AWS CLI

(Prasyarat) Menyiapkan AWS CLI

1. Unduh dan konfigurasikan AWS CLI. Untuk melakukannya, lihat topik berikut di Panduan Pengguna AWS Command Line Interface :

   Memasang AWS Command Line Interface

   Mengkonfigurasi AWS Command Line Interface

2. Verifikasi AWS CLI pengaturan Anda dengan memasukkan perintah berikut pada prompt perintah. Perintah ini tidak memberikan kredensial secara eksplisit, sehingga kredensial profil default digunakan.

   - Coba gunakan perintah bantuan.

     ```
     aws help
     ```

   - Untuk mendapatkan daftar brankas Amazon Glacier pada akun yang dikonfigurasi, gunakan perintah. `list-vaults` Ganti *123456789012* dengan Akun AWS ID Anda.

     ```
     aws glacier list-vaults --account-id 123456789012
     ```

   - Untuk melihat data konfigurasi saat ini untuk AWS CLI, gunakan `aws configure list` perintah.

```
aws configure list
```

Contoh: Menghapus Arsip dengan Menggunakan AWS CLI

1. Gunakan perintah `initiate-job` untuk memulai tugas pengambilan inventaris. Untuk informasi selengkapnya tentang `initiate-job` perintah, lihat Memulai Job.

   ```
   aws glacier initiate-job --vault-name awsexamplevault --account-id 111122223333 --
   job-parameters "{\"Type\": \"inventory-retrieval\"}"
   ```

   Keluaran yang diharapkan

   ```
   {
       "location": "/111122223333/vaults/awsexamplevault/jobs/*** jobid ***",
       "jobId": "*** jobid ***"
   }
   ```

2. Gunakan `describe-job` perintah untuk memeriksa status pekerjaan pengambilan sebelumnya. Untuk informasi selengkapnya tentang `describe-job` perintah, lihat Menjelaskan Pekerjaan.

   ```
   aws glacier describe-job --vault-name awsexamplevault --account-id 111122223333 --
   job-id *** jobid ***
   ```

   Keluaran yang diharapkan

   ```
   {
       "InventoryRetrievalParameters": {
           "Format": "JSON"
       },
       "VaultARN": "*** vault arn ***",
       "Completed": false,
       "JobId": "*** jobid ***",
       "Action": "InventoryRetrieval",
       "CreationDate": "*** job creation date ***",
       "StatusCode": "InProgress"
   }
   ```

3. Tunggu sampai pekerjaan selesai.

Anda harus menunggu hingga output tugas siap diunduh. Jika Anda menyetel konfigurasi notifikasi di vault atau menetapkan topik Simple Notification Service Amazon (Amazon SNS) saat memulai pekerjaan, Amazon Glacier akan mengirimkan pesan ke topik tersebut setelah menyelesaikan pekerjaan.

Anda dapat menetapkan konfigurasi notifikasi untuk peristiwa tertentu di vault. Untuk informasi selengkapnya, lihat Mengkonfigurasi Pemberitahuan Vault di Amazon Glacier. Amazon Glacier mengirim pesan ke topik Amazon SNS yang ditentukan kapan saja peristiwa tertentu terjadi.

4. Saat pekerjaan selesai, gunakan `get-job-output` perintah untuk mengunduh pekerjaan pengambilan ke file`output.json`. Untuk informasi selengkapnya tentang `get-job-output` perintah, lihat Mendapatkan Keluaran Pekerjaan.

```
aws glacier get-job-output --vault-name awsexamplevault --account-id 111122223333
 --job-id *** jobid *** output.json
```

Perintah ini menghasilkan file dengan bidang berikut.

```
{
"VaultARN":"arn:aws:glacier:region:111122223333:vaults/awsexamplevault",
"InventoryDate":"*** job completion date ***",
"ArchiveList":[{
{"ArchiveId":"*** archiveid ***",
"ArchiveDescription":"*** archive description (if set) ***",
"CreationDate":"*** archive creation date ***",
"Size":"*** archive size (in bytes) ***",
"SHA256TreeHash":"*** archive hash ***"
}],
"ArchiveId": 123456789

}
```

5. Gunakan perintah `delete-archive` untuk menghapus setiap arsip dari vault sampai tidak ada yang tersisa.

```
aws glacier delete-archive --vault-name awsexamplevault --account-id 111122223333
 --archive-id="*** archiveid ***"
```

Untuk informasi selengkapnya tentang `delete-archive` perintah, lihat Menghapus Arsip.

Langkah 6: Hapus Vault di Amazon Glacier

Vault adalah kontainer untuk menyimpan arsip. Untuk menghapus brankas Amazon Glacier, Anda harus terlebih dahulu menghapus semua arsip yang ada di brankas pada inventaris terakhir yang dihitung Amazon Glacier.

Anda dapat menghapus vault secara terprogram atau dengan menggunakan konsol Amazon Glacier. Untuk informasi tentang menghapus vault secara terprogram, lihat Menghapus Vault di Amazon Glacier.

> **Important**
>
> Jika Anda mengunggah arsip ke brankas atau menghapus arsip dari brankas dalam 24 jam terakhir, Anda harus menunggu hingga inventaris vault terakhir diperbarui untuk mencerminkan informasi terbaru. Amazon Glacier menyiapkan inventaris untuk setiap lemari besi secara berkala, setiap 24 jam.

Untuk menghapus vault kosong

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Glacier di rumah. https://console.aws.amazon.com/glacier/
2. Dari menu Pilih Wilayah, pilih vault yang ingin Anda hapus. AWS Region

   Dalam latihan memulai ini, lemari besi contoh Anda ada di Wilayah AS Barat (Oregon).
3. Pilih tombol opsi di sebelah brankas kosong yang ingin Anda hapus. Jika lemari besi tidak kosong, Anda harus menghapus semua arsip sebelum menghapus brankas. Untuk informasi selengkapnya, lihat Menghapus Arsip di Amazon Glacier.

   > **Important**
   >
   > Menghapus brankas tidak dapat dibatalkan.
4. Pilih Hapus.
5. Kotak dialog Delete vault muncul. Pilih Hapus.

Untuk menghapus vault yang tidak kosong

1. Jika Anda menghapus brankas yang tidak kosong, Anda harus menghapus semua arsip yang ada terlebih dahulu sebelum menghapus vault. Anda dapat melakukan ini dengan menulis kode untuk membuat permintaan hapus arsip dengan menggunakan REST API AWS SDK for Java,, AWS SDK for .NET atau AWS CLI. Untuk informasi tentang menghapus arsip, lihat Langkah 5: Hapus Arsip dari Vault di Amazon Glacier.
2. Setelah brankas kosong, ikuti langkah-langkah untuk menghapus brankas kosong di prosedur sebelumnya.

Dari Sini, Ke Mana Lagi?

Sekarang setelah Anda mencoba latihan memulai, Anda dapat menjelajahi bagian berikut untuk mempelajari lebih lanjut tentang Amazon Glacier.

- Bekerja dengan Vaults di Amazon Glacier
- Bekerja dengan Arsip di Amazon Glacier

Bekerja dengan Vaults di Amazon Glacier

Vault adalah kontainer untuk menyimpan arsip. Saat membuat vault, Anda menentukan nama vault dan AWS Region tempat Anda ingin membuat vault. Untuk daftar yang Wilayah AWS didukung oleh Amazon Glacier, lihat titik akhir dan kuota Amazon Glacier di Referensi Umum.AWS

Anda dapat menyimpan arsip dalam jumlah yang tidak terbatas di vault.

> **Important**
>
> Amazon Glacier memang menyediakan konsol. Namun, setiap operasi arsip, seperti mengunggah, mengunduh, atau menghapus, mengharuskan Anda untuk menggunakan AWS Command Line Interface (AWS CLI) atau menulis kode. Tidak ada dukungan konsol untuk operasi arsip. Misalnya, untuk mengunggah data, seperti foto, video, dan dokumen lainnya, Anda harus menggunakan AWS CLI atau menulis kode untuk membuat permintaan, dengan menggunakan REST API secara langsung atau dengan menggunakan AWS SDKs. Untuk menginstal AWS CLI, lihat AWS Command Line Interface. Untuk informasi selengkapnya tentang penggunaan Amazon Glacier dengan, lihat AWS CLIAWS CLI Referensi untuk Amazon Glacier. Untuk contoh penggunaan untuk mengunggah arsip AWS CLI ke Amazon Glacier, lihat Menggunakan Amazon Glacier with the. AWS Command Line Interface

Topik

- Operasi Vault di Amazon Glacier
- Membuat Vault di Amazon Glacier
- Mengambil Metadata Vault di Amazon Glacier
- Mengunduh Inventaris Vault di Amazon Glacier
- Mengkonfigurasi Pemberitahuan Vault di Amazon Glacier
- Menghapus Vault di Amazon Glacier
- Menandai Brankas Amazon Glacier Anda
- Kunci Gletser Amazon Glacier

Operasi Vault di Amazon Glacier

Amazon Glacier mendukung berbagai operasi lemari besi. Operasi Vault khusus untuk khusus Wilayah AWS. Dengan kata lain, saat Anda membuat brankas, Anda membuatnya secara spesifik AWS Region. Saat Anda mencantumkan vault, Amazon Glacier mengembalikan daftar vault dari yang Anda tentukan AWS Region dalam permintaan.

Membuat dan Menghapus Vault

Sebuah Akun AWS dapat membuat hingga 1.000 brankas per. AWS RegionUntuk daftar yang Wilayah AWS didukung oleh Amazon Glacier, lihat titik akhir dan kuota Amazon Glacier di Referensi Umum.AWS

Anda dapat menghapus brankas hanya jika tidak ada arsip di brankas pada inventaris terakhir yang dihitung Amazon Glacier dan jika belum ada penulisan ke brankas sejak inventaris terakhir.

> **Note**
>
> Amazon Glacier menyiapkan inventaris untuk setiap lemari besi secara berkala, setiap 24 jam. Karena inventaris mungkin tidak mencerminkan informasi terbaru, Amazon Glacier memastikan bahwa lemari besi memang kosong dengan memeriksa apakah ada operasi penulisan sejak inventaris brankas terakhir.

Untuk informasi selengkapnya, lihat Membuat Vault di Amazon Glacier dan Menghapus Vault di Amazon Glacier.

Mengambil Metadata Vault

Anda dapat mengambil informasi vault seperti tanggal pembuatan vault, jumlah arsip di vault, dan ukuran total semua arsip di vault. Amazon Glacier menyediakan panggilan API bagi Anda untuk mengambil informasi ini untuk brankas tertentu atau semua brankas tertentu di akun Anda. AWS Region Untuk informasi selengkapnya, lihat Mengambil Metadata Vault di Amazon Glacier.

Mengunduh Inventaris Vault

Inventaris vault mengacu pada daftar arsip di brankas. Untuk setiap arsip dalam daftar, inventaris menyediakan informasi arsip, seperti ID arsip, tanggal pembuatan, dan ukuran. Amazon Glacier

memperbarui inventaris brankas sekali sehari, dimulai pada hari arsip pertama diunggah ke lemari besi. Inventaris vault harus ada untuk Anda agar dapat mengunduhnya.

Mengunduh inventaris vault adalah operasi asinkron. Anda harus terlebih dahulu memulai tugas untuk mengunduh inventaris. Setelah menerima permintaan pekerjaan, Amazon Glacier menyiapkan inventaris Anda untuk diunduh. Setelah pekerjaan selesai, Anda dapat mengunduh data inventaris.

Mengingat sifat asinkron pekerjaan, Anda dapat menggunakan notifikasi Amazon Simple Notification Service (Amazon SNS) untuk memberi tahu Anda saat pekerjaan selesai. Anda dapat menentukan topik Amazon SNS untuk setiap permintaan tugas individu atau mengonfigurasi vault Anda untuk mengirim notifikasi ketika peristiwa vault tertentu terjadi.

Amazon Glacier menyiapkan inventaris untuk setiap lemari besi secara berkala, setiap 24 jam. Jika tidak ada penambahan arsip atau penghapusan vault sejak inventaris terakhir, tanggal inventaris tidak diperbarui.

Saat Anda memulai pekerjaan untuk inventaris vault, Amazon Glacier mengembalikan inventaris terakhir yang dihasilkannya, yang merupakan point-in-time snapshot dan bukan data real-time. Anda mungkin merasa mengambil inventaris vault untuk setiap unggahan arsip tidak berguna. Namun, misalkan Anda memelihara database di sisi klien yang berisi metadata yang terkait dengan arsip yang Anda unggah ke Amazon Glacier. Dalam hal ini, Anda mungkin menemukan inventaris vault berguna untuk merekonsiliasi informasi dalam database Anda dengan inventaris vault yang sebenarnya.

Untuk informasi selengkapnya tentang mengambil inventaris vault, lihat Mengunduh Inventaris Vault di Amazon Glacier.

Mengonfigurasi Notifikasi Vault

Mengambil apa pun dari Amazon Glacier, seperti arsip dari lemari besi atau inventaris lemari besi, adalah proses dua langkah. Pertama, Anda memulai pekerjaan. Setelah pekerjaan selesai, Anda mengunduh outputnya. Untuk mempelajari kapan pekerjaan Anda selesai, Anda dapat menggunakan notifikasi Amazon Glacier. Amazon Glacier mengirimkan pesan notifikasi ke topik Simple Notification Service Amazon (Amazon SNS) yang Anda berikan.

Anda dapat mengonfigurasi notifikasi di vault dan mengidentifikasi peristiwa vault serta topik Amazon SNS yang diberi tahu ketika peristiwa terjadi. Kapan saja peristiwa vault terjadi, Amazon Glacier mengirimkan pemberitahuan ke topik Amazon SNS yang ditentukan. Lihat informasi yang lebih lengkap di Mengkonfigurasi Pemberitahuan Vault di Amazon Glacier.

Membuat Vault di Amazon Glacier

Membuat vault menambahkan vault ke serangkaian vault di akun Anda. An Akun AWS dapat membuat hingga 1.000 brankas per AWS Wilayah. Untuk daftar AWS Wilayah yang didukung oleh Amazon Glacier (Amazon Glacier), lihat Wilayah dan Titik Akhir dalam Referensi Umum.AWS

Ketika Anda membuat vault, Anda harus memberi nama vault. Berikut adalah persyaratan penamaan vault:

- Panjang nama dapat antara 1 dan 255 karakter.
- Karakter yang diperbolehkan adalah a–z, A–Z, 0–9, '_' (garis bawah), '-' (tanda hubung), dan '.' (titik).

Nama Vault harus unik dalam akun dan AWS Wilayah tempat vault dibuat. Artinya, akun dapat membuat brankas dengan nama yang sama di Wilayah yang berbeda tetapi tidak di AWS Wilayah yang sama AWS .

Topik

- Membuat Vault di Amazon Glacier Menggunakan AWS SDK for Java
- Membuat Vault di Amazon Glacier Menggunakan AWS SDK for .NET
- Membuat Vault di Amazon Glacier Menggunakan REST API
- Membuat Vault Menggunakan Konsol Amazon Glacier
- Membuat Vault di Amazon Glacier Menggunakan AWS Command Line Interface

Membuat Vault di Amazon Glacier Menggunakan AWS SDK for Java

API tingkat rendah menyediakan metode untuk semua operasi vault, termasuk membuat dan menghapus vault, mendapatkan deskripsi vault, dan mendapatkan daftar vault yang dibuat secara spesifik. AWS Region Berikut adalah langkah-langkah untuk membuat vault menggunakan AWS SDK for Java.

1. Buat instans dari kelas `AmazonGlacierClient` (klien).

   Anda perlu menentukan AWS Region di mana Anda ingin membuat lemari besi. Semua operasi yang Anda lakukan menggunakan klien ini berlaku untuk itu AWS Region.

2. Berikan informasi permintaan dengan membuat instans kelas `CreateVaultRequest`.

   Amazon Glacier (Amazon Glacier) mengharuskan Anda untuk memberikan nama vault dan ID akun Anda. Jika Anda tidak memberikan ID akun, ID akun yang terkait dengan kredensial yang Anda berikan untuk menandatangani permintaan akan digunakan. Untuk informasi selengkapnya, lihat Menggunakan AWS SDK for Java dengan Amazon Glacier.
3. Jalankan metode `createVault` dengan menyediakan objek permintaan sebagai parameter.

   Respons pengembalian Amazon Glacier tersedia di objek. `CreateVaultResult`

Potongan kode Java berikut menggambarkan langkah-langkah sebelumnya. Potongan membuat vault di Wilayah `us-west-2`. `Location`Cetaknya adalah URI relatif dari vault yang menyertakan ID akun Anda, nama AWS Region, dan vault.

```
AmazonGlacierClient client = new AmazonGlacierClient(credentials);
client.setEndpoint("https://glacier.us-west-2.amazonaws.com");

CreateVaultRequest request = new CreateVaultRequest()
    .withVaultName("*** provide vault name ***");
CreateVaultResult result = client.createVault(request);

System.out.println("Created vault successfully: " + result.getLocation());
```

> **Note**
>
> Untuk informasi tentang REST API yang mendasarinya, lihat Membuat Vault (PUT vault).

Contoh: Membuat Vault Menggunakan AWS SDK for Java

Contoh kode Java berikut membuat vault di `us-west-2` Wilayah (untuk informasi selengkapnya Wilayah AWS, lihatMengakses Amazon Glacier). Selain itu, contoh kode mengambil informasi vault, mencantumkan semua vault yang sama AWS Region, dan kemudian menghapus vault yang dibuat.

Untuk step-by-step petunjuk tentang cara menjalankan contoh berikut, lihatMenjalankan Contoh Java untuk Amazon Glacier Menggunakan Eclipse.

Example

```
import java.io.IOException;
```

```
import java.util.List;

import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.glacier.AmazonGlacierClient;
import com.amazonaws.services.glacier.model.CreateVaultRequest;
import com.amazonaws.services.glacier.model.CreateVaultResult;
import com.amazonaws.services.glacier.model.DeleteVaultRequest;
import com.amazonaws.services.glacier.model.DescribeVaultOutput;
import com.amazonaws.services.glacier.model.DescribeVaultRequest;
import com.amazonaws.services.glacier.model.DescribeVaultResult;
import com.amazonaws.services.glacier.model.ListVaultsRequest;
import com.amazonaws.services.glacier.model.ListVaultsResult;


public class AmazonGlacierVaultOperations {

    public static AmazonGlacierClient client;

    public static void main(String[] args) throws IOException {

     ProfileCredentialsProvider credentials = new ProfileCredentialsProvider();

        client = new AmazonGlacierClient(credentials);
        client.setEndpoint("https://glacier.us-east-1.amazonaws.com/");

        String vaultName = "examplevaultfordelete";

        try {
            createVault(client, vaultName);
            describeVault(client, vaultName);
            listVaults(client);
            deleteVault(client, vaultName);

        } catch (Exception e) {
            System.err.println("Vault operation failed." + e.getMessage());
        }
    }

    private static void createVault(AmazonGlacierClient client, String vaultName) {
        CreateVaultRequest createVaultRequest = new CreateVaultRequest()
            .withVaultName(vaultName);
        CreateVaultResult createVaultResult = client.createVault(createVaultRequest);
```

```
        System.out.println("Created vault successfully: " +
createVaultResult.getLocation());
    }

    private static void describeVault(AmazonGlacierClient client, String vaultName) {
        DescribeVaultRequest describeVaultRequest = new DescribeVaultRequest()
            .withVaultName(vaultName);
        DescribeVaultResult describeVaultResult  =
client.describeVault(describeVaultRequest);

        System.out.println("Describing the vault: " + vaultName);
        System.out.print(
                "CreationDate: " + describeVaultResult.getCreationDate() +
                "\nLastInventoryDate: " + describeVaultResult.getLastInventoryDate() +
                "\nNumberOfArchives: " + describeVaultResult.getNumberOfArchives() +
                "\nSizeInBytes: " + describeVaultResult.getSizeInBytes() +
                "\nVaultARN: " + describeVaultResult.getVaultARN() +
                "\nVaultName: " + describeVaultResult.getVaultName());
    }

    private static void listVaults(AmazonGlacierClient client) {
        ListVaultsRequest listVaultsRequest = new ListVaultsRequest();
        ListVaultsResult listVaultsResult = client.listVaults(listVaultsRequest);

        List<DescribeVaultOutput> vaultList = listVaultsResult.getVaultList();
        System.out.println("\nDescribing all vaults (vault list):");
        for (DescribeVaultOutput vault : vaultList) {
            System.out.println(
                    "\nCreationDate: " + vault.getCreationDate() +
                    "\nLastInventoryDate: " + vault.getLastInventoryDate() +
                    "\nNumberOfArchives: " + vault.getNumberOfArchives() +
                    "\nSizeInBytes: " + vault.getSizeInBytes() +
                    "\nVaultARN: " + vault.getVaultARN() +
                    "\nVaultName: " + vault.getVaultName());
        }
    }

    private static void deleteVault(AmazonGlacierClient client, String vaultName) {
        DeleteVaultRequest request = new DeleteVaultRequest()
            .withVaultName(vaultName);
        client.deleteVault(request);
        System.out.println("Deleted vault: " + vaultName);
    }
```

```
}
```

Membuat Vault di Amazon Glacier Menggunakan AWS SDK for .NET

Baik level tinggi maupun level rendah yang APIs disediakan oleh Amazon SDK for .NET menyediakan metode untuk membuat vault.

Topik

- Membuat Vault Menggunakan API Tingkat Tinggi AWS SDK for .NET
- Membuat Vault Menggunakan API Tingkat Rendah AWS SDK for .NET

Membuat Vault Menggunakan API Tingkat Tinggi AWS SDK for .NET

Kelas `ArchiveTransferManager` dari API tingkat tinggi menyediakan metode `CreateVault` yang dapat Anda gunakan untuk membuat vault di Wilayah AWS .

Contoh: Operasi Vault Menggunakan API Tingkat Tinggi AWS SDK for .NET

Contoh kode C# berikut membuat dan menghapus vault di Wilayah US West (Oregon). Untuk daftar Wilayah AWS di mana Anda dapat membuat brankas, lihat. Mengakses Amazon Glacier

Untuk step-by-step petunjuk tentang cara menjalankan contoh berikut, lihatMenjalankan Contoh Kode. Anda perlu memperbarui kode seperti yang ditunjukkan dengan nama vault.

Example

```
using System;
using Amazon.Glacier;
using Amazon.Glacier.Transfer;
using Amazon.Runtime;

namespace glacier.amazon.com.docsamples
{
  class VaultCreateDescribeListVaultsDeleteHighLevel
  {
    static string vaultName = "*** Provide vault name ***";

    public static void Main(string[] args)
    {
```

```
      try
      {
          var manager = new ArchiveTransferManager(Amazon.RegionEndpoint.USWest2);
          manager.CreateVault(vaultName);
          Console.WriteLine("Vault created. To delete the vault, press Enter");
          Console.ReadKey();
          manager.DeleteVault(vaultName);
          Console.WriteLine("\nVault deleted. To continue, press Enter");
          Console.ReadKey();
      }
      catch (AmazonGlacierException e) { Console.WriteLine(e.Message); }
      catch (AmazonServiceException e) { Console.WriteLine(e.Message); }
      catch (Exception e) { Console.WriteLine(e.Message); }
      Console.WriteLine("To continue, press Enter");
      Console.ReadKey();
    }
  }
}
```

Membuat Vault Menggunakan API Tingkat Rendah AWS SDK for .NET

API tingkat rendah menyediakan metode untuk semua operasi vault, termasuk membuat dan menghapus vault, mendapatkan deskripsi vault, dan mendapatkan daftar vault yang dibuat secara spesifik. AWS Region Berikut adalah langkah-langkah untuk membuat vault menggunakan AWS SDK for .NET.

1. Buat instans dari kelas `AmazonGlacierClient` (klien).

   Anda perlu menentukan AWS Region di mana Anda ingin membuat lemari besi. Semua operasi yang Anda lakukan menggunakan klien ini berlaku untuk itu AWS Region.

2. Berikan informasi permintaan dengan membuat instans kelas `CreateVaultRequest`.

   Amazon Glacier (Amazon Glacier) mengharuskan Anda untuk memberikan nama vault dan ID akun Anda. Jika Anda tidak memberikan ID akun, ID akun yang terkait dengan kredensial yang Anda berikan untuk menandatangani permintaan akan diambil. Untuk informasi selengkapnya, lihat Menggunakan AWS SDK for .NET dengan Amazon Glacier.

3. Jalankan metode `CreateVault` dengan menyediakan objek permintaan sebagai parameter.

   Respons pengembalian Amazon Glacier tersedia di objek. `CreateVaultResponse`

Contoh: Operasi Vault Menggunakan API Tingkat Rendah AWS SDK for .NET

Contoh kode C# berikut menjelaskan langkah sebelumnya. Contoh ini membuat vault di Wilayah US West (Oregon). Selain itu, contoh kode mengambil informasi vault, mencantumkan semua vault yang sama AWS Region, dan kemudian menghapus vault yang dibuat. Yang `Location` dicetak adalah URI relatif dari vault yang menyertakan ID akun Anda, nama AWS Region, dan vault.

> **Note**
>
> Untuk informasi tentang REST API yang mendasarinya, lihat Membuat Vault (PUT vault).

Untuk step-by-step petunjuk tentang cara menjalankan contoh berikut, lihatMenjalankan Contoh Kode. Anda perlu memperbarui kode seperti yang ditunjukkan dengan nama vault.

Example

```
using System;
using Amazon.Glacier;
using Amazon.Glacier.Model;
using Amazon.Runtime;

namespace glacier.amazon.com.docsamples
{
  class VaultCreateDescribeListVaultsDelete
  {
    static string vaultName = "*** Provide vault name ***";
    static AmazonGlacierClient client;

    public static void Main(string[] args)
    {
       try
      {
         using (client = new AmazonGlacierClient(Amazon.RegionEndpoint.USWest2))
        {
          Console.WriteLine("Creating a vault.");
          CreateAVault();
          DescribeVault();
          GetVaultsList();
          Console.WriteLine("\nVault created. Now press Enter to delete the vault...");
          Console.ReadKey();
          DeleteVault();
```

```
        }
      }
      catch (AmazonGlacierException e) { Console.WriteLine(e.Message); }
      catch (AmazonServiceException e) { Console.WriteLine(e.Message); }
      catch (Exception e) { Console.WriteLine(e.Message); }
      Console.WriteLine("To continue, press Enter");
      Console.ReadKey();
    }

    static void CreateAVault()
    {
      CreateVaultRequest request = new CreateVaultRequest()
      {
        VaultName = vaultName
      };
      CreateVaultResponse response = client.CreateVault(request);
      Console.WriteLine("Vault created: {0}\n", response.Location);
    }

    static void DescribeVault()
    {
      DescribeVaultRequest describeVaultRequest = new DescribeVaultRequest()
      {
        VaultName = vaultName
      };

      DescribeVaultResponse describeVaultResponse =
client.DescribeVault(describeVaultRequest);
      Console.WriteLine("\nVault description...");
      Console.WriteLine(
        "\nVaultName: " + describeVaultResponse.VaultName +
        "\nVaultARN: " + describeVaultResponse.VaultARN +
        "\nVaultCreationDate: " + describeVaultResponse.CreationDate +
        "\nNumberOfArchives: " + describeVaultResponse.NumberOfArchives +
        "\nSizeInBytes: " + describeVaultResponse.SizeInBytes +
        "\nLastInventoryDate: " + describeVaultResponse.LastInventoryDate
        );
    }

    static void GetVaultsList()
    {
      string lastMarker = null;
      Console.WriteLine("\n List of vaults in your account in the specific
region ...");
```

```
      do
      {
        ListVaultsRequest request = new ListVaultsRequest()
        {
          Marker = lastMarker
        };
        ListVaultsResponse response = client.ListVaults(request);

        foreach (DescribeVaultOutput output in response.VaultList)
        {
          Console.WriteLine("Vault Name: {0} \tCreation Date: {1} \t #of archives:
 {2}",
                            output.VaultName, output.CreationDate,
 output.NumberOfArchives);
        }
        lastMarker = response.Marker;
      } while (lastMarker != null);
    }

    static void DeleteVault()
    {
      DeleteVaultRequest request = new DeleteVaultRequest()
      {
        VaultName = vaultName
      };
      DeleteVaultResponse response = client.DeleteVault(request);
    }
  }
}
```

Membuat Vault di Amazon Glacier Menggunakan REST API

Untuk membuat vault menggunakan REST API, lihat Membuat Vault (PUT vault).

Membuat Vault Menggunakan Konsol Amazon Glacier

Untuk membuat vault menggunakan konsol Amazon Glacier (Amazon Glacier)Langkah 2: Buat Vault di Amazon Glacier, lihat di tutorial Memulai.

Membuat Vault di Amazon Glacier Menggunakan AWS Command Line Interface

Ikuti langkah-langkah ini untuk membuat lemari besi di Amazon Glacier (Amazon Glacier) menggunakan (). AWS Command Line Interface AWS CLI

Topik

- (Prasyarat) Menyiapkan AWS CLI
- Contoh: Membuat Vault Menggunakan AWS CLI

(Prasyarat) Menyiapkan AWS CLI

1. Unduh dan konfigurasikan AWS CLI. Untuk melakukannya, lihat topik berikut di Panduan Pengguna AWS Command Line Interface :

   Instalasi AWS Command Line Interface

   Mengkonfigurasi AWS Command Line Interface
2. Verifikasi AWS CLI pengaturan Anda dengan memasukkan perintah berikut pada prompt perintah. Perintah ini tidak memberikan kredensial secara eksplisit, sehingga kredensial profil default digunakan.

   - Coba gunakan perintah bantuan.

     ```
     aws help
     ```

   - Untuk mendapatkan daftar brankas Amazon Glacier pada akun yang dikonfigurasi, gunakan perintah. `list-vaults` Ganti *123456789012* dengan Akun AWS ID Anda.

     ```
     aws glacier list-vaults --account-id 123456789012
     ```

   - Untuk melihat data konfigurasi saat ini untuk AWS CLI, gunakan `aws configure list` perintah.

     ```
     aws configure list
     ```

Contoh: Membuat Vault Menggunakan AWS CLI

1. Gunakan `create-vault` perintah untuk membuat vault bernama *awsexamplevault* di bawah akun*111122223333*.

   ```
   aws glacier create-vault --vault-name awsexamplevault --account-id 111122223333
   ```

   Keluaran yang diharapkan

   ```
   {
       "location": "/111122223333/vaults/awsexamplevault"
   }
   ```

2. Verifikasi pembuatan menggunakan perintah `describe-vault`.

   ```
   aws glacier describe-vault --vault-name awsexamplevault --account-id 111122223333
   ```

Mengambil Metadata Vault di Amazon Glacier

Anda dapat mengambil informasi vault seperti tanggal pembuatan vault, jumlah arsip di vault, dan ukuran total semua arsip di vault. Amazon Glacier (Amazon Glacier) menyediakan panggilan API bagi Anda untuk mengambil informasi ini untuk brankas tertentu atau semua brankas di Wilayah tertentu di akun Anda. AWS

Jika Anda mengambil daftar vault, Amazon Glacier mengembalikan daftar yang diurutkan berdasarkan nilai ASCII dari nama vault. Daftar ini berisi hingga 1.000 vault. Anda harus selalu memeriksa respons untuk penanda tempat untuk melanjutkan daftar; jika tidak ada lagi item, bidang penanda adalah `null`. Anda dapat secara opsional membatasi jumlah vault yang dikembalikan dalam respons. Jika ada lebih banyak vault daripada yang dikembalikan dalam respons, hasilnya adalah pemberian nomor halaman. Anda perlu mengirim permintaan tambahan untuk mengambil sekumpulan vault berikutnya.

Topik

- Mengambil Metadata Vault di Amazon Glacier Menggunakan AWS SDK for Java
- Mengambil Metadata Vault di Amazon Glacier Menggunakan AWS SDK for .NET

- Mengambil Metadata Vault menggunakan REST API
- Mengambil Metadata Vault di Amazon Glacier Menggunakan AWS Command Line Interface

Mengambil Metadata Vault di Amazon Glacier Menggunakan AWS SDK for Java

Topik

- Mengambil Metadata Vault untuk Vault
- Mengambil Metadata Vault untuk Semua Vault dalam Wilayah
- Contoh: Mengambil Metadata Vault Menggunakan Amazon SDK for Java

Mengambil Metadata Vault untuk Vault

Anda dapat mengambil metadata untuk brankas tertentu atau semua brankas di Wilayah tertentu. AWS Berikut adalah langkah-langkah mengambil metadata vault untuk vault tertentu menggunakan API tingkat rendah dari Amazon SDK for Java.

1. Buat instans dari kelas `AmazonGlacierClient` (klien).

   Anda perlu menentukan AWS Wilayah tempat brankas berada. Semua operasi yang Anda lakukan menggunakan klien ini berlaku untuk AWS Wilayah tersebut.
2. Berikan informasi permintaan dengan membuat instans kelas `DescribeVaultRequest`.

   Amazon Glacier (Amazon Glacier) mengharuskan Anda untuk memberikan nama vault dan ID akun Anda. Jika Anda tidak memberikan ID akun, ID akun yang terkait dengan kredensial yang Anda berikan untuk menandatangani permintaan diambil. Untuk informasi selengkapnya, lihat Menggunakan AWS SDK for Java dengan Amazon Glacier.
3. Jalankan metode `describeVault` dengan menyediakan objek permintaan sebagai parameter.

   Informasi metadata vault yang dikembalikan Amazon Glacier tersedia di objek. `DescribeVaultResult`

Potongan kode Java berikut menggambarkan langkah-langkah sebelumnya.

```
DescribeVaultRequest request = new DescribeVaultRequest()
```

```
 .withVaultName("*** provide vault name***");

DescribeVaultResult result = client.describeVault(request);

System.out.print(
        "\nCreationDate: " + result.getCreationDate() +
        "\nLastInventoryDate: " + result.getLastInventoryDate() +
        "\nNumberOfArchives: " + result.getNumberOfArchives() +
        "\nSizeInBytes: " + result.getSizeInBytes() +
        "\nVaultARN: " + result.getVaultARN() +
        "\nVaultName: " + result.getVaultName());
```

> **Note**
>
> Untuk informasi tentang REST API yang mendasarinya, lihat Menjelaskan Vault (GET vault).

Mengambil Metadata Vault untuk Semua Vault dalam Wilayah

Anda juga dapat menggunakan `listVaults` metode ini untuk mengambil metadata untuk semua vault di Wilayah tertentu. AWS

Potongan kode Java berikut mengambil daftar vault di Wilayah `us-west-2`. Permintaan membatasi jumlah vault yang dikembalikan dalam merespons 5. Cuplikan kode kemudian membuat serangkaian `listVaults` panggilan untuk mengambil seluruh daftar vault dari Wilayah. AWS

```
AmazonGlacierClient client;
client.setEndpoint("https://glacier.us-west-2.amazonaws.com/");

String marker = null;
do {
    ListVaultsRequest request = new ListVaultsRequest()
        .withLimit("5")
        .withMarker(marker);
    ListVaultsResult listVaultsResult = client.listVaults(request);

    List<DescribeVaultOutput> vaultList = listVaultsResult.getVaultList();
    marker = listVaultsResult.getMarker();
    for (DescribeVaultOutput vault : vaultList) {
        System.out.println(
```

```
                "\nCreationDate: " + vault.getCreationDate() +
                "\nLastInventoryDate: " + vault.getLastInventoryDate() +
                "\nNumberOfArchives: " + vault.getNumberOfArchives() +
                "\nSizeInBytes: " + vault.getSizeInBytes() +
                "\nVaultARN: " + vault.getVaultARN() +
                "\nVaultName: " + vault.getVaultName());
    }
} while (marker != null);
```

Di segmen kode sebelumnya, jika Anda tidak menentukan `Limit` nilai dalam permintaan, Amazon Glacier mengembalikan hingga 10 vault, seperti yang ditetapkan oleh Amazon Glacier API. Jika ada lebih banyak vault yang dicantumkan, bidang `marker` respons berisi vault Amazon Resource Name (ARN) tempat untuk melanjutkan daftar dengan permintaan baru; jika tidak, bidang `marker` adalah null.

Perhatikan bahwa informasi yang dikembalikan untuk setiap vault dalam daftar sama dengan informasi yang Anda dapatkan dengan memanggil metode `describeVault` untuk vault tertentu.

> **Note**
>
> Metode `listVaults` memanggil REST API yang mendasarinya (lihat Mencantumkan Vault (GET vaults)).

Contoh: Mengambil Metadata Vault Menggunakan Amazon SDK for Java

Untuk contoh kode kerja, lihat Contoh: Membuat Vault Menggunakan AWS SDK for Java. Contoh kode Java membuat vault dan mengambil metadata vault.

Mengambil Metadata Vault di Amazon Glacier Menggunakan AWS SDK for .NET

Topik

- Mengambil Metadata Vault untuk Vault
- Mengambil Metadata Vault untuk Semua Vault dalam Wilayah
- Contoh: Mengambil Metadata Vault Menggunakan API Tingkat Rendah AWS SDK for .NET

Mengambil Metadata Vault untuk Vault

Anda dapat mengambil metadata untuk brankas tertentu atau semua brankas di Wilayah tertentu. AWS Berikut adalah langkah-langkah mengambil metadata vault untuk vault tertentu menggunakan API tingkat rendah dari AWS SDK for .NET.

1. Buat instans dari kelas `AmazonGlacierClient` (klien).

   Anda perlu menentukan AWS Wilayah tempat brankas berada. Semua operasi yang Anda lakukan menggunakan klien ini berlaku untuk AWS Wilayah tersebut.
2. Berikan informasi permintaan dengan membuat instans kelas `DescribeVaultRequest`.

   Amazon Glacier (Amazon Glacier) mengharuskan Anda untuk memberikan nama vault dan ID akun Anda. Jika Anda tidak memberikan ID akun, ID akun yang terkait dengan kredensial yang Anda berikan untuk menandatangani permintaan diambil. Untuk informasi selengkapnya, lihat Menggunakan AWS SDK for .NET dengan Amazon Glacier.
3. Jalankan metode `DescribeVault` dengan menyediakan objek permintaan sebagai parameter.

   Informasi metadata vault yang dikembalikan Amazon Glacier tersedia di objek. `DescribeVaultResult`

Potongan kode C# berikut menggambarkan langkah-langkah sebelumnya. Potongan mengambil informasi metadata dari vault yang ada di Wilayah US West (Oregon).

```
AmazonGlacierClient client;
client = new AmazonGlacierClient(Amazon.RegionEndpoint.USWest2);

DescribeVaultRequest describeVaultRequest = new DescribeVaultRequest()
{
  VaultName = "*** Provide vault name ***"
};
DescribeVaultResponse describeVaultResponse =
 client.DescribeVault(describeVaultRequest);
Console.WriteLine("\nVault description...");
Console.WriteLine(
   "\nVaultName: " + describeVaultResponse.VaultName +
   "\nVaultARN: " + describeVaultResponse.VaultARN +
   "\nVaultCreationDate: " + describeVaultResponse.CreationDate +
   "\nNumberOfArchives: " + describeVaultResponse.NumberOfArchives +
   "\nSizeInBytes: " + describeVaultResponse.SizeInBytes +
```

```
    "\nLastInventoryDate: " + describeVaultResponse.LastInventoryDate
    );
```

> **Note**
>
> Untuk informasi tentang REST API yang mendasarinya, lihat Menjelaskan Vault (GET vault).

Mengambil Metadata Vault untuk Semua Vault dalam Wilayah

Anda juga dapat menggunakan `ListVaults` metode ini untuk mengambil metadata untuk semua vault di Wilayah tertentu. AWS

Potongan kode C# berikut mengambil daftar vault di Wilayah US West (Oregon). Permintaan membatasi jumlah vault yang dikembalikan dalam merespons 5. Cuplikan kode kemudian membuat serangkaian `ListVaults` panggilan untuk mengambil seluruh daftar vault dari Wilayah. AWS

```
AmazonGlacierClient client;
client = new AmazonGlacierClient(Amazon.RegionEndpoint.USWest2);
string lastMarker = null;
Console.WriteLine("\n List of vaults in your account in the specific AWS Region ...");
do
{
  ListVaultsRequest request = new ListVaultsRequest()
  {
    Limit = 5,
    Marker = lastMarker
  };
  ListVaultsResponse response = client.ListVaults(request);

  foreach (DescribeVaultOutput output in response.VaultList)
  {
    Console.WriteLine("Vault Name: {0} \tCreation Date: {1} \t #of archives: {2}",
                      output.VaultName, output.CreationDate, output.NumberOfArchives);
  }
  lastMarker = response.Marker;
} while (lastMarker != null);
```

Di segmen kode sebelumnya, jika Anda tidak menentukan `Limit` nilai dalam permintaan, Amazon Glacier mengembalikan hingga 10 vault, seperti yang ditetapkan oleh Amazon Glacier API.

Perhatikan bahwa informasi yang dikembalikan untuk setiap vault dalam daftar sama dengan informasi yang Anda dapatkan dengan memanggil metode `DescribeVault` untuk vault tertentu.

> **Note**
>
> Metode `ListVaults` memanggil REST API yang mendasarinya (lihat Mencantumkan Vault (GET vaults)).

Contoh: Mengambil Metadata Vault Menggunakan API Tingkat Rendah AWS SDK for .NET

Untuk contoh kode pekerjaan, lihat Contoh: Operasi Vault Menggunakan API Tingkat Rendah AWS SDK for .NET. Contoh kode C# membuat vault dan mengambil metadata vault.

Mengambil Metadata Vault menggunakan REST API

Untuk mencantumkan vault menggunakan REST API, lihat Mencantumkan Vault (GET vaults). Untuk mendeskripsikan satu vault, lihat Menjelaskan Vault (GET vault).

Mengambil Metadata Vault di Amazon Glacier Menggunakan AWS Command Line Interface

Contoh ini menunjukkan cara mengambil informasi vault dan metadata di Amazon Glacier (Amazon Glacier) menggunakan (). AWS Command Line Interface AWS CLI

Topik

- (Prasyarat) Menyiapkan AWS CLI
- Contoh: Mengambil Metadata Vault Menggunakan AWS CLI

(Prasyarat) Menyiapkan AWS CLI

1. Unduh dan konfigurasikan AWS CLI. Untuk melakukannya, lihat topik berikut di Panduan Pengguna AWS Command Line Interface :

   Memasang AWS Command Line Interface

Mengkonfigurasi AWS Command Line Interface

2. Verifikasi AWS CLI pengaturan Anda dengan memasukkan perintah berikut pada prompt perintah. Perintah ini tidak memberikan kredensial secara eksplisit, sehingga kredensial profil default digunakan.

   - Coba gunakan perintah bantuan.

     ```
     aws help
     ```

   - Untuk mendapatkan daftar brankas Amazon Glacier pada akun yang dikonfigurasi, gunakan perintah. `list-vaults` Ganti *123456789012* dengan Akun AWS ID Anda.

     ```
     aws glacier list-vaults --account-id 123456789012
     ```

   - Untuk melihat data konfigurasi saat ini untuk AWS CLI, gunakan `aws configure list` perintah.

     ```
     aws configure list
     ```

Contoh: Mengambil Metadata Vault Menggunakan AWS CLI

- Gunakan `describe-vault` perintah untuk mendeskripsikan vault bernama *awsexamplevault* di bawah akun*111122223333*.

  ```
  aws glacier describe-vault --vault-name awsexamplevault --account-id 111122223333
  ```

Mengunduh Inventaris Vault di Amazon Glacier

Setelah Anda mengunggah arsip pertama Anda ke brankas Anda, Amazon Glacier (Amazon Glacier) secara otomatis membuat inventaris vault dan kemudian memperbaruinya kira-kira sekali sehari. Setelah Amazon Glacier membuat inventaris pertama, biasanya dibutuhkan setengah hari dan hingga sehari sebelum inventaris itu tersedia untuk diambil. Anda dapat mengambil inventaris vault dari Amazon Glacier dengan proses dua langkah berikut:

1. Mulai tugas pengambilan inventaris menggunakan operasi Memulai Tugas (POST jobs).

> **Important**
>
> Kebijakan pengambilan data dapat menyebabkan permintaan tugas pengambilan awal Anda gagal dengan pengecualian `PolicyEnforcedException`. Untuk informasi selengkapnya tentang kebijakan pengambilan data, lihat Kebijakan Pengambilan Data Amazon Glacier. Untuk informasi selengkapnya tentang pengecualian `PolicyEnforcedException`, lihat Respons Kesalahan.

2. Setelah tugas selesai, unduh byte menggunakan operasi Mendapatkan Output Tugas (GET output).

Misalnya, mengambil arsip atau inventaris vault mengharuskan Anda memulai tugas pengambilan terlebih dahulu. Permintaan tugas dijalankan secara asinkron. Ketika Anda memulai tugas pengambilan, Amazon Glacier membuat tugas dan menghasilkan ID tugas dalam respons. Ketika Amazon Glacier menyelesaikan tugas, Anda bisa mendapatkan output tugas, bita arsip, atau data inventaris vault.

Tugas harus selesai sebelum Anda bisa mendapatkan outputnya. Untuk menentukan status tugas, Anda memiliki opsi berikut:

- Tunggu pemberitahuan penyelesaian pekerjaan —Anda dapat menentukan topik Amazon Simple Notification Service (Amazon SNS) tempat Amazon Glacier dapat memposting pemberitahuan setelah pekerjaan selesai. Anda dapat menentukan topik Amazon SNS menggunakan metode berikut:
  - Tentukan topik Amazon SNS per tugas.

    Ketika Anda memulai tugas, Anda dapat secara opsional menentukan topik Amazon SNS.
  - Atur konfigurasi notifikasi di vault.

    Anda dapat menetapkan konfigurasi notifikasi untuk peristiwa tertentu di vault (lihat Mengkonfigurasi Pemberitahuan Vault di Amazon Glacier). Amazon Glacier mengirim pesan ke topik SNS yang ditentukan setiap kali peristiwa tertentu terjadi.

  Jika konfigurasi notifikasi disetel di vault dan Anda juga menentukan topik Amazon SNS saat memulai pekerjaan, Amazon Glacier mengirimkan pesan penyelesaian pekerjaan ke kedua topik tersebut.

Anda dapat mengonfigurasi topik SNS untuk memberi tahu Anda melalui email atau menyimpan pesan di Amazon Simple Queue Service (Amazon SQS) yang dapat dibuat polling oleh aplikasi Anda. Ketika pesan muncul di antrean, Anda dapat memeriksa apakah tugas berhasil diselesaikan, lalu mengunduh output tugas.

- Minta informasi pekerjaan secara eksplisit —Amazon Glacier juga menyediakan deskripsi operasi pekerjaan (Mendeskripsikan Tugas (GET JobID)) yang memungkinkan Anda melakukan polling untuk informasi pekerjaan. Anda dapat mengirim permintaan ini secara berkala untuk mendapatkan informasi tugas. Namun, menggunakan notifikasi Amazon SNS adalah opsi yang disarankan.

> **Note**
>
> Informasi yang Anda dapatkan melalui notifikasi SNS sama dengan informasi yang Anda dapatkan dengan memanggil Jelaskan Tugas.

Topik

- Tentang Inventaris
- Mengunduh Inventaris Vault di Amazon Glacier Menggunakan AWS SDK for Java
- Mengunduh Inventaris Vault di Amazon Glacier Menggunakan AWS SDK for .NET
- Mengunduh Inventaris Vault Menggunakan REST API
- Mengunduh Inventaris Vault di Amazon Glacier Menggunakan AWS Command Line Interface

Tentang Inventaris

Amazon Glacier memperbarui inventaris brankas setidaknya sekali per hari, dimulai pada hari Anda pertama kali mengunggah arsip ke lemari besi. Jika tidak ada penambahan arsip atau penghapusan vault sejak inventaris terakhir, tanggal inventaris tidak diperbarui. Saat Anda memulai pekerjaan untuk inventaris vault, Amazon Glacier mengembalikan inventaris terakhir yang dihasilkannya, yang merupakan point-in-time snapshot dan bukan data real-time. Perhatikan bahwa setelah Amazon Glacier membuat inventaris pertama untuk lemari besi, biasanya dibutuhkan setengah hari dan hingga sehari sebelum inventaris tersebut tersedia untuk diambil.

Anda mungkin merasa mengambil inventaris vault untuk setiap unggahan arsip tidak berguna. Namun, misalkan Anda memelihara database di sisi klien terkait metadata tentang arsip yang Anda

unggah ke Amazon Glacier. Selanjutnya, Anda mungkin merasa inventaris vault berguna untuk mencocokkan informasi, jika perlu, dalam basis data Anda dengan inventaris vault aktual. Anda dapat membatasi jumlah item inventaris yang diambil dengan memfilter tanggal pembuatan arsip atau dengan menetapkan batas. Untuk informasi selengkapnya tentang membatasi pengambilan inventaris, lihat Pengambilan Inventaris Rentang.

Inventaris dapat dikembalikan dalam dua format, nilai yang dipisahkan oleh koma (CSV) atau JSON. Anda secara opsional dapat menentukan format ketika Anda memulai tugas inventaris. Format default adalah JSON. Untuk informasi selengkapnya tentang bidang data yang dikembalikan dalam output tugas inventaris, lihat Isi Respons dari API Dapatkan Output Tugas.

Mengunduh Inventaris Vault di Amazon Glacier Menggunakan AWS SDK for Java

Berikut adalah langkah-langkah untuk mengambil inventaris vault menggunakan API tingkat rendah dari AWS SDK for Java. API tingkat tinggi tidak mendukung pengambilan inventaris vault.

1. Buat instans dari kelas `AmazonGlacierClient` (klien).

   Anda perlu menentukan AWS Wilayah tempat brankas berada. Semua operasi yang Anda lakukan menggunakan klien ini berlaku untuk AWS Wilayah tersebut.
2. Mulai tugas pengambilan inventaris dengan menjalankan metode `initiateJob`.

   Jalankan `initiateJob` dengan memberikan informasi tugas di objek `InitiateJobRequest`.

   > **Note**
   >
   > Perhatikan bahwa jika inventaris belum diselesaikan untuk vault, kesalahan dikembalikan. Amazon Glacier (Amazon Glacier) menyiapkan inventaris untuk setiap lemari besi secara berkala, setiap 24 jam.

   Amazon Glacier mengembalikan ID pekerjaan sebagai tanggapan. Respons tersedia dalam instans dari kelas `InitiateJobResult`.

```
InitiateJobRequest initJobRequest = new InitiateJobRequest()
    .withVaultName("*** provide vault name ***")
```

```
    .withJobParameters(
            new JobParameters()
                .withType("inventory-retrieval")
                .withSNSTopic("*** provide SNS topic ARN ****")
      );

InitiateJobResult initJobResult = client.initiateJob(initJobRequest);
String jobId = initJobResult.getJobId();
```

3. Tunggu hingga tugas selesai.

   Anda harus menunggu hingga output tugas siap diunduh. Jika Anda telah menyetel konfigurasi notifikasi di vault, atau menetapkan topik Amazon Simple Notification Service (Amazon SNS) saat memulai pekerjaan, Amazon Glacier akan mengirimkan pesan ke topik tersebut setelah menyelesaikan pekerjaan.

   Anda juga dapat melakukan polling Amazon Glacier dengan memanggil metode untuk menentukan `describeJob` status penyelesaian pekerjaan. Namun, menggunakan topik Amazon SNS untuk notifikasi adalah pendekatan yang disarankan. Contoh kode yang diberikan di bagian berikut menggunakan Amazon SNS untuk Amazon Glacier untuk mempublikasikan pesan.

4. Unduh output tugas (data inventaris vault) dengan menjalankan metode `getJobOutput`.

   Anda memberikan ID akun, ID tugas, dan nama vault dengan membuat instans kelas `GetJobOutputRequest`. Jika Anda tidak memberikan ID akun, ID akun yang terkait dengan kredensial yang Anda berikan untuk menandatangani permintaan akan digunakan. Untuk informasi selengkapnya, lihat Menggunakan AWS SDK for Java dengan Amazon Glacier.

   Output yang dikembalikan Amazon Glacier tersedia di objek. `GetJobOutputResult`

```
GetJobOutputRequest jobOutputRequest = new GetJobOutputRequest()
        .withVaultName("*** provide vault name ***")
        .withJobId("*** provide job ID ***");
GetJobOutputResult jobOutputResult = client.getJobOutput(jobOutputRequest);
// jobOutputResult.getBody(); provides the output stream.
```

> **Note**
>
> Untuk informasi tentang tugas yang terkait REST API yang mendasarinya, lihat Operasi Tugas.

Contoh: Mengambil inventaris Vault Menggunakan Amazon SDK for Java

Contoh kode Java berikut mengambil inventaris vault untuk vault yang ditentukan.

Contoh tersebut melakukan tugas-tugas berikut:

- Membuat topik Amazon Simple Notification Service (Amazon SNS).

  Amazon Glacier mengirimkan pemberitahuan ke topik ini setelah menyelesaikan pekerjaan.
- Membuat antrean Amazon Simple Queue Service (Amazon SQS).

  Contoh melampirkan kebijakan ke antrean untuk mengizinkan topik Amazon SNS mengirim pesan ke antrean.
- Memulai tugas untuk mengunduh arsip yang ditentukan.

  Dalam permintaan pekerjaan, topik Amazon SNS yang dibuat ditentukan sehingga Amazon Glacier dapat mempublikasikan pemberitahuan ke topik setelah menyelesaikan pekerjaan.
- Memeriksa antrean Amazon SQS untuk pesan yang berisi ID tugas.

  Jika ada pesan, urai JSON dan periksa apakah tugas berhasil diselesaikan. Jika ya, unduh arsipnya.
- Membersihkan dengan menghapus topik Amazon SNS dan antrean Amazon SQS yang dibuat.

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import com.fasterxml.jackson.core.JsonFactory;
```

```
import com.fasterxml.jackson.core.JsonParseException;
import com.fasterxml.jackson.core.JsonParser;
import com.fasterxml.jackson.databind.JsonNode;
import com.fasterxml.jackson.databind.ObjectMapper;

import com.amazonaws.AmazonClientException;
import com.amazonaws.auth.policy.Policy;
import com.amazonaws.auth.policy.Principal;
import com.amazonaws.auth.policy.Resource;
import com.amazonaws.auth.policy.Statement;
import com.amazonaws.auth.policy.Statement.Effect;
import com.amazonaws.auth.policy.actions.SQSActions;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.glacier.AmazonGlacierClient;
import com.amazonaws.services.glacier.model.GetJobOutputRequest;
import com.amazonaws.services.glacier.model.GetJobOutputResult;
import com.amazonaws.services.glacier.model.InitiateJobRequest;
import com.amazonaws.services.glacier.model.InitiateJobResult;
import com.amazonaws.services.glacier.model.JobParameters;
import com.amazonaws.services.sns.AmazonSNSClient;
import com.amazonaws.services.sns.model.CreateTopicRequest;
import com.amazonaws.services.sns.model.CreateTopicResult;
import com.amazonaws.services.sns.model.DeleteTopicRequest;
import com.amazonaws.services.sns.model.SubscribeRequest;
import com.amazonaws.services.sns.model.SubscribeResult;
import com.amazonaws.services.sns.model.UnsubscribeRequest;
import com.amazonaws.services.sqs.AmazonSQSClient;
import com.amazonaws.services.sqs.model.CreateQueueRequest;
import com.amazonaws.services.sqs.model.CreateQueueResult;
import com.amazonaws.services.sqs.model.DeleteQueueRequest;
import com.amazonaws.services.sqs.model.GetQueueAttributesRequest;
import com.amazonaws.services.sqs.model.GetQueueAttributesResult;
import com.amazonaws.services.sqs.model.Message;
import com.amazonaws.services.sqs.model.ReceiveMessageRequest;
import com.amazonaws.services.sqs.model.SetQueueAttributesRequest;


public class AmazonGlacierDownloadInventoryWithSQSPolling {

    public static String vaultName = "*** provide vault name ***";
    public static String snsTopicName = "*** provide topic name ***";
    public static String sqsQueueName = "*** provide queue name ***";
    public static String sqsQueueARN;
    public static String sqsQueueURL;
```

```
    public static String snsTopicARN;
    public static String snsSubscriptionARN;
    public static String fileName = "*** provide file name ***";
    public static String region = "*** region ***";
    public static long sleepTime = 600;
    public static AmazonGlacierClient client;
    public static AmazonSQSClient sqsClient;
    public static AmazonSNSClient snsClient;

    public static void main(String[] args) throws IOException {

     ProfileCredentialsProvider credentials = new ProfileCredentialsProvider();

        client = new AmazonGlacierClient(credentials);
        client.setEndpoint("https://glacier." + region + ".amazonaws.com");
        sqsClient = new AmazonSQSClient(credentials);
        sqsClient.setEndpoint("https://sqs." + region + ".amazonaws.com");
        snsClient = new AmazonSNSClient(credentials);
        snsClient.setEndpoint("https://sns." + region + ".amazonaws.com");

        try {
            setupSQS();

            setupSNS();

            String jobId = initiateJobRequest();
            System.out.println("Jobid = " + jobId);

            Boolean success = waitForJobToComplete(jobId, sqsQueueURL);
            if (!success) { throw new Exception("Job did not complete
successfully."); }

            downloadJobOutput(jobId);

            cleanUp();

        } catch (Exception e) {
            System.err.println("Inventory retrieval failed.");
            System.err.println(e);
        }
    }

    private static void setupSQS() {
        CreateQueueRequest request = new CreateQueueRequest()
```

```
            .withQueueName(sqsQueueName);
        CreateQueueResult result = sqsClient.createQueue(request);
        sqsQueueURL = result.getQueueUrl();

        GetQueueAttributesRequest qRequest = new GetQueueAttributesRequest()
            .withQueueUrl(sqsQueueURL)
            .withAttributeNames("QueueArn");

        GetQueueAttributesResult qResult = sqsClient.getQueueAttributes(qRequest);
        sqsQueueARN = qResult.getAttributes().get("QueueArn");

        Policy sqsPolicy =
            new Policy().withStatements(
                    new Statement(Effect.Allow)
                    .withPrincipals(Principal.AllUsers)
                    .withActions(SQSActions.SendMessage)
                    .withResources(new Resource(sqsQueueARN)));
        Map<String, String> queueAttributes = new HashMap<String, String>();
        queueAttributes.put("Policy", sqsPolicy.toJson());
        sqsClient.setQueueAttributes(new SetQueueAttributesRequest(sqsQueueURL,
queueAttributes));

    }
    private static void setupSNS() {
        CreateTopicRequest request = new CreateTopicRequest()
            .withName(snsTopicName);
        CreateTopicResult result = snsClient.createTopic(request);
        snsTopicARN = result.getTopicArn();

        SubscribeRequest request2 = new SubscribeRequest()
            .withTopicArn(snsTopicARN)
            .withEndpoint(sqsQueueARN)
            .withProtocol("sqs");
        SubscribeResult result2 = snsClient.subscribe(request2);

        snsSubscriptionARN = result2.getSubscriptionArn();
    }
    private static String initiateJobRequest() {

        JobParameters jobParameters = new JobParameters()
            .withType("inventory-retrieval")
            .withSNSTopic(snsTopicARN);

        InitiateJobRequest request = new InitiateJobRequest()
```

```
            .withVaultName(vaultName)
            .withJobParameters(jobParameters);

        InitiateJobResult response = client.initiateJob(request);

        return response.getJobId();
    }

    private static Boolean waitForJobToComplete(String jobId, String sqsQueueUrl)
throws InterruptedException, JsonParseException, IOException {

        Boolean messageFound = false;
        Boolean jobSuccessful = false;
        ObjectMapper mapper = new ObjectMapper();
        JsonFactory factory = mapper.getFactory();

        while (!messageFound) {
            List<Message> msgs = sqsClient.receiveMessage(
               new
ReceiveMessageRequest(sqsQueueUrl).withMaxNumberOfMessages(10)).getMessages();

            if (msgs.size() > 0) {
                for (Message m : msgs) {
                    JsonParser jpMessage = factory.createJsonParser(m.getBody());
                    JsonNode jobMessageNode = mapper.readTree(jpMessage);
                    String jobMessage = jobMessageNode.get("Message").textValue();

                    JsonParser jpDesc = factory.createJsonParser(jobMessage);
                    JsonNode jobDescNode = mapper.readTree(jpDesc);
                    String retrievedJobId = jobDescNode.get("JobId").textValue();
                    String statusCode = jobDescNode.get("StatusCode").textValue();
                    if (retrievedJobId.equals(jobId)) {
                        messageFound = true;
                        if (statusCode.equals("Succeeded")) {
                            jobSuccessful = true;
                        }
                    }
                }

            } else {
              Thread.sleep(sleepTime * 1000);
            }
          }
        return (messageFound && jobSuccessful);
```

```
    }

    private static void downloadJobOutput(String jobId) throws IOException {

        GetJobOutputRequest getJobOutputRequest = new GetJobOutputRequest()
            .withVaultName(vaultName)
            .withJobId(jobId);
        GetJobOutputResult getJobOutputResult =
 client.getJobOutput(getJobOutputRequest);

        FileWriter fstream = new FileWriter(fileName);
        BufferedWriter out = new BufferedWriter(fstream);
        BufferedReader in = new BufferedReader(new
 InputStreamReader(getJobOutputResult.getBody()));
        String inputLine;
        try {
            while ((inputLine = in.readLine()) != null) {
                out.write(inputLine);
            }
        }catch(IOException e) {
            throw new AmazonClientException("Unable to save archive", e);
        }finally{
            try {in.close();}  catch (Exception e) {}
            try {out.close();}  catch (Exception e) {}
        }
        System.out.println("Retrieved inventory to " + fileName);
    }

    private static void cleanUp() {
        snsClient.unsubscribe(new UnsubscribeRequest(snsSubscriptionARN));
        snsClient.deleteTopic(new DeleteTopicRequest(snsTopicARN));
        sqsClient.deleteQueue(new DeleteQueueRequest(sqsQueueURL));
    }
}
```

Mengunduh Inventaris Vault di Amazon Glacier Menggunakan AWS SDK for .NET

Berikut adalah langkah-langkah untuk mengambil inventaris vault menggunakan API tingkat rendah dari AWS SDK for .NET. API tingkat tinggi tidak mendukung pengambilan inventaris vault.

1. Buat instans dari kelas `AmazonGlacierClient` (klien).

Anda perlu menentukan AWS Wilayah tempat brankas berada. Semua operasi yang Anda lakukan menggunakan klien ini berlaku untuk AWS Wilayah tersebut.

2. Mulai tugas pengambilan inventaris dengan menjalankan metode `InitiateJob`.

   Anda memberikan informasi tugas di objek `InitiateJobRequest`. Amazon Glacier (Amazon Glacier) mengembalikan ID pekerjaan sebagai tanggapan. Respons tersedia dalam instans dari kelas `InitiateJobResponse`.

```
AmazonGlacierClient client;
client = new AmazonGlacierClient(Amazon.RegionEndpoint.USWest2);

InitiateJobRequest initJobRequest = new InitiateJobRequest()
{
  VaultName = vaultName,
  JobParameters = new JobParameters()
  {
    Type = "inventory-retrieval",
    SNSTopic = "*** Provide Amazon SNS topic arn ***",
  }
};
InitiateJobResponse initJobResponse = client.InitiateJob(initJobRequest);
string jobId = initJobResponse.JobId;
```

3. Tunggu hingga tugas selesai.

   Anda harus menunggu hingga output tugas siap diunduh. Jika Anda telah menyetel konfigurasi notifikasi di vault yang mengidentifikasi topik Amazon Simple Notification Service (Amazon SNS), atau menetapkan topik Amazon SNS saat memulai pekerjaan, Amazon Glacier akan mengirimkan pesan ke topik tersebut setelah menyelesaikan pekerjaan. Contoh kode yang diberikan di bagian berikut menggunakan Amazon SNS untuk Amazon Glacier untuk mempublikasikan pesan.

   Anda juga dapat melakukan polling Amazon Glacier dengan memanggil metode untuk menentukan `DescribeJob` status penyelesaian pekerjaan. Meskipun, menggunakan topik Amazon SNS untuk notifikasi adalah pendekatan yang disarankan.

4. Unduh output tugas (data inventaris vault) dengan menjalankan metode `GetJobOutput`.

   Anda memberikan ID akun, nama vault, dan informasi ID tugas dengan membuat instans dari kelas `GetJobOutputRequest`. Jika Anda tidak memberikan ID akun, ID akun yang terkait

dengan kredensial yang Anda berikan untuk menandatangani permintaan diambil. Untuk informasi selengkapnya, lihat Menggunakan AWS SDK for .NET dengan Amazon Glacier.

Output yang dikembalikan Amazon Glacier tersedia di objek. `GetJobOutputResponse`

```
GetJobOutputRequest getJobOutputRequest = new GetJobOutputRequest()
{
  JobId = jobId,
  VaultName = vaultName
};

GetJobOutputResponse getJobOutputResponse =
 client.GetJobOutput(getJobOutputRequest);
using (Stream webStream = getJobOutputResponse.Body)
{
   using (Stream fileToSave = File.OpenWrite(fileName))
   {
     CopyStream(webStream, fileToSave);
   }
}
```

> **Note**
>
> Untuk informasi tentang tugas yang terkait REST API yang mendasarinya, lihat Operasi Tugas.

Contoh: Mengambil Inventaris Vault Menggunakan API Tingkat Rendah AWS SDK for .NET

Contoh kode C# berikut mengambil inventaris vault untuk vault yang ditentukan.

Contoh tersebut melakukan tugas-tugas berikut:

- Menyiapkan topik Amazon SNS.

  Amazon Glacier mengirimkan pemberitahuan ke topik ini setelah menyelesaikan pekerjaan.
- Siapkan antrean Amazon SQS.

Contoh melampirkan kebijakan ke antrean untuk mengizinkan topik Amazon SNS mengirim pesan.

- Mulai tugas untuk mengunduh arsip yang ditentukan.

  Dalam permintaan pekerjaan, contoh menentukan topik Amazon SNS sehingga Amazon Glacier dapat mengirim pesan setelah menyelesaikan pekerjaan.

- Memeriksa antrean Amazon SQS secara berkala untuk pesan.

  Jika ada pesan, urai JSON dan periksa apakah tugas berhasil diselesaikan. Jika ya, unduh arsipnya. Contoh kode menggunakan pustaka JSON.NET (lihat JSON.NET) untuk mengurai JSON.

- Bersihkan dengan menghapus topik Amazon SNS dan antrean Amazon SQS yang dibuat.

Example

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Threading;
using Amazon.Glacier;
using Amazon.Glacier.Model;
using Amazon.Glacier.Transfer;
using Amazon.Runtime;
using Amazon.SimpleNotificationService;
using Amazon.SimpleNotificationService.Model;
using Amazon.SQS;
using Amazon.SQS.Model;
using Newtonsoft.Json;

namespace glacier.amazon.com.docsamples
{
  class VaultInventoryJobLowLevelUsingSNSSQS
  {
    static string topicArn;
    static string queueUrl;
    static string queueArn;
    static string vaultName = "*** Provide vault name ***";
    static string fileName  = "*** Provide file name and path where to store inventory
 ***";
    static AmazonSimpleNotificationServiceClient snsClient;
    static AmazonSQSClient sqsClient;
    const string SQS_POLICY =
```

```
        "{" +
        "    \"Version\" : \"2012-10-17\",&TCX5-2025-waiver;" +
        "    \"Statement\" : [" +
        "        {" +
        "            \"Sid\" : \"sns-rule\"," +
        "            \"Effect\" : \"Allow\"," +
        "            \"Principal\" : {\"AWS\" : \"arn:aws:iam::123456789012:root\" }," +
        "            \"Action\"    : \"sqs:SendMessage\"," +
        "            \"Resource\"  : \"{QuernArn}\"," +
        "            \"Condition\" : {" +
        "                \"ArnLike\" : {" +
        "                    \"aws:SourceArn\" : \"{TopicArn}\"" +
        "                }" +
        "            }" +
        "        }" +
        "    ]" +
        "}";

    public static void Main(string[] args)
    {
      AmazonGlacierClient client;
      try
      {
        using (client = new AmazonGlacierClient(Amazon.RegionEndpoint.USWest2))
        {
            Console.WriteLine("Setup SNS topic and SQS queue.");
            SetupTopicAndQueue();
            Console.WriteLine("To continue, press Enter"); Console.ReadKey();

            Console.WriteLine("Retrieve Inventory List");
            GetVaultInventory(client);
        }
        Console.WriteLine("Operations successful.");
        Console.WriteLine("To continue, press Enter"); Console.ReadKey();
      }
      catch (AmazonGlacierException e) { Console.WriteLine(e.Message); }
      catch (AmazonServiceException e) { Console.WriteLine(e.Message); }
      catch (Exception e) { Console.WriteLine(e.Message); }
      finally
      {
       // Delete SNS topic and SQS queue.
       snsClient.DeleteTopic(new DeleteTopicRequest() { TopicArn = topicArn });
       sqsClient.DeleteQueue(new DeleteQueueRequest() { QueueUrl = queueUrl });
```

```
      }
    }

    static void SetupTopicAndQueue()
    {
      long ticks = DateTime.Now.Ticks;

      // Setup SNS topic.
      snsClient = new
AmazonSimpleNotificationServiceClient(Amazon.RegionEndpoint.USWest2);
      sqsClient = new AmazonSQSClient(Amazon.RegionEndpoint.USWest2);

      topicArn = snsClient.CreateTopic(new CreateTopicRequest { Name =
"GlacierDownload-" + ticks }).TopicArn;
      Console.Write("topicArn: "); Console.WriteLine(topicArn);

      CreateQueueRequest createQueueRequest =  new CreateQueueRequest();
      createQueueRequest.QueueName = "GlacierDownload-" + ticks;
      CreateQueueResponse createQueueResponse =
sqsClient.CreateQueue(createQueueRequest);
      queueUrl = createQueueResponse.QueueUrl;
      Console.Write("QueueURL: "); Console.WriteLine(queueUrl);

      GetQueueAttributesRequest getQueueAttributesRequest = new
GetQueueAttributesRequest();
      getQueueAttributesRequest.AttributeNames = new List<string> { "QueueArn" };
      getQueueAttributesRequest.QueueUrl = queueUrl;
      GetQueueAttributesResponse response =
sqsClient.GetQueueAttributes(getQueueAttributesRequest);
      queueArn = response.QueueARN;
      Console.Write("QueueArn: ");Console.WriteLine(queueArn);

      // Setup the Amazon SNS topic to publish to the SQS queue.
      snsClient.Subscribe(new SubscribeRequest()
      {
        Protocol = "sqs",
        Endpoint = queueArn,
        TopicArn = topicArn
      });

      // Add the policy to the queue so SNS can send messages to the queue.
      var policy = SQS_POLICY.Replace("{TopicArn}", topicArn).Replace("{QuernArn}",
queueArn);
```

```
      sqsClient.SetQueueAttributes(new SetQueueAttributesRequest()
      {
          QueueUrl = queueUrl,
          Attributes = new Dictionary<string, string>
          {
              { QueueAttributeName.Policy, policy }
          }
      });

   }

   static void GetVaultInventory(AmazonGlacierClient client)
   {
     // Initiate job.
     InitiateJobRequest initJobRequest = new InitiateJobRequest()
     {
       VaultName = vaultName,
       JobParameters = new JobParameters()
       {
         Type = "inventory-retrieval",
         Description = "This job is to download a vault inventory.",
         SNSTopic = topicArn,
       }
     };

     InitiateJobResponse initJobResponse = client.InitiateJob(initJobRequest);
     string jobId = initJobResponse.JobId;

     // Check queue for a message and if job completed successfully, download
inventory.
     ProcessQueue(jobId, client);
   }

   private static void ProcessQueue(string jobId, AmazonGlacierClient client)
   {
     ReceiveMessageRequest receiveMessageRequest = new ReceiveMessageRequest()
{ QueueUrl = queueUrl, MaxNumberOfMessages = 1 };
     bool jobDone = false;
     while (!jobDone)
     {
       Console.WriteLine("Poll SQS queue");
       ReceiveMessageResponse receiveMessageResponse =
sqsClient.ReceiveMessage(receiveMessageRequest);
       if (receiveMessageResponse.Messages.Count == 0)
```

```
        {
          Thread.Sleep(10000 * 60);
          continue;
        }
        Console.WriteLine("Got message");
        Message message = receiveMessageResponse.Messages[0];
        Dictionary<string, string> outerLayer =
JsonConvert.DeserializeObject<Dictionary<string, string>>(message.Body);
        Dictionary<string, object> fields =
JsonConvert.DeserializeObject<Dictionary<string, object>>(outerLayer["Message"]);
        string statusCode = fields["StatusCode"] as string;

        if (string.Equals(statusCode, GlacierUtils.JOB_STATUS_SUCCEEDED,
StringComparison.InvariantCultureIgnoreCase))
        {
          Console.WriteLine("Downloading job output");
          DownloadOutput(jobId, client); // Save job output to the specified file
location.
        }
        else if (string.Equals(statusCode, GlacierUtils.JOB_STATUS_FAILED,
StringComparison.InvariantCultureIgnoreCase))
          Console.WriteLine("Job failed... cannot download the inventory.");

        jobDone = true;
        sqsClient.DeleteMessage(new DeleteMessageRequest() { QueueUrl = queueUrl,
ReceiptHandle = message.ReceiptHandle });
      }
    }

    private static void DownloadOutput(string jobId, AmazonGlacierClient client)
    {
      GetJobOutputRequest getJobOutputRequest = new GetJobOutputRequest()
      {
        JobId = jobId,
        VaultName = vaultName
      };

      GetJobOutputResponse getJobOutputResponse =
client.GetJobOutput(getJobOutputRequest);
      using (Stream webStream = getJobOutputResponse.Body)
      {
        using (Stream fileToSave = File.OpenWrite(fileName))
        {
          CopyStream(webStream, fileToSave);
```

```
        }
      }
    }

    public static void CopyStream(Stream input, Stream output)
    {
      byte[] buffer = new byte[65536];
      int length;
      while ((length = input.Read(buffer, 0, buffer.Length)) > 0)
      {
        output.Write(buffer, 0, length);
      }
    }
  }
}
```

Mengunduh Inventaris Vault Menggunakan REST API

Untuk mengunduh inventaris vault menggunakan REST API

Mengunduh inventaris vault adalah proses dua langkah.

1. Mulai tugas dari tipe `inventory-retrieval`. Untuk informasi selengkapnya, lihat Memulai Tugas (POST jobs).
2. Setelah tugas selesai, unduh data inventaris. Lihat informasi yang lebih lengkap di Mendapatkan Output Tugas (GET output).

Mengunduh Inventaris Vault di Amazon Glacier Menggunakan AWS Command Line Interface

Ikuti langkah-langkah ini untuk mengunduh inventaris vault di Amazon Glacier (Amazon Glacier) menggunakan (). AWS Command Line Interface AWS CLI

Topik

- (Prasyarat) Menyiapkan AWS CLI
- Contoh: Mengunduh Inventaris Vault Menggunakan AWS CLI

(Prasyarat) Menyiapkan AWS CLI

1. Unduh dan konfigurasikan AWS CLI. Untuk melakukannya, lihat topik berikut di Panduan Pengguna AWS Command Line Interface :

   Instalasi AWS Command Line Interface

   Mengkonfigurasi AWS Command Line Interface
2. Verifikasi AWS CLI pengaturan Anda dengan memasukkan perintah berikut pada prompt perintah. Perintah ini tidak memberikan kredensial secara eksplisit, sehingga kredensial profil default digunakan.

   - Coba gunakan perintah bantuan.

     ```
     aws help
     ```

   - Untuk mendapatkan daftar brankas Amazon Glacier pada akun yang dikonfigurasi, gunakan perintah. `list-vaults` Ganti *123456789012* dengan Akun AWS ID Anda.

     ```
     aws glacier list-vaults --account-id 123456789012
     ```

   - Untuk melihat data konfigurasi saat ini untuk AWS CLI, gunakan `aws configure list` perintah.

     ```
     aws configure list
     ```

Contoh: Mengunduh Inventaris Vault Menggunakan AWS CLI

1. Gunakan perintah `initiate-job` untuk memulai tugas pengambilan inventaris.

   ```
   aws glacier initiate-job --vault-name awsexamplevault --account-id 111122223333 --
   job-parameters='{"Type": "inventory-retrieval"}'
   ```

   Output yang diharapkan:

   ```
   {
       "location": "/111122223333/vaults/awsexamplevault/jobs/*** jobid ***",
       "jobId": "*** jobid ***"
   ```

```
}
```

2. Gunakan perintah `describe-job` untuk memeriksa status tugas pengambilan sebelumnya.

```
aws glacier describe-job --vault-name awsexamplevault --account-id 111122223333 --
job-id *** jobid ***
```

Output yang diharapkan:

```
{
    "InventoryRetrievalParameters": {
        "Format": "JSON"
    },
    "VaultARN": "*** vault arn ***",
    "Completed": false,
    "JobId": "*** jobid ***",
    "Action": "InventoryRetrieval",
    "CreationDate": "*** job creation date ***",
    "StatusCode": "InProgress"
}
```

3. Tunggu hingga tugas selesai.

Anda harus menunggu hingga output tugas siap diunduh. ID pekerjaan tidak kedaluwarsa setidaknya 24 jam setelah Amazon Glacier menyelesaikan pekerjaan. Jika Anda telah menyetel konfigurasi notifikasi di vault, atau menetapkan topik Amazon Simple Notification Service (Amazon SNS) saat memulai pekerjaan, Amazon Glacier akan mengirimkan pesan ke topik tersebut setelah menyelesaikan pekerjaan.

Anda dapat menetapkan konfigurasi notifikasi untuk peristiwa tertentu di vault. Untuk informasi selengkapnya, lihat Mengkonfigurasi Pemberitahuan Vault di Amazon Glacier. Amazon Glacier mengirim pesan ke topik SNS yang ditentukan kapan saja peristiwa tertentu terjadi.

4. Setelah selesai, gunakan perintah `get-job-output` untuk mengunduh tugas pengambilan ke file `output.json`.

```
aws glacier get-job-output --vault-name awsexamplevault --account-id 111122223333
 --job-id *** jobid *** output.json
```

Perintah ini menghasilkan file dengan bidang berikut.

```
{
"VaultARN":"arn:aws:glacier:region:111122223333:vaults/awsexamplevault",
"InventoryDate":"*** job completion date ***",
"ArchiveList":[
{"ArchiveId":"*** archiveid ***",
"ArchiveDescription":"*** archive description (if set) ***",
"CreationDate":"*** archive creation date ***",
"Size":"*** archive size (in bytes) ***",
"SHA256TreeHash":"*** archive hash ***"
}
{"ArchiveId":
...
]}
```

Mengkonfigurasi Pemberitahuan Vault di Amazon Glacier

Mengambil apa pun dari Amazon Glacier, seperti arsip dari lemari besi atau inventaris lemari besi, adalah proses dua langkah.

1. Mulai tugas pengambilan.
2. Setelah pekerjaan selesai, unduh output pekerjaan.

Anda dapat mengatur konfigurasi notifikasi pada vault sehingga ketika pekerjaan selesai, pesan dikirim ke topik Simple Notification Service Amazon (Amazon SNS).

Topik

- Mengkonfigurasi Pemberitahuan Vault di Amazon Glacier: Konsep Umum
- Mengonfigurasi Pemberitahuan Vault di Amazon Glacier Menggunakan AWS SDK for Java
- Mengonfigurasi Pemberitahuan Vault di Amazon Glacier Menggunakan AWS SDK for .NET
- Mengonfigurasi Pemberitahuan Vault di Amazon Glacier Menggunakan REST API
- Mengonfigurasi Pemberitahuan Vault dengan Menggunakan Konsol Amazon Glacier
- Mengkonfigurasi Pemberitahuan Vault Menggunakan AWS Command Line Interface

Mengkonfigurasi Pemberitahuan Vault di Amazon Glacier: Konsep Umum

Permintaan pekerjaan pengambilan Amazon Glacier dijalankan secara asinkron. Anda harus menunggu hingga Amazon Glacier menyelesaikan pekerjaan sebelum Anda bisa mendapatkan outputnya. Anda dapat secara berkala polling Amazon Glacier untuk menentukan status pekerjaan, tetapi itu bukan pendekatan yang optimal. Amazon Glacier juga mendukung notifikasi. Ketika pekerjaan selesai, pekerjaan dapat memposting pesan ke topik Amazon Simple Notification Service (Amazon SNS). Menggunakan fitur ini mengharuskan Anda untuk mengatur konfigurasi notifikasi pada vault. Dalam konfigurasi, Anda mengidentifikasi satu atau beberapa peristiwa dan topik Amazon SNS yang Anda inginkan Amazon Glacier untuk mengirim pesan ketika peristiwa terjadi.

Amazon Glacier mendefinisikan peristiwa yang secara khusus terkait dengan penyelesaian pekerjaan `ArchiveRetrievalCompleted` (`InventoryRetrievalCompleted`,) yang dapat Anda tambahkan ke konfigurasi notifikasi vault. Saat pekerjaan tertentu selesai, Amazon Glacier menerbitkan pesan notifikasi ke topik SNS.

Konfigurasi notifikasi adalah dokumen JSON seperti yang ditunjukkan dalam contoh berikut.

```
{
   "SNSTopic": "arn:aws:sns:us-west-2:012345678901:mytopic",
   "Events": ["ArchiveRetrievalCompleted", "InventoryRetrievalCompleted"]
}
```

Anda hanya dapat mengonfigurasi satu topik Amazon SNS untuk vault.

> **Note**
>
> Menambahkan konfigurasi notifikasi ke vault menyebabkan Amazon Glacier mengirim notifikasi setiap kali peristiwa yang ditentukan dalam konfigurasi notifikasi terjadi. Anda juga dapat menentukan topik Amazon SNS di setiap permintaan inisiasi tugas secara opsional. Jika Anda menambahkan konfigurasi notifikasi pada vault dan juga menentukan topik Amazon SNS dalam permintaan pekerjaan inisiasi, Amazon Glacier akan mengirimkan kedua notifikasi tersebut.

Pesan penyelesaian pekerjaan yang dikirimkan Amazon Glacier mencakup informasi seperti jenis pekerjaan `InventoryRetrieval` (`ArchiveRetrieval`,), status penyelesaian pekerjaan, nama

topik SNS, kode status pekerjaan, dan ARN vault. Berikut ini adalah contoh pemberitahuan Amazon Glacier dikirim ke topik SNS setelah `InventoryRetrieval` pekerjaan selesai.

```
{
 "Action": "InventoryRetrieval",
 "ArchiveId": null,
 "ArchiveSizeInBytes": null,
 "Completed": true,
 "CompletionDate": "2012-06-12T22:20:40.790Z",
 "CreationDate": "2012-06-12T22:20:36.814Z",
 "InventorySizeInBytes":11693,
 "JobDescription": "my retrieval job",
 "JobId":"HkF9p6o7yjhFx-
K3CGl6fuSm6VzW9T7esGQfco8nUXVYwS0jlb5gq1JZ55yHgt5vP54ZShjoQzQVVh7vEXAMPLEjobID",
 "SHA256TreeHash":null,
 "SNSTopic": "arn:aws:sns:us-west-2:012345678901:mytopic",
 "StatusCode":"Succeeded",
 "StatusMessage": "Succeeded",
 "VaultARN": "arn:aws:glacier:us-west-2:012345678901:vaults/examplevault"
}
```

Jika bidang `Completed` benar, Anda juga harus memeriksa `StatusCode` untuk memeriksa apakah tugas berhasil diselesaikan atau gagal.

> **Note**
>
> Topik Amazon SNS harus mengizinkan brankas untuk mempublikasikan pemberitahuan. Secara default, hanya pemilik topik Amazon SNS yang dapat mempublikasikan pesan ke topik tersebut. Namun, jika topik Amazon SNS dan brankas dimiliki oleh yang berbeda Akun AWS, maka Anda harus mengonfigurasi topik Amazon SNS untuk menerima publikasi dari brankas. Anda dapat mengonfigurasi kebijakan topik Amazon SNS di konsol Amazon SNS.

Untuk informasi selengkapnya tentang Amazon SNS, lihat Mulai Menggunakan Amazon SNS.

Mengonfigurasi Pemberitahuan Vault di Amazon Glacier Menggunakan AWS SDK for Java

Berikut adalah langkah-langkah untuk mengonfigurasi notifikasi di vault menggunakan API tingkat rendah dari AWS SDK for Java.

1. Buat instans dari kelas `AmazonGlacierClient` (klien).

   Anda perlu menentukan AWS Wilayah tempat brankas berada. Semua operasi yang Anda lakukan menggunakan klien ini berlaku untuk AWS Wilayah tersebut.

2. Berikan informasi konfigurasi notifikasi dengan membuat instans dari kelas `SetVaultNotificationsRequest`.

   Anda harus memberikan nama vault, informasi konfigurasi notifikasi, dan ID akun. Dalam menentukan konfigurasi notifikasi, Anda memberikan Amazon Resource Name (ARN) dari topik Amazon SNS yang ada dan satu atau beberapa peristiwa yang ingin Anda beri tahu. Untuk daftar peristiwa yang didukung, lihat Menetapkan Konfigurasi Notifikasi Vault (PUT notification-configuration)).

3. Jalankan metode `setVaultNotifications` dengan menyediakan objek permintaan sebagai parameter.

Potongan kode Java berikut menggambarkan langkah-langkah sebelumnya. Potongan menetapkan konfigurasi notifikasi di vault. Konfigurasi meminta Amazon Glacier (Amazon Glacier) untuk mengirim pemberitahuan ke topik Amazon SNS yang ditentukan saat peristiwa atau peristiwa terjadi. `ArchiveRetrievalCompleted` `InventoryRetrievalCompleted`

```
SetVaultNotificationsRequest request = new SetVaultNotificationsRequest()
        .withAccountId("-")
        .withVaultName("*** provide vault name ***")
        .withVaultNotificationConfig(
                new VaultNotificationConfig()
                .withSNSTopic("*** provide SNS topic ARN ***")
                .withEvents("ArchiveRetrievalCompleted", "InventoryRetrievalCompleted")
         );
client.setVaultNotifications(request);
```

> **Note**
>
> Untuk informasi tentang REST API yang mendasarinya, lihat Operasi Vault.

Contoh: Menyetel Konfigurasi Pemberitahuan di Vault Menggunakan AWS SDK for Java

Contoh kode Java berikut menetapkan konfigurasi notifikasi vault, menghapus konfigurasi, lalu mengembalikan konfigurasi. Untuk step-by-step petunjuk tentang cara menjalankan contoh berikut, lihatMenggunakan AWS SDK for Java dengan Amazon Glacier.

Example

```
import java.io.IOException;

import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.glacier.AmazonGlacierClient;
import com.amazonaws.services.glacier.model.DeleteVaultNotificationsRequest;
import com.amazonaws.services.glacier.model.GetVaultNotificationsRequest;
import com.amazonaws.services.glacier.model.GetVaultNotificationsResult;
import com.amazonaws.services.glacier.model.SetVaultNotificationsRequest;
import com.amazonaws.services.glacier.model.VaultNotificationConfig;


public class AmazonGlacierVaultNotifications {

    public static AmazonGlacierClient client;
    public static String vaultName = "*** provide vault name ****";
    public static String snsTopicARN = "*** provide sns topic ARN ***";

    public static void main(String[] args) throws IOException {

     ProfileCredentialsProvider credentials = new ProfileCredentialsProvider();

        client = new AmazonGlacierClient(credentials);
        client.setEndpoint("https://glacier.us-east-1.amazonaws.com/");

        try {

            System.out.println("Adding notification configuration to the vault.");
            setVaultNotifications();
            getVaultNotifications();
            deleteVaultNotifications();

        } catch (Exception e) {
            System.err.println("Vault operations failed." + e.getMessage());
        }
```

```
    }

    private static void setVaultNotifications() {
        VaultNotificationConfig config = new VaultNotificationConfig()
            .withSNSTopic(snsTopicARN)
            .withEvents("ArchiveRetrievalCompleted", "InventoryRetrievalCompleted");

        SetVaultNotificationsRequest request = new SetVaultNotificationsRequest()
                .withVaultName(vaultName)
                .withVaultNotificationConfig(config);

        client.setVaultNotifications(request);
        System.out.println("Notification configured for vault: " + vaultName);
    }

    private static void getVaultNotifications() {
        VaultNotificationConfig notificationConfig = null;
        GetVaultNotificationsRequest request = new GetVaultNotificationsRequest()
                .withVaultName(vaultName);
        GetVaultNotificationsResult result = client.getVaultNotifications(request);
        notificationConfig = result.getVaultNotificationConfig();

        System.out.println("Notifications configuration for vault: "
                + vaultName);
        System.out.println("Topic: " + notificationConfig.getSNSTopic());
        System.out.println("Events: " + notificationConfig.getEvents());
    }

    private static void deleteVaultNotifications() {
            DeleteVaultNotificationsRequest request = new
 DeleteVaultNotificationsRequest()
                .withVaultName(vaultName);
            client.deleteVaultNotifications(request);
            System.out.println("Notifications configuration deleted for vault: " +
 vaultName);
    }
}
```

Mengonfigurasi Pemberitahuan Vault di Amazon Glacier Menggunakan AWS SDK for .NET

Berikut adalah langkah-langkah untuk mengonfigurasi notifikasi di vault menggunakan API tingkat rendah dari AWS SDK for .NET.

1. Buat instans dari kelas `AmazonGlacierClient` (klien).

   Anda perlu menentukan AWS Wilayah tempat brankas berada. Semua operasi yang Anda lakukan menggunakan klien ini berlaku untuk AWS Wilayah tersebut.
2. Berikan informasi konfigurasi notifikasi dengan membuat instans dari kelas `SetVaultNotificationsRequest`.

   Anda harus memberikan nama vault, informasi konfigurasi notifikasi, dan ID akun. Jika Anda tidak memberikan ID akun, ID akun yang terkait dengan kredensial yang Anda berikan untuk menandatangani permintaan diambil. Untuk informasi selengkapnya, lihat Menggunakan AWS SDK for .NET dengan Amazon Glacier.

   Dalam menentukan konfigurasi notifikasi, Anda memberikan Amazon Resource Name (ARN) dari topik Amazon SNS yang ada dan satu atau beberapa peristiwa yang ingin Anda beri tahu. Untuk daftar peristiwa yang didukung, lihat Menetapkan Konfigurasi Notifikasi Vault (PUT notification-configuration)).
3. Jalankan metode `SetVaultNotifications` dengan menyediakan objek permintaan sebagai parameter.
4. Setelah mengatur konfigurasi notifikasi di vault, Anda dapat mengambil informasi konfigurasi dengan memanggil metode `GetVaultNotifications`, dan menghapusnya dengan memanggil metode `DeleteVaultNotifications` yang disediakan oleh klien.

Contoh: Menyetel Konfigurasi Pemberitahuan di Vault Menggunakan AWS SDK for .NET

Contoh kode #C berikut menjelaskan langkah sebelumnya. Contoh mengatur konfigurasi notifikasi di vault ("`examplevault`") di Wilayah US West (Oregon), mengambil konfigurasi, lalu menghapusnya. Konfigurasi meminta Amazon Glacier (Amazon Glacier) untuk mengirim pemberitahuan ke topik Amazon SNS yang ditentukan saat peristiwa atau peristiwa terjadi. `ArchiveRetrievalCompleted` `InventoryRetrievalCompleted`

> **Note**
>
> Untuk informasi tentang REST API yang mendasarinya, lihat Operasi Vault.

Untuk step-by-step instruksi untuk menjalankan contoh berikut, lihatMenjalankan Contoh Kode. Anda perlu memperbarui kode seperti yang ditunjukkan serta memberikan nama vault yang ada dan topik Amazon SNS.

Example

```
using System;
using System.Collections.Generic;
using Amazon.Glacier;
using Amazon.Glacier.Model;
using Amazon.Runtime;

namespace glacier.amazon.com.docsamples
{
  class VaultNotificationSetGetDelete
  {
    static string vaultName   = "examplevault";
    static string snsTopicARN = "*** Provide Amazon SNS topic ARN ***";

    static IAmazonGlacier client;

    public static void Main(string[] args)
    {
      try
      {
        using (client = new AmazonGlacierClient(Amazon.RegionEndpoint.USWest2))
        {
          Console.WriteLine("Adding notification configuration to the vault.");
          SetVaultNotificationConfig();
          GetVaultNotificationConfig();
          Console.WriteLine("To delete vault notification configuration, press Enter");
          Console.ReadKey();
          DeleteVaultNotificationConfig();
        }
      }
      catch (AmazonGlacierException e) { Console.WriteLine(e.Message); }
      catch (AmazonServiceException e) { Console.WriteLine(e.Message); }
```

```
      catch (Exception e) { Console.WriteLine(e.Message); }
      Console.WriteLine("To continue, press Enter");
      Console.ReadKey();
    }

    static void SetVaultNotificationConfig()
    {

      SetVaultNotificationsRequest request = new SetVaultNotificationsRequest()
      {
        VaultName = vaultName,
        VaultNotificationConfig = new VaultNotificationConfig()
        {
          Events   = new List<string>() { "ArchiveRetrievalCompleted",
"InventoryRetrievalCompleted" },
          SNSTopic = snsTopicARN
        }
      };
      SetVaultNotificationsResponse response = client.SetVaultNotifications(request);
    }

    static void GetVaultNotificationConfig()
    {
      GetVaultNotificationsRequest request = new GetVaultNotificationsRequest()
      {
        VaultName = vaultName,
        AccountId = "-"
      };
      GetVaultNotificationsResponse response = client.GetVaultNotifications(request);
      Console.WriteLine("SNS Topic ARN: {0}",
response.VaultNotificationConfig.SNSTopic);
      foreach (string s in response.VaultNotificationConfig.Events)
        Console.WriteLine("Event : {0}", s);
    }

    static void DeleteVaultNotificationConfig()
    {
      DeleteVaultNotificationsRequest request = new DeleteVaultNotificationsRequest()
      {
        VaultName = vaultName
      };
      DeleteVaultNotificationsResponse response =
client.DeleteVaultNotifications(request);
    }
```

```
  }
}
```

Mengonfigurasi Pemberitahuan Vault di Amazon Glacier Menggunakan REST API

Mengonfigurasi notifikasi vault menggunakan REST API, lihat Menetapkan Konfigurasi Notifikasi Vault (PUT notification-configuration). Selain itu, Anda juga bisa mendapatkan notifikasi vault (Mendapatkan Notifikasi Vault (GET notification-configuration)) dan menghapus notifikasi vault (Menghapus Notifikasi Vault (DELETE notification-configuration)).

Mengonfigurasi Pemberitahuan Vault dengan Menggunakan Konsol Amazon Glacier

Bagian ini menjelaskan cara mengonfigurasi pemberitahuan vault dengan menggunakan konsol Amazon Glacier. Saat mengonfigurasi notifikasi, Anda menentukan peristiwa penyelesaian pekerjaan yang mengirim pemberitahuan ke topik Simple Notification Service Amazon (Amazon SNS). Selain mengonfigurasi notifikasi terkait vault, Anda juga dapat menentukan topik untuk memublikasikan notifikasi saat Anda memulai tugas. Jika vault Anda dikonfigurasi untuk mengirim pemberitahuan untuk acara tertentu dan Anda juga mengonfigurasi notifikasi dalam permintaan inisiasi pekerjaan, maka dua notifikasi akan dikirim.

Untuk mengonfigurasi notifikasi vault

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Glacier di rumah. https://console.aws.amazon.com/glacier/
2. Di panel navigasi kiri, pilih Vaults.
3. Dalam daftar Vaults, pilih brankas.
4. Di bagian Notifikasi, pilih Edit.
5. Pada halaman Pemberitahuan acara, pilih Aktifkan notifikasi.
6. Di bagian Notifikasi, pilih salah satu opsi Amazon Simple Notification Service (Amazon SNS) berikut, lalu ikuti langkah-langkah yang sesuai:

Opsi Amazon SNS	Tindakan
Buat topik SNS baru	1.

Opsi Amazon SNS	Tindakan
	Pilih Buat topik SNS baru. 2. Untuk nama Topik, masukkan nama topik baru. Nama topik dapat mencapai 256 karakter. Karakter alfanumerik, tanda hubung (-), dan garis bawah (_) diperbolehkan. Nama topik harus unik di dalam akun dan AWS Region. 3. (Opsional) Jika Anda ingin berlangganan topik dengan menggunakan pesan SMS, masukkan nama untuk Nama tampilan. Nama tampilan dapat memiliki hingga 100 karakter.

Opsi Amazon SNS	Tindakan
Pilih topik SNS yang ada	1. Pilih Pilih topik SNS yang ada. 2. Di bawah Tentukan topik SNS, pilih salah satu opsi berikut: • Pilih dari topik SNS Anda Daftar dropdown topik SNS muncul. Pilih topik yang ada dari daftar dropdown. • Masukkan SNS topik ARN Kotak teks ARN topik Amazon SNS muncul. Masukkan Nama Sumber Daya Amazon (ARN) untuk topik SNS Anda. Topik SNS ARN memiliki format berikut: `arn:aws:sns: region:account-id :topic-name` Anda dapat menemukan topik SNS ARN di konsol Amazon SNS.

7. Di bawah Acara, pilih salah satu atau kedua acara yang ingin Anda kirimi notifikasi:

   - Untuk mengirim pemberitahuan hanya ketika pekerjaan pengambilan arsip selesai, pilih Arsip Pekerjaan Pengambilan Lengkap.
   - Untuk mengirim pemberitahuan hanya ketika pekerjaan inventaris vault selesai, pilih Vault Inventory Retrieval Job Complete.

Mengkonfigurasi Pemberitahuan Vault Menggunakan AWS Command Line Interface

Bagian ini menjelaskan cara mengonfigurasi pemberitahuan vault menggunakan. AWS Command Line Interface Ketika Anda mengonfigurasi notifikasi, Anda menentukan peristiwa penyelesaian tugas yang memicu notifikasi untuk topik Amazon Simple Notification Service (Amazon SNS). Selain mengonfigurasi notifikasi untuk vault, Anda juga dapat menentukan topik untuk menerbitkan notifikasi ketika Anda memulai tugas. Jika vault Anda dikonfigurasi untuk memberi tahu peristiwa tertentu dan Anda menentukan notifikasi dalam permintaan inisiasi tugas, dua notifikasi akan dikirim.

Ikuti langkah-langkah ini untuk mengonfigurasi pemberitahuan vault menggunakan. AWS CLI

Topik

- (Prasyarat) Menyiapkan AWS CLI
- Contoh: Konfigurasikan Pemberitahuan Vault Menggunakan AWS CLI

(Prasyarat) Menyiapkan AWS CLI

1. Unduh dan konfigurasikan AWS CLI. Untuk melakukannya, lihat topik berikut di Panduan Pengguna AWS Command Line Interface :

   Memasang AWS Command Line Interface

   Mengkonfigurasi AWS Command Line Interface

2. Verifikasi AWS CLI pengaturan Anda dengan memasukkan perintah berikut pada prompt perintah. Perintah ini tidak memberikan kredensial secara eksplisit, sehingga kredensial profil default digunakan.

   - Coba gunakan perintah bantuan.

     ```
     aws help
     ```

   - Untuk mendapatkan daftar brankas Amazon Glacier pada akun yang dikonfigurasi, gunakan perintah. `list-vaults` Ganti *123456789012* dengan Akun AWS ID Anda.

     ```
     aws glacier list-vaults --account-id 123456789012
     ```

- Untuk melihat data konfigurasi saat ini untuk AWS CLI, gunakan `aws configure list` perintah.

```
aws configure list
```

Contoh: Konfigurasikan Pemberitahuan Vault Menggunakan AWS CLI

1. Gunakan `set-vault-notifications` perintah untuk mengonfigurasi notifikasi yang akan dikirim saat peristiwa tertentu terjadi pada brankas. Secara default, Anda tidak mendapatkan notifikasi apa pun.

```
aws glacier set-vault-notifications --vault-name examplevault --account-
id 111122223333 --vault-notification-config file://notificationconfig.json
```

2. Konfigurasi notifikasi adalah dokumen JSON seperti yang ditunjukkan dalam contoh berikut.

```
{
   "SNSTopic": "arn:aws:sns:us-west-2:012345678901:mytopic",
   "Events": ["ArchiveRetrievalCompleted", "InventoryRetrievalCompleted"]
}
```

   Untuk informasi selengkapnya tentang menggunakan topik Amazon SNS untuk Amazon Glacier, lihat, Mengonfigurasi Pemberitahuan Vault di Amazon Glacier: Konsep Umum

   Untuk informasi selengkapnya tentang Amazon SNS, lihat Mulai Menggunakan Amazon SNS.

Menghapus Vault di Amazon Glacier

Amazon Glacier (Amazon Glacier) menghapus lemari besi hanya jika tidak ada arsip di lemari besi pada inventaris terakhir yang dihitung dan belum ada penulisan ke lemari besi sejak inventaris terakhir. Untuk informasi tentang menghapus arsip, lihat Menghapus Arsip di Amazon Glacier. Untuk informasi tentang mengunduh inventaris vault, Mengunduh Inventaris Vault di Amazon Glacier.

> **Note**
>
> Amazon Glacier menyiapkan inventaris untuk setiap lemari besi secara berkala, setiap 24 jam. Karena inventaris mungkin tidak mencerminkan informasi terbaru, Amazon Glacier

memastikan brankas memang kosong dengan memeriksa apakah ada operasi penulisan sejak inventaris brankas terakhir.

> **Note**
>
> Untuk penghapusan otomatis arsip vault, lihat Penghapusan otomatis arsip vault di Amazon S3 Glacier.

Topik

- Menghapus Vault di Amazon Glacier Menggunakan AWS SDK for Java
- Menghapus Vault di Amazon Glacier Menggunakan AWS SDK for .NET
- Menghapus Vault di Amazon Glacier Menggunakan REST API
- Menghapus Vault Kosong dengan Menggunakan Konsol Amazon Glacier
- Menghapus Vault di Amazon Glacier Menggunakan AWS Command Line Interface

Menghapus Vault di Amazon Glacier Menggunakan AWS SDK for Java

Berikut adalah langkah-langkah untuk menghapus vault menggunakan API tingkat rendah dari AWS SDK for Java.

1. Buat instans dari kelas `AmazonGlacierClient` (klien).

   Anda perlu menentukan AWS Wilayah dari mana Anda ingin menghapus brankas. Semua operasi yang Anda lakukan menggunakan klien ini berlaku untuk AWS Wilayah tersebut.
2. Berikan informasi permintaan dengan membuat instans kelas `DeleteVaultRequest`.

   Anda harus memberi nama vault dan ID akun. Jika Anda tidak memberikan ID akun, ID akun yang terkait dengan kredensial yang Anda berikan untuk menandatangani permintaan akan diambil. Untuk informasi selengkapnya, lihat Menggunakan AWS SDK for Java dengan Amazon Glacier.
3. Jalankan metode `deleteVault` dengan menyediakan objek permintaan sebagai parameter.

   Amazon Glacier (Amazon Glacier) menghapus lemari besi hanya jika kosong. Untuk informasi selengkapnya, lihat Menghapus Vault (DELETE vault).

Potongan kode Java berikut menggambarkan langkah-langkah sebelumnya.

```
try {
    DeleteVaultRequest request = new DeleteVaultRequest()
        .withVaultName("*** provide vault name ***");

    client.deleteVault(request);
    System.out.println("Deleted vault: " + vaultName);
} catch (Exception e) {
    System.err.println(e.getMessage());
}
```

> **Note**
>
> Untuk informasi tentang REST API yang mendasarinya, lihat Menghapus Vault (DELETE vault).

Contoh: Menghapus Vault Menggunakan AWS SDK for Java

Untuk contoh kode pekerjaan, lihat Contoh: Membuat Vault Menggunakan AWS SDK for Java. Contoh kode Java menunjukkan operasi vault dasar, termasuk membuat dan menghapus vault.

Menghapus Vault di Amazon Glacier Menggunakan AWS SDK for .NET

Baik level tinggi maupun level rendah yang APIs disediakan oleh Amazon SDK for .NET menyediakan metode untuk menghapus vault.

Topik

- Menghapus Vault Menggunakan API Tingkat Tinggi AWS SDK for .NET
- Menghapus Vault Menggunakan API Tingkat Rendah AWS SDK for .NET

Menghapus Vault Menggunakan API Tingkat Tinggi AWS SDK for .NET

Kelas `ArchiveTransferManager` dari API tingkat tinggi menyediakan metode `DeleteVault` yang dapat Anda gunakan untuk menghapus vault.

Contoh: Menghapus Vault Menggunakan API Tingkat Tinggi AWS SDK for .NET

Untuk contoh kode pekerjaan, lihat Contoh: Operasi Vault Menggunakan API Tingkat Tinggi AWS SDK for .NET. Contoh kode C# menunjukkan operasi vault dasar, termasuk membuat dan menghapus vault.

Menghapus Vault Menggunakan API Tingkat Rendah AWS SDK for .NET

Berikut adalah langkah-langkah untuk menghapus vault menggunakan AWS SDK for .NET.

1. Buat instans dari kelas `AmazonGlacierClient` (klien).

   Anda perlu menentukan AWS Wilayah dari mana Anda ingin menghapus brankas. Semua operasi yang Anda lakukan menggunakan klien ini berlaku untuk AWS Wilayah tersebut.
2. Berikan informasi permintaan dengan membuat instans kelas `DeleteVaultRequest`.

   Anda harus memberi nama vault dan ID akun. Jika Anda tidak memberikan ID akun, ID akun yang terkait dengan kredensial yang Anda berikan untuk menandatangani permintaan akan diambil. Untuk informasi selengkapnya, lihat Menggunakan AWS SDK for .NET dengan Amazon Glacier.
3. Jalankan metode `DeleteVault` dengan menyediakan objek permintaan sebagai parameter.

   Amazon Glacier (Amazon Glacier) menghapus lemari besi hanya jika kosong. Untuk informasi selengkapnya, lihat Menghapus Vault (DELETE vault).

Potongan kode C# berikut menggambarkan langkah-langkah sebelumnya. Cuplikan mengambil informasi metadata dari vault yang ada di Region default. AWS

```
AmazonGlacier client;
client = new AmazonGlacierClient(Amazon.RegionEndpoint.USEast1);

DeleteVaultRequest request = new DeleteVaultRequest()
{
  VaultName = "*** provide vault name ***"
};

DeleteVaultResponse response = client.DeleteVault(request);
```

> **Note**
>
> Untuk informasi tentang REST API yang mendasarinya, lihat Menghapus Vault (DELETE vault).

Contoh: Menghapus Vault Menggunakan API Tingkat Rendah AWS SDK for .NET

Untuk contoh kode pekerjaan, lihat Contoh: Operasi Vault Menggunakan API Tingkat Rendah AWS SDK for .NET. Contoh kode C# menunjukkan operasi vault dasar, termasuk membuat dan menghapus vault.

Menghapus Vault di Amazon Glacier Menggunakan REST API

Untuk menghapus vault menggunakan REST API, lihat Menghapus Vault (DELETE vault).

Menghapus Vault Kosong dengan Menggunakan Konsol Amazon Glacier

> **Note**
>
> Sebelum menghapus brankas, Anda harus menghapus semua arsip yang ada di dalam brankas. Anda dapat melakukannya dengan menulis kode untuk membuat permintaan hapus arsip dengan menggunakan REST API, the AWS SDK for Java, the AWS SDK for .NET, atau dengan menggunakan AWS Command Line Interface (AWS CLI). Untuk informasi tentang menghapus arsip, lihat Langkah 5: Hapus Arsip dari Vault di Amazon Glacier.

Setelah brankas kosong, Anda dapat menghapusnya dengan menggunakan langkah-langkah berikut.

Untuk menghapus vault kosong dengan menggunakan konsol Amazon Glacier

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Glacier di Amazon Glacier Console.
2. Di bawah Pilih Wilayah, pilih AWS Region tempat vault berada.
3. Di panel navigasi kiri, pilih Vaults.
4. Dalam daftar Vaults, pilih tombol opsi di sebelah nama brankas yang ingin Anda hapus, lalu pilih Hapus di bagian atas halaman.

5. Di kotak dialog Hapus vault, konfirmasikan bahwa Anda ingin menghapus vault dengan memilih Hapus.

> **Important**
>
> Menghapus brankas tidak dapat dibatalkan.

6. Untuk memverifikasi bahwa Anda telah menghapus vault, buka daftar Vaults dan masukkan nama vault yang Anda hapus. Jika lemari besi tidak dapat ditemukan, penghapusan Anda berhasil.

Menghapus Vault di Amazon Glacier Menggunakan AWS Command Line Interface

Anda dapat menghapus kubah kosong dan tidak kosong di Amazon Glacier (Amazon Glacier) menggunakan (). AWS Command Line Interface AWS CLI

Topik

- (Prasyarat) Menyiapkan AWS CLI
- Contoh: Menghapus Brankas Kosong Menggunakan AWS CLI
- Contoh: Menghapus Vault Tidak Kosong Menggunakan AWS CLI

(Prasyarat) Menyiapkan AWS CLI

1. Unduh dan konfigurasikan AWS CLI. Untuk melakukannya, lihat topik berikut di Panduan Pengguna AWS Command Line Interface :

   Instalasi AWS Command Line Interface

   Mengkonfigurasi AWS Command Line Interface

2. Verifikasi AWS CLI pengaturan Anda dengan memasukkan perintah berikut pada prompt perintah. Perintah ini tidak memberikan kredensial secara eksplisit, sehingga kredensial profil default digunakan.

   - Coba gunakan perintah bantuan.

```
aws help
```

- Untuk mendapatkan daftar brankas Amazon Glacier pada akun yang dikonfigurasi, gunakan perintah. `list-vaults` Ganti *123456789012* dengan Akun AWS ID Anda.

```
aws glacier list-vaults --account-id 123456789012
```

- Untuk melihat data konfigurasi saat ini untuk AWS CLI, gunakan `aws configure list` perintah.

```
aws configure list
```

Contoh: Menghapus Brankas Kosong Menggunakan AWS CLI

- Gunakan perintah `delete-vault` untuk menghapus vault yang tidak berisi arsip.

- ```
  aws glacier delete-vault --vault-name awsexamplevault --account-id 111122223333
  ```

Contoh: Menghapus Vault Tidak Kosong Menggunakan AWS CLI

Amazon Glacier menghapus lemari besi hanya jika tidak ada arsip di lemari besi pada inventaris terakhir yang dihitung, dan belum ada penulisan ke lemari besi sejak inventaris terakhir. Menghapus vault nonempty adalah proses tiga langkah: mengambil arsip IDs dari laporan inventaris vault, menghapus setiap arsip, dan kemudian menghapus vault.

1. Gunakan perintah `initiate-job` untuk memulai tugas pengambilan inventaris.

   ```
   aws glacier initiate-job --vault-name awsexamplevault --account-id 111122223333 --
   job-parameters '{"Type": "inventory-retrieval"}'
   ```

   Output yang diharapkan:

   ```
   {
       "location": "/111122223333/vaults/awsexamplevault/jobs/*** jobid ***",
       "jobId": "*** jobid ***"
   }
   ```
   ```

2. Gunakan perintah `describe-job` untuk memeriksa status tugas pengambilan sebelumnya.

```
aws glacier describe-job --vault-name awsexamplevault --account-id 111122223333 --
job-id *** jobid ***
```

Output yang diharapkan:

```
{
    "InventoryRetrievalParameters": {
        "Format": "JSON"
    },
    "VaultARN": "*** vault arn ***",
    "Completed": false,
    "JobId": "*** jobid ***",
    "Action": "InventoryRetrieval",
    "CreationDate": "*** job creation date ***",
    "StatusCode": "InProgress"
}
```

3. Tunggu hingga tugas selesai.

Anda harus menunggu hingga output tugas siap diunduh. Jika Anda menyetel konfigurasi notifikasi di vault atau menetapkan topik Simple Notification Service Amazon (Amazon SNS) saat memulai pekerjaan, Amazon Glacier akan mengirimkan pesan ke topik tersebut setelah menyelesaikan pekerjaan.

Anda dapat menetapkan konfigurasi notifikasi untuk peristiwa tertentu di vault. Untuk informasi selengkapnya, lihat Mengkonfigurasi Pemberitahuan Vault di Amazon Glacier. Amazon Glacier mengirim pesan ke topik SNS yang ditentukan kapan saja peristiwa tertentu terjadi.

4. Setelah selesai, gunakan perintah `get-job-output` untuk mengunduh tugas pengambilan ke file `output.json`.

```
aws glacier get-job-output --vault-name awsexamplevault --account-id 111122223333
 --job-id *** jobid *** output.json
```

Perintah ini menghasilkan file dengan bidang berikut.

```
{
"VaultARN":"arn:aws:glacier:region:111122223333:vaults/awsexamplevault",
"InventoryDate":"*** job completion date ***",
"ArchiveList":[
{"ArchiveId":"*** archiveid ***",
"ArchiveDescription":*** archive description (if set) ***,
"CreationDate":"*** archive creation date ***",
"Size":"*** archive size (in bytes) ***",
"SHA256TreeHash":"*** archive hash ***"
}
{"ArchiveId":
...
]}
```

5. Gunakan perintah `delete-archive` untuk menghapus setiap arsip dari vault sampai tidak ada yang tersisa.

```
aws glacier delete-archive --vault-name awsexamplevault --account-id 111122223333
 --archive-id "*** archiveid ***"
```

> **Note**
>
> Jika ID arsip Anda dimulai dengan tanda hubung atau karakter khusus lainnya, Anda harus melampirkan ID arsip dalam tanda kutip untuk menjalankan perintah ini.

6. Gunakan perintah `initiate-job` untuk memulai tugas pengambilan inventaris baru.

```
aws glacier initiate-job --vault-name awsexamplevault --account-id 111122223333 --
job-parameters '{"Type": "inventory-retrieval"}'
```

7. Setelah selesai, gunakan perintah `delete-vault` untuk menghapus vault tanpa arsip.

```
aws glacier delete-vault --vault-name awsexamplevault --account-id 111122223333
```

Menandai Brankas Amazon Glacier Anda

Anda dapat menetapkan metadata Anda sendiri ke brankas Amazon Glacier dalam bentuk tag. Tanda adalah pasangan nilai-kunci yang Anda tetapkan untuk vault. Untuk informasi dasar tentang penandaan, termasuk pembatasan tag, lihat Menandai Sumber Daya Amazon Glacier.

Topik berikut menjelaskan bagaimana Anda dapat menambahkan, mencantumkan, dan menghapus tanda untuk vault.

Topik

- Menandai Vaults dengan Menggunakan Konsol Amazon Glacier
- Menandai Vaults dengan Menggunakan AWS CLI
- Menandai Vaults dengan Menggunakan Amazon Glacier API
- Bagian Terkait

Menandai Vaults dengan Menggunakan Konsol Amazon Glacier

Anda dapat menambahkan, membuat daftar, dan menghapus tag menggunakan konsol Amazon Glacier, seperti yang dijelaskan dalam prosedur berikut.

Untuk melihat tanda untuk vault

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Glacier di rumah. https://console.aws.amazon.com/glacier/
2. Di bawah Pilih Wilayah, pilih AWS Region dari pemilih Wilayah.
3. Di panel navigasi kiri, pilih Vaults.
4. Dalam daftar Vaults, pilih brankas.
5. Pilih tab properti Vaults. Gulir ke bagian Tag untuk melihat tag yang terkait dengan brankas.

Untuk menambahkan tanda ke vault

Anda dapat mengaitkan hingga 50 tag ke brankas. Tag yang terkait dengan vault harus memiliki kunci tag yang unik.

Untuk informasi selengkapnya tentang pembatasan tag, lihat Menandai Sumber Daya Amazon Glacier.

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Glacier di rumah. https://console.aws.amazon.com/glacier/
2. Di bawah Pilih Wilayah, pilih AWS Region dari pemilih Wilayah.
3. Di panel navigasi kiri, pilih Vaults.
4. Dalam daftar Vaults, pilih nama brankas yang ingin Anda tambahkan tag.
5. Pilih tab properti Vault.
6. Di bagian Tag, pilih Tambah. Laman Tambah tag akan muncul.
7. Pada halaman Tambahkan tag, tentukan kunci tag di bidang Kunci, dan secara opsional tentukan nilai tag di bidang Nilai.
8. Pilih Simpan perubahan.

Untuk mengedit tag

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Glacier di rumah. https://console.aws.amazon.com/glacier/
2. Di bawah Pilih Wilayah, pilih AWS Region dari pemilih Wilayah.
3. Di panel navigasi kiri, pilih Vaults.
4. Dalam daftar Vaults, pilih nama vault.
5. Pilih tab Properti Vault, lalu gulir ke bawah ke bagian Tag.
6. Di bawah Tag, pilih kotak centang di samping tag yang ingin Anda ubah, lalu pilih Edit. Halaman Edit tag muncul.
7. Perbarui kunci tag di bidang Kunci, dan opsional memperbarui nilai tag di bidang Nilai.
8. Pilih Simpan perubahan.

Untuk menghapus tag dari vault

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Glacier di rumah. https://console.aws.amazon.com/glacier/
2. Di bawah Pilih Wilayah, pilih AWS Region dari pemilih Wilayah.
3. Di panel navigasi kiri, pilih Vaults.
4. Dalam daftar Vaults, pilih nama brankas yang ingin Anda hapus tag.
5. Pilih tab properti Vault. Gulir ke bawah ke bagian Tag.
6. Di bawah Tag, pilih kotak centang di samping tag yang ingin Anda hapus, lalu pilih Hapus.

7. Kotak dialog Hapus tag terbuka. Untuk mengonfirmasi bahwa Anda ingin menghapus tag yang dipilih, pilih Hapus.

Menandai Vaults dengan Menggunakan AWS CLI

Ikuti langkah-langkah ini untuk menambah, membuat daftar, atau menghapus tag dengan menggunakan AWS Command Line Interface (AWS CLI).

Setiap tanda terdiri dari kunci dan nilai. Setiap vault dapat memiliki hingga 50 tanda.

1. Untuk menambahkan tag ke brankas, gunakan `add-tags-to-vault` perintah.

   ```
   aws glacier add-tags-to-vault --vault-name examplevault --account-id 111122223333
    --tags id=1234,date=2020
   ```

   Untuk informasi selengkapnya tentang operasi vault ini, lihat Menambahkan Tag Ke Vault.
2. Untuk mencantumkan semua tag yang dilampirkan ke lemari besi, gunakan `list-tags-for-vault` perintah.

   ```
   aws glacier list-tags-for-vault --vault-name examplevault --account-id 111122223333
   ```

   Untuk informasi selengkapnya tentang operasi vault ini, lihat Daftar Tag Untuk Vault.
3. Untuk menghapus satu atau beberapa tag dari kumpulan tag yang dilampirkan ke lemari besi, gunakan `remove-tags-from-vault` perintah.

   ```
   aws glacier remove-tags-from-vault --vault-name examplevault --account-
   id 111122223333 --tag-keys date
   ```

   Untuk informasi selengkapnya tentang operasi vault ini, lihat Menghapus Tag Dari Vault.

Menandai Vaults dengan Menggunakan Amazon Glacier API

Anda dapat menambahkan, membuat daftar, dan menghapus tag dengan menggunakan Amazon Glacier API. Untuk contoh, lihat dokumentasi berikut:

Menambahkan Tanda ke Vault (POST tag add)

Menambahkan atau memperbarui tanda untuk vault tertentu.

Mencantumkan Tanda untuk Vault (GET tags)

Mencantumkan tanda untuk vault tertentu.

Hapus Tanda Dari Vault (POST tags remove)

Menghapus tanda dari vault tertentu.

Bagian Terkait

- Menandai Sumber Daya Amazon Glacier

Kunci Gletser Amazon Glacier

Topik berikut menjelaskan cara mengunci brankas di Amazon Glacier dan cara menggunakan kebijakan Vault Lock.

Topik

- Gambaran Umum Penguncian Vault
- Mengunci Vault dengan Menggunakan Amazon Glacier API
- Mengunci Vault menggunakan AWS Command Line Interface
- Mengunci Vault dengan Menggunakan Konsol Amazon Glacier

Gambaran Umum Penguncian Vault

Amazon Glacier Vault Lock membantu Anda menerapkan dan menerapkan kontrol kepatuhan dengan mudah untuk kubah Amazon Glacier individual dengan kebijakan Vault Lock. Anda dapat menentukan kontrol seperti “tulis sekali baca banyak” (WORM) dalam kebijakan Vault Lock dan mengunci kebijakan dari pengeditan di masa mendatang.

> **Important**
>
> Setelah kebijakan Vault Lock dikunci, kebijakan tidak dapat lagi diubah atau dihapus.

Amazon Glacier memberlakukan kontrol yang ditetapkan dalam kebijakan Vault Lock untuk membantu mencapai tujuan kepatuhan Anda. Misalnya, Anda dapat menggunakan kebijakan Vault

Lock untuk menerapkan penyimpanan data. Anda dapat menerapkan berbagai kontrol kepatuhan dalam kebijakan Vault Lock menggunakan bahasa kebijakan AWS Identity and Access Management (IAM). Untuk informasi selengkapnya tentang kebijakan Vault Lock, lihatKebijakan Vault Lock.

Kebijakan Vault Lock berbeda dengan kebijakan akses vault. Kedua kebijakan mengatur kontrol akses ke vault Anda. Namun, kebijakan Vault Lock dapat dikunci untuk mencegah perubahan di masa mendatang, yang memberikan penegakan hukum yang kuat untuk kontrol kepatuhan Anda. Anda dapat menggunakan kebijakan Vault Lock untuk menerapkan kontrol regulasi dan kepatuhan, yang biasanya memerlukan kontrol ketat pada akses data.

> **Important**
>
> Kami menyarankan Anda terlebih dahulu membuat vault, menyelesaikan kebijakan Vault Lock, lalu mengunggah arsip Anda ke brankas sehingga kebijakan tersebut akan diterapkan padanya.

Sebaliknya, Anda menggunakan kebijakan akses vault untuk menerapkan kontrol akses yang tidak sesuai, sementara, dan tunduk pada perubahan yang sering. Anda dapat menggunakan kebijakan kunci Vault dan akses vault secara bersamaan. Misalnya, Anda dapat menerapkan aturan penyimpanan data berbasis waktu dalam kebijakan Vault Lock (tolak penghapusan), dan memberikan akses baca ke pihak ketiga yang ditunjuk atau mitra bisnis Anda (izinkan pembacaan) dalam kebijakan akses vault Anda.

Mengunci vault memerlukan dua langkah:

1. Memulai kunci dengan melampirkan kebijakan Vault Lock ke vault Anda, yang menyetel kunci ke status sedang berlangsung dan mengembalikan ID kunci. Saat kebijakan sedang berlangsung, Anda memiliki waktu 24 jam untuk memvalidasi kebijakan Vault Lock sebelum ID kunci kedaluwarsa. Untuk mencegah vault Anda keluar dari status sedang berlangsung, Anda harus menyelesaikan proses Vault Lock dalam 24 jam ini. Jika tidak, kebijakan Vault Lock Anda akan dihapus.
2. Gunakan ID kunci untuk menyelesaikan proses penguncian. Jika kebijakan Vault Lock tidak berfungsi seperti yang diharapkan, Anda dapat menghentikan proses Vault Lock dan memulai ulang dari awal. Untuk informasi tentang cara menggunakan Amazon Glacier API untuk mengunci vault, lihat. Mengunci Vault dengan Menggunakan Amazon Glacier API

Mengunci Vault dengan Menggunakan Amazon Glacier API

Untuk mengunci vault Anda dengan Amazon Glacier API, Memulai Kunci Vault (POST lock-policy) pertama-tama Anda menelepon dengan kebijakan Vault Lock yang menentukan kontrol yang ingin Anda terapkan. `Initiate Vault Lock`Operasi melampirkan kebijakan ke vault Anda, mentransisikan Kunci Vault ke status sedang berlangsung, dan mengembalikan ID kunci unik. Setelah Vault Lock memasuki status dalam proses, Anda memiliki waktu 24 jam untuk menyelesaikan kunci dengan memanggil Menyelesaikan Kunci Vault (POST lockId) dengan ID kunci yang dikembalikan dari panggilan. `Initiate Vault Lock`

> **Important**
>
> - Kami menyarankan Anda terlebih dahulu membuat vault, menyelesaikan kebijakan Vault Lock, lalu mengunggah arsip Anda ke brankas sehingga kebijakan tersebut akan diterapkan padanya.
> - Setelah kebijakan Vault Lock terkunci, kebijakan tersebut tidak dapat diubah atau dihapus.

Jika Anda tidak menyelesaikan proses Vault Lock dalam waktu 24 jam setelah memasuki status dalam proses, vault Anda secara otomatis keluar dari status sedang berlangsung, dan kebijakan Kunci Vault dihapus. Anda dapat menelepon `Initiate Vault Lock` lagi untuk menginstal kebijakan Vault Lock baru dan beralih ke status sedang berlangsung.

Status yang sedang berlangsung memberikan kesempatan untuk menguji kebijakan Kunci Vault Anda sebelum Anda menguncinya. Kebijakan Kunci Vault Anda berlaku penuh selama status sedang berlangsung seperti jika vault telah dikunci, kecuali Anda dapat menghapus kebijakan dengan menelepon. Membatalkan Kunci Vault (DELETE lock-policy) Untuk menyempurnakan kebijakan Anda, Anda dapat mengulangi `Initiate Vault Lock` kombinasi`Abort Vault Lock`/sebanyak yang diperlukan untuk memvalidasi perubahan kebijakan Vault Lock.

Setelah memvalidasi kebijakan Vault Lock, Anda dapat menelepon Menyelesaikan Kunci Vault (POST lockId) dengan ID kunci terbaru untuk menyelesaikan proses penguncian vault. Vault Anda bertransisi ke status terkunci, di mana kebijakan Kunci Vault tidak dapat diubah dan tidak dapat lagi dihapus dengan menelepon. `Abort Vault Lock`

Bagian Terkait

- Kebijakan Vault Lock
- Membatalkan Kunci Vault (DELETE lock-policy)
- Menyelesaikan Kunci Vault (POST lockId)
- Mendapatkan Kunci Vault (GET lock-policy)
- Memulai Kunci Vault (POST lock-policy)

Mengunci Vault menggunakan AWS Command Line Interface

Anda dapat mengunci lemari besi Anda menggunakan. AWS Command Line Interface Ini akan menginstal kebijakan kunci vault pada brankas yang ditentukan dan mengembalikan ID kunci. Anda harus menyelesaikan proses penguncian vault dalam waktu 24 jam jika tidak, kebijakan kunci vault dihapus dari lemari besi.

(Prasyarat) Menyiapkan AWS CLI

1. Unduh dan konfigurasikan AWS CLI. Untuk melakukannya, lihat topik berikut di Panduan Pengguna AWS Command Line Interface :

   Memasang AWS Command Line Interface

   Mengkonfigurasi AWS Command Line Interface
2. Verifikasi AWS CLI pengaturan Anda dengan memasukkan perintah berikut pada prompt perintah. Perintah ini tidak memberikan kredensial secara eksplisit, sehingga kredensial profil default digunakan.

   - Coba gunakan perintah bantuan.

     ```
     aws help
     ```

   - Untuk mendapatkan daftar brankas Amazon Glacier pada akun yang dikonfigurasi, gunakan perintah. `list-vaults` Ganti *123456789012* dengan Akun AWS ID Anda.

     ```
     aws glacier list-vaults --account-id 123456789012
     ```

   - Untuk melihat data konfigurasi saat ini untuk AWS CLI, gunakan `aws configure list` perintah.

     ```
     aws configure list
     ```

1. Gunakan `initiate-vault-lock` untuk menginstal kebijakan kunci vault dan atur status kunci kunci vault. `InProgress`

```
aws glacier initiate-vault-lock --vault-name examplevault --account-id 111122223333
 --policy file://lockconfig.json
```

2. Konfigurasi kunci adalah dokumen JSON seperti yang ditunjukkan pada contoh berikut. Sebelum menggunakan perintah ini, ganti *VAULT_ARN* dan *Principal* dengan nilai yang sesuai untuk kasus penggunaan Anda.

   Untuk menemukan ARN dari lemari besi yang ingin Anda kunci, Anda dapat menggunakan perintah. `list-vaults`

```
{"Policy":"{\"Version\":\"2012-10-17\",       \"Statement\":[{\"Sid
\":\"Define-vault-lock\",\"Effect\":\"Deny\",\"Principal\":{\"AWS\":
\"arn:aws:iam::111122223333:root\"},\"Action\":\"glacier:DeleteArchive
\",\"Resource\":\"VAULT_ARN\",\"Condition\":{\"NumericLessThanEquals\":
{\"glacier:ArchiveAgeinDays\":\"365\"}}}]}"}
```

3. Setelah memulai kunci brankas, Anda akan melihat yang dikembalikan. `lockId`

```
{
    "lockId": "LOCK_ID"
}
```

Untuk menyelesaikan kunci vault Anda harus menjalankan `complete-vault-lock` dalam waktu 24 jam jika tidak kebijakan kunci vault dihapus dari lemari besi.

```
aws glacier complete-vault-lock --vault-name examplevault --account-id 111122223333 --
lock-id LOCK_ID
```

Bagian Terkait

- initiate-vault-lockdalam Referensi AWS CLI Perintah
- daftar-vault di Referensi Perintah AWS CLI
- complete-vault-lockdalam Referensi AWS CLI Perintah
- Kebijakan Vault Lock
- Membatalkan Kunci Vault (DELETE lock-policy)

- Menyelesaikan Kunci Vault (POST lockId)
- Mendapatkan Kunci Vault (GET lock-policy)
- Memulai Kunci Vault (POST lock-policy)

Mengunci Vault dengan Menggunakan Konsol Amazon Glacier

Amazon Glacier Vault Lock membantu Anda menerapkan dan menerapkan kontrol kepatuhan dengan mudah untuk kubah Amazon Glacier individual dengan kebijakan Vault Lock. Untuk informasi selengkapnya tentang Amazon Glacier Vault Lock, lihat Kontrol Akses Amazon Glacier dengan Kebijakan Kunci Vault.

> ⚠ Important
>
> - Kami menyarankan Anda terlebih dahulu membuat vault, menyelesaikan kebijakan Vault Lock, lalu mengunggah arsip Anda ke brankas sehingga kebijakan tersebut akan diterapkan padanya.
> - Setelah kebijakan Vault Lock terkunci, kebijakan tersebut tidak dapat diubah atau dihapus.

Untuk memulai kebijakan Vault Lock di vault Anda dengan menggunakan konsol Amazon Glacier

Anda memulai kunci dengan melampirkan kebijakan Vault Lock ke vault Anda, yang menyetel kunci ke status sedang berlangsung dan mengembalikan ID kunci. Saat kebijakan sedang berlangsung, Anda memiliki waktu 24 jam untuk memvalidasi kebijakan Vault Lock sebelum ID kunci kedaluwarsa.

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Glacier di rumah. https://console.aws.amazon.com/glacier/
2. Di bawah Pilih Wilayah, pilih AWS Region dari pemilih Wilayah.
3. Di panel navigasi kiri, pilih Vaults.
4. Pada halaman Vaults, pilih Create vault.
5. Buat lemari besi baru.

> **Important**
>
> Kami menyarankan Anda terlebih dahulu membuat vault, menyelesaikan kebijakan Vault Lock, lalu mengunggah arsip Anda ke brankas sehingga kebijakan tersebut akan diterapkan padanya.

6. Pilih lemari besi baru Anda dari daftar Vaults.
7. Pilih tab Kebijakan Vault.
8. Di bagian kebijakan Vault Lock, pilih kebijakan Initiate Vault Lock.
9. Pada halaman kebijakan Initiate Vault Lock, tentukan kontrol penyimpanan rekaman dalam kebijakan Kunci Vault Anda dalam format teks di kotak teks standar.

> **Note**
>
> Anda dapat menentukan kontrol penyimpanan rekaman dalam kebijakan Vault Lock dalam format teks dan memulai Vault Lock dengan memanggil operasi `Initiate Vault Lock` API atau melalui UI interaktif di konsol Amazon Glacier. Untuk informasi tentang memformat kebijakan Kunci Vault, lihat Contoh Kebijakan Kunci Amazon Glacier Vault.

10. Pilih Simpan perubahan.
11. Dalam kotak dialog Record Vault Lock ID, salin ID Kunci Anda dan simpan di tempat yang aman.

> **Important**
>
> Setelah kebijakan Vault Lock dimulai, Anda memiliki waktu 24 jam untuk memvalidasi kebijakan dan menyelesaikan proses penguncian. Untuk menyelesaikan proses penguncian, Anda harus memberikan ID kunci. Jika tidak diberikan dalam waktu 24 jam, ID kunci akan kedaluwarsa dan kebijakan Anda yang sedang berlangsung akan dihapus.

12. Setelah menyimpan ID kunci Anda di tempat yang aman, pilih Tutup.
13. Uji kebijakan Vault Lock Anda dalam 24 jam ke depan. Jika kebijakan berfungsi sebagaimana dimaksud, pilih kebijakan Complete Vault Lock.
14. Di kotak dialog Complete Vault Lock, pilih kotak centang untuk mengetahui bahwa menyelesaikan proses kebijakan Vault Lock tidak dapat diubah.

15. Masukkan ID Kunci yang Anda berikan di kotak teks.
16. Pilih Complete Vault Lock.

Bekerja dengan Arsip di Amazon Glacier

Arsip adalah objek apa pun, seperti foto, video, atau dokumen yang Anda simpan di vault. Ini adalah unit dasar penyimpanan di Amazon Glacier (Amazon Glacier). Setiap arsip memiliki ID unik dan deskripsi opsional. Saat Anda mengunggah arsip, Amazon Glacier menampilkan respons yang menyertakan ID arsip. ID arsip ini unik di AWS Wilayah tempat arsip disimpan. Berikut adalah contoh ID arsip.

```
TJgHcrOSfAkV6hdPqOATYfp_0ZaxL1pIBOc02iZ0gDPMr2ig-
nhwd_PafstsdIf6HSrjHnP-3p6LCJClYytFT_CBhT9CwNxbRaM5MetS3I-
GqwxI3Y8QtgbJbhEQPs0mJ3KExample
```

Arsip IDs memiliki panjang 138 byte. Saat mengunggah arsip, Anda dapat memberikan deskripsi opsional. Anda dapat mengambil arsip menggunakan ID-nya, tetapi tidak deskripsinya.

> **Important**
>
> Amazon Glacier menyediakan konsol manajemen. Anda dapat menggunakan konsol untuk membuat dan menghapus vault. Namun, semua interaksi lain dengan Amazon Glacier mengharuskan Anda menggunakan AWS Command Line Interface (CLI) atau menulis kode. Misalnya, untuk mengunggah data, seperti foto, video, dan dokumen lainnya, Anda harus menggunakan AWS CLI atau menulis kode untuk membuat permintaan, baik menggunakan REST API secara langsung atau dengan menggunakan Amazon SDKs. Untuk informasi lebih lanjut tentang menggunakan Amazon Glacier dengan AWS CLI, buka Referensi AWS CLI untuk Amazon Glacier. Untuk menginstal AWS CLI, pergi ke AWS Command Line Interface.

Topik

- Arsip Operasi di Amazon Glacier
- Mempertahankan Metadata Arsip Sisi Klien
- Mengunggah Arsip di Amazon Glacier
- Mengunduh Arsip di Amazon Glacier
- Menghapus Arsip di Amazon Glacier

Arsip Operasi di Amazon Glacier

Amazon Glacier mendukung operasi arsip dasar berikut: unggah, unduh, dan hapus. Mengunduh arsip adalah operasi asinkron.

Mengunggah Arsip di Amazon Glacier

Anda dapat mengunggah arsip dalam satu operasi atau mengunggahnya dalam beberapa bagian. Panggilan API yang Anda gunakan untuk mengunggah arsip dalam beberapa bagian disebut sebagai Unggahan Multipart. Untuk informasi selengkapnya, lihat Mengunggah Arsip di Amazon Glacier.

> **Important**
>
> Amazon Glacier menyediakan konsol manajemen. Anda dapat menggunakan konsol untuk membuat dan menghapus vault. Namun, semua interaksi lain dengan Amazon Glacier mengharuskan Anda menggunakan AWS Command Line Interface (CLI) atau menulis kode. Misalnya, untuk mengunggah data, seperti foto, video, dan dokumen lainnya, Anda harus menggunakan AWS CLI atau menulis kode untuk membuat permintaan, baik menggunakan REST API secara langsung atau dengan menggunakan Amazon SDKs. Untuk informasi lebih lanjut tentang menggunakan Amazon Glacier dengan AWS CLI, buka Referensi AWS CLI untuk Amazon Glacier. Untuk menginstal AWS CLI, pergi ke AWS Command Line Interface.

Menemukan ID Arsip di Amazon Glacier

Anda bisa mendapatkan ID arsip dengan mengunduh inventaris vault untuk vault yang berisi arsip. Untuk informasi selengkapnya tentang mengunduh inventaris vault, lihat Mengunduh Inventaris Vault di Amazon Glacier.

Mengunduh Arsip di Amazon Glacier

Mengunduh arsip adalah operasi asinkron. Anda pertama-tama harus memulai tugas untuk mengunduh arsip tertentu. Setelah menerima permintaan pekerjaan, Amazon Glacier menyiapkan arsip Anda untuk diunduh. Setelah tugas selesai, Anda dapat mengunduh data arsip. Karena sifat pekerjaan yang tidak sinkron, Anda dapat meminta Amazon Glacier untuk mengirim pemberitahuan ke topik Simple Notification Service Amazon (Amazon SNS) saat pekerjaan selesai. Anda dapat menentukan topik SNS untuk setiap permintaan tugas individu atau mengonfigurasi vault Anda

untuk mengirim notifikasi ketika peristiwa tertentu terjadi. Untuk informasi selengkapnya tentang mengunduh arsip, lihat Mengunduh Arsip di Amazon Glacier.

Menghapus Arsip di Amazon Glacier

Amazon Glacier menyediakan panggilan API yang dapat Anda gunakan untuk menghapus satu arsip sekaligus. Untuk informasi selengkapnya, lihat Menghapus Arsip di Amazon Glacier.

Memperbarui Arsip di Amazon Glacier

Setelah mengunggah arsip, Anda tidak dapat memperbarui konten atau deskripsinya. Satu-satunya cara agar Anda dapat memperbarui konten arsip atau deskripsinya adalah dengan menghapus arsip dan mengunggah arsip lain. Perhatikan bahwa setiap kali Anda mengunggah arsip, Amazon Glacier mengembalikan ID arsip unik kepada Anda.

Mempertahankan Metadata Arsip Sisi Klien

Kecuali untuk deskripsi arsip opsional, Amazon Glacier tidak mendukung metadata tambahan apa pun untuk arsip. Saat Anda mengunggah arsip Amazon Glacier memberikan ID, urutan karakter buram, yang darinya Anda tidak dapat menyimpulkan arti apa pun tentang arsip tersebut. Anda dapat mempertahankan metadata tentang arsip di sisi klien. Metadata dapat mencakup nama arsip dan beberapa informasi bermakna lainnya tentang arsip.

> **Note**
>
> Jika Anda adalah pelanggan Amazon Simple Storage Service (Amazon S3), Anda tahu bahwa ketika Anda mengunggah objek ke bucket, Anda dapat menetapkan objek kunci untuk objek seperti `MyDocument.txt` atau `SomePhoto.jpg`. Di Amazon Glacier, Anda tidak dapat menetapkan kunci objek ke arsip yang Anda unggah.

Jika Anda mempertahankan metadata arsip sisi klien, perhatikan bahwa Amazon Glacier menyimpan inventaris vault yang menyertakan arsip IDs dan deskripsi apa pun yang Anda berikan selama unggahan arsip. Anda terkadang dapat mengunduh inventaris vault untuk menyelesaikan masalah di basis data sisi klien yang Anda pertahankan untuk metadata arsip. Namun, Amazon Glacier mengambil inventaris lemari besi kira-kira setiap hari. Saat Anda meminta inventaris brankas, Amazon Glacier mengembalikan inventaris terakhir yang disiapkan, snapshot titik waktu.

Mengunggah Arsip di Amazon Glacier

Amazon Glacier (Amazon Glacier) menyediakan konsol manajemen, yang dapat Anda gunakan untuk membuat dan menghapus brankas. Namun, Anda tidak dapat mengunggah arsip ke Amazon Glacier dengan menggunakan konsol manajemen. Untuk mengunggah data, seperti foto, video, dan dokumen lainnya, Anda harus menggunakan AWS CLI atau menulis kode untuk membuat permintaan, dengan menggunakan REST API secara langsung atau dengan menggunakan Amazon SDKs.

Untuk informasi tentang menggunakan Amazon Glacier dengan AWS CLI, buka Referensi AWS CLI untuk Amazon Glacier. Untuk menginstal AWS CLI, pergi ke AWS Command Line Interface. Topik Pengunggahan berikut menjelaskan cara mengunggah arsip ke Amazon Glacier dengan menggunakan Amazon SDK for Java, Amazon SDK for .NET, dan REST API.

Topik

- Opsi untuk Mengunggah Arsip ke Amazon Glacier
- Mengunggah Arsip dalam Satu Operasi
- Mengunggah Arsip Besar dalam Beberapa Bagian (Unggahan Multipart)

Opsi untuk Mengunggah Arsip ke Amazon Glacier

Bergantung pada ukuran data yang Anda unggah, Amazon Glacier menawarkan opsi berikut:

- Unggah arsip dalam satu operasi – Dalam satu operasi, Anda dapat mengunggah arsip berukuran mulai dari 1 byte hingga 4 GB. Namun, kami mendorong pelanggan Amazon Glacier untuk menggunakan unggahan multipart untuk mengunggah arsip lebih dari 100 MB. Untuk informasi selengkapnya, lihat Mengunggah Arsip dalam Satu Operasi.
- Unggah arsip dalam beberapa bagian – Dengan menggunakan API unggahan multipart, Anda dapat mengunggah arsip besar, hingga sekitar 40.000 GB (10.000 * 4 GB).

  Panggilan API unggahan multipart dirancang untuk meningkatkan pengalaman unggah untuk arsip yang lebih besar. Anda dapat mengunggah arsip dalam beberapa bagian. Bagian-bagian ini dapat diunggah secara independen, dalam urutan apa pun, dan secara paralel. Jika satu bagian gagal diunggah, Anda hanya perlu mengunggah kembali bagian tersebut dan bukan seluruh arsip. Anda dapat menggunakan unggahan multipart untuk arsip berukuran mulai dari 1 byte hingga sekitar 40.000 GB. Untuk informasi selengkapnya, lihat Mengunggah Arsip Besar dalam Beberapa Bagian (Unggahan Multipart).

> **Important**
>
> Inventaris lemari besi Amazon Glacier hanya diperbarui sekali sehari. Saat mengunggah arsip, Anda tidak akan segera melihat arsip baru yang ditambahkan ke vault Anda (di konsol atau di daftar inventaris vault yang diunduh) hingga inventaris vault diperbarui.

Menggunakan AWS Snowball Edge Layanan

AWS Snowball Edge mempercepat pemindahan sejumlah besar data masuk dan keluar AWS menggunakan perangkat milik Amazon, melewati internet. Untuk informasi selengkapnya, lihat halaman detail AWS Snowball Edge.

Untuk mengunggah data yang ada ke Amazon Glacier (Amazon Glacier), Anda dapat mempertimbangkan untuk menggunakan salah satu jenis AWS Snowball Edge perangkat untuk mengimpor data ke Amazon S3, lalu memindahkannya ke kelas penyimpanan Amazon Glacier untuk arsip menggunakan aturan siklus hidup. Saat Anda mentransisikan objek Amazon S3 ke kelas penyimpanan Amazon Glacier, Amazon S3 secara internal menggunakan Amazon Glacier untuk penyimpanan tahan lama dengan biaya lebih rendah. Meskipun objek disimpan di Amazon Glacier, mereka tetap menjadi objek Amazon S3 yang Anda kelola di Amazon S3, dan Anda tidak dapat mengaksesnya langsung melalui Amazon Glacier.

Untuk informasi selengkapnya tentang konfigurasi siklus hidup Amazon S3 dan objek transisi ke kelas penyimpanan Amazon Glacier, lihat Objek Manajemen dan Transisi Siklus Hidup Objek di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Mengunggah Arsip dalam Satu Operasi

Seperti yang dijelaskan di Mengunggah Arsip di Amazon Glacier, Anda dapat mengunggah arsip yang lebih kecil dalam satu operasi. Namun, kami mendorong pelanggan Amazon Glacier (Amazon Glacier) untuk menggunakan Unggahan Multipart untuk mengunggah arsip lebih dari 100 MB.

Topik

- Mengunggah Arsip dalam Satu Operasi Menggunakan AWS Command Line Interface
- Mengunggah Arsip dalam Satu Operasi Menggunakan AWS SDK for Java
- Mengunggah Arsip dalam Satu Operasi Menggunakan Amazon Glacier AWS SDK for .NET di Amazon

- Mengunggah Arsip dalam Satu Operasi Menggunakan REST API

Mengunggah Arsip dalam Satu Operasi Menggunakan AWS Command Line Interface

Anda dapat mengunggah arsip di Amazon Glacier (Amazon Glacier) menggunakan (). AWS Command Line Interface AWS CLI

Topik

- (Prasyarat) Menyiapkan AWS CLI
- Contoh: Unggah Arsip Menggunakan AWS CLI

(Prasyarat) Menyiapkan AWS CLI

1. Unduh dan konfigurasikan AWS CLI. Untuk melakukannya, lihat topik berikut di Panduan Pengguna AWS Command Line Interface :

   Instalasi AWS Command Line Interface

   Mengkonfigurasi AWS Command Line Interface
2. Verifikasi AWS CLI pengaturan Anda dengan memasukkan perintah berikut pada prompt perintah. Perintah ini tidak memberikan kredensial secara eksplisit, sehingga kredensial profil default digunakan.

   - Coba gunakan perintah bantuan.

     ```
     aws help
     ```

   - Untuk mendapatkan daftar brankas Amazon Glacier pada akun yang dikonfigurasi, gunakan perintah. `list-vaults` Ganti *123456789012* dengan Akun AWS ID Anda.

     ```
     aws glacier list-vaults --account-id 123456789012
     ```

   - Untuk melihat data konfigurasi saat ini untuk AWS CLI, gunakan `aws configure list` perintah.

     ```
     aws configure list
     ```

Contoh: Unggah Arsip Menggunakan AWS CLI

Untuk mengunggah arsip, Anda harus membuat brankas. Untuk informasi selengkapnya tentang membuat brankas, lihat. Membuat Vault di Amazon Glacier

1. Gunakan `upload-archive` perintah untuk menambahkan arsip ke brankas yang ada. Pada contoh di bawah ini ganti `vault name` dan`account ID`. Untuk `body` parameter tentukan jalur ke file yang ingin Anda unggah.

```
aws glacier upload-archive --vault-name awsexamplevault --account-id 123456789012
 --body archive.zip
```

2. Keluaran yang diharapkan

```
{
    "archiveId": "kKB7ymWJVpPSwhGP6ycSOAekp9ZYe_--zM_mw6k76ZFGEIWQX-
ybtRDvc2VkPSDtfKmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hHO0bjbGehXTcApVud_wyDw",
    "checksum": "969fb39823836d81f0cc028195fcdbcbbe76cdde932d4646fa7de5f21e18aa67",
    "location": "/123456789012/vaults/awsexamplevault/archives/
kKB7ymWJVpPSwhGP6ycSOAekp9ZYe_--zM_mw6k76ZFGEIWQX-ybtRDvc2VkPSDtfKmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hHO0bjbGehXTcApVud_wyDw"
}
```

   Setelah selesai perintah akan menampilkan ID arsip, checksum, dan lokasi di Amazon Glacier. Untuk informasi selengkapnya tentang perintah upload-archive, lihat upload-archive di Command Reference.AWS CLI

Mengunggah Arsip dalam Satu Operasi Menggunakan AWS SDK for Java

Baik tingkat tinggi dan tingkat rendah yang APIs disediakan oleh Amazon SDK for Java menyediakan metode untuk mengunggah arsip.

Topik

- Mengunggah Arsip Menggunakan API Tingkat Tinggi AWS SDK for Java
- Mengunggah Arsip dalam Satu Operasi Menggunakan API Tingkat Rendah AWS SDK for Java

Mengunggah Arsip Menggunakan API Tingkat Tinggi AWS SDK for Java

Kelas `ArchiveTransferManager` dari API tingkat tinggi menyediakan metode `upload`, yang dapat Anda gunakan untuk mengunggah arsip ke vault.

> **Note**
>
> Anda dapat menggunakan metode `upload` untuk mengunggah arsip kecil atau besar. Bergantung pada ukuran arsip yang Anda unggah, metode ini menentukan apakah akan mengunggahnya dalam satu operasi atau menggunakan API unggahan multipart untuk mengunggah arsip dalam beberapa bagian.

Contoh: Mengunggah Arsip Menggunakan API Tingkat Tinggi AWS SDK for Java

Contoh kode Java berikut mengunggah arsip ke vault (`examplevault`) di Wilayah US West (Oregon) (`us-west-2`). Untuk daftar AWS Wilayah dan titik akhir yang didukung, lihatMengakses Amazon Glacier.

Untuk step-by-step petunjuk tentang cara menjalankan contoh ini, lihatMenjalankan Contoh Java untuk Amazon Glacier Menggunakan Eclipse. Anda perlu memperbarui kode seperti yang ditunjukkan dengan nama vault yang ingin Anda unggah dan nama file yang ingin Anda unggah.

Example

```
import java.io.File;
import java.io.IOException;
import java.util.Date;

import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.glacier.AmazonGlacierClient;
import com.amazonaws.services.glacier.transfer.ArchiveTransferManager;
import com.amazonaws.services.glacier.transfer.UploadResult;


public class ArchiveUploadHighLevel {
    public static String vaultName = "*** provide vault name ***";
    public static String archiveToUpload = "*** provide name of file to upload ***";

    public static AmazonGlacierClient client;
```

```
    public static void main(String[] args) throws IOException {


     ProfileCredentialsProvider credentials = new ProfileCredentialsProvider();

        client = new AmazonGlacierClient(credentials);
        client.setEndpoint("https://glacier.us-west-2.amazonaws.com/");

        try {
            ArchiveTransferManager atm = new ArchiveTransferManager(client,
 credentials);

            UploadResult result = atm.upload(vaultName, "my archive " + (new Date()),
 new File(archiveToUpload));
            System.out.println("Archive ID: " + result.getArchiveId());

        } catch (Exception e)
        {
            System.err.println(e);
        }
    }
}
```

Mengunggah Arsip dalam Satu Operasi Menggunakan API Tingkat Rendah AWS SDK for Java

API tingkat rendah menyediakan metode untuk semua operasi arsip. Berikut adalah langkah-langkah untuk mengunggah arsip menggunakan AWS SDK for Java.

1. Buat instans dari kelas `AmazonGlacierClient` (klien).

   Anda perlu menentukan AWS Wilayah tempat Anda ingin mengunggah arsip. Semua operasi yang Anda lakukan menggunakan klien ini berlaku untuk AWS Wilayah tersebut.
2. Berikan informasi permintaan dengan membuat instans kelas `UploadArchiveRequest`.

   Selain data yang ingin Anda unggah, Anda perlu menyediakan checksum (hash pohon SHA-256) dari muatan, nama vault, panjang konten data, dan ID akun Anda.

   Jika Anda tidak memberikan ID akun, ID akun yang terkait dengan kredensial yang Anda berikan untuk menandatangani permintaan diambil. Untuk informasi selengkapnya, lihat Menggunakan AWS SDK for Java dengan Amazon Glacier.
3. Jalankan metode `uploadArchive` dengan menyediakan objek permintaan sebagai parameter.

Sebagai tanggapan, Amazon Glacier (Amazon Glacier) mengembalikan ID arsip dari arsip yang baru diunggah.

Potongan kode Java berikut menggambarkan langkah-langkah sebelumnya.

```
AmazonGlacierClient client;

UploadArchiveRequest request = new UploadArchiveRequest()
    .withVaultName("*** provide vault name ***")
    .withChecksum(checksum)
    .withBody(new ByteArrayInputStream(body))
    .withContentLength((long)body.length);

UploadArchiveResult uploadArchiveResult = client.uploadArchive(request);

System.out.println("Location (includes ArchiveID): " +
 uploadArchiveResult.getLocation());
```

Contoh: Mengunggah Arsip dalam Operasi Tunggal Menggunakan API Tingkat Rendah AWS SDK for Java

Contoh kode Java berikut menggunakan AWS SDK for Java untuk mengunggah arsip ke vault (`examplevault`). Untuk step-by-step petunjuk tentang cara menjalankan contoh ini, lihatMenjalankan Contoh Java untuk Amazon Glacier Menggunakan Eclipse. Anda perlu memperbarui kode seperti yang ditunjukkan dengan nama vault yang ingin Anda unggah dan nama file yang ingin Anda unggah.

```
import java.io.ByteArrayInputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;

import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.glacier.AmazonGlacierClient;
import com.amazonaws.services.glacier.TreeHashGenerator;
import com.amazonaws.services.glacier.model.UploadArchiveRequest;
import com.amazonaws.services.glacier.model.UploadArchiveResult;
public class ArchiveUploadLowLevel {
```

```
    public static String vaultName = "*** provide vault name ****";
    public static String archiveFilePath = "*** provide to file upload ****";
    public static AmazonGlacierClient client;

    public static void main(String[] args) throws IOException {

     ProfileCredentialsProvider credentials = new ProfileCredentialsProvider();

        client = new AmazonGlacierClient(credentials);
        client.setEndpoint("https://glacier.us-east-1.amazonaws.com/");

        try {
            // First open file and read.
            File file = new File(archiveFilePath);
            InputStream is = new FileInputStream(file);
            byte[] body = new byte[(int) file.length()];
            is.read(body);

            // Send request.
            UploadArchiveRequest request = new UploadArchiveRequest()
                .withVaultName(vaultName)
                .withChecksum(TreeHashGenerator.calculateTreeHash(new
 File(archiveFilePath)))
                .withBody(new ByteArrayInputStream(body))
                .withContentLength((long)body.length);

            UploadArchiveResult uploadArchiveResult = client.uploadArchive(request);

            System.out.println("ArchiveID: " + uploadArchiveResult.getArchiveId());

        } catch (Exception e)
        {
            System.err.println("Archive not uploaded.");
            System.err.println(e);
        }
    }
}
```

Mengunggah Arsip dalam Satu Operasi Menggunakan Amazon Glacier AWS SDK for .NET di Amazon

Baik tingkat tinggi dan tingkat rendah yang APIs disediakan oleh Amazon SDK untuk.NET menyediakan metode untuk mengunggah arsip dalam satu operasi.

Topik

- Mengunggah Arsip Menggunakan API Tingkat Tinggi AWS SDK for .NET
- Mengunggah Arsip dalam Operasi Tunggal Menggunakan API Tingkat Rendah AWS SDK for .NET

Mengunggah Arsip Menggunakan API Tingkat Tinggi AWS SDK for .NET

Kelas `ArchiveTransferManager` dari API tingkat tinggi menyediakan metode `Upload` yang dapat Anda gunakan untuk mengunggah arsip ke vault.

> **Note**
>
> Anda dapat menggunakan metode `Upload` untuk mengunggah file kecil atau besar. Bergantung pada ukuran file yang Anda unggah, metode ini menentukan apakah akan mengunggahnya dalam satu operasi atau menggunakan API unggahan multipart untuk mengunggah file dalam beberapa bagian.

Contoh: Mengunggah Arsip Menggunakan API Tingkat Tinggi AWS SDK for .NET

Contoh kode C# berikut mengunggah arsip ke vault (`examplevault`) di Wilayah US West (Oregon).

Untuk step-by-step petunjuk tentang cara menjalankan contoh ini, lihatMenjalankan Contoh Kode. Anda perlu memperbarui kode seperti yang ditunjukkan dengan nama file yang ingin Anda unggah.

Example

```
using System;
using Amazon.Glacier;
using Amazon.Glacier.Transfer;
using Amazon.Runtime;

namespace glacier.amazon.com.docsamples
{
  class ArchiveUploadHighLevel
  {
    static string vaultName = "examplevault";
    static string archiveToUpload = "*** Provide file name (with full path) to upload
 ***";

    public static void Main(string[] args)
```

```
    {
       try
      {
         var manager = new ArchiveTransferManager(Amazon.RegionEndpoint.USWest2);
         // Upload an archive.
         string archiveId = manager.Upload(vaultName, "upload archive test",
 archiveToUpload).ArchiveId;
         Console.WriteLine("Archive ID: (Copy and save this ID for use in other
 examples.) : {0}", archiveId);
         Console.WriteLine("To continue, press Enter");
         Console.ReadKey();
      }
      catch (AmazonGlacierException e) { Console.WriteLine(e.Message); }
      catch (AmazonServiceException e) { Console.WriteLine(e.Message); }
      catch (Exception e) { Console.WriteLine(e.Message); }
      Console.WriteLine("To continue, press Enter");
      Console.ReadKey();
    }
  }
}
```

Mengunggah Arsip dalam Operasi Tunggal Menggunakan API Tingkat Rendah AWS SDK for .NET

API tingkat rendah menyediakan metode untuk semua operasi arsip. Berikut adalah langkah-langkah untuk mengunggah arsip menggunakan AWS SDK for .NET.

1. Buat instans dari kelas `AmazonGlacierClient` (klien).

   Anda perlu menentukan AWS Wilayah tempat Anda ingin mengunggah arsip. Semua operasi yang Anda lakukan menggunakan klien ini berlaku untuk AWS Wilayah tersebut.
2. Berikan informasi permintaan dengan membuat instans kelas `UploadArchiveRequest`.

   Selain data yang ingin Anda unggah, Anda perlu menyediakan checksum (hash pohon SHA-256) dari muatan, nama vault, dan ID akun Anda.

   Jika Anda tidak memberikan ID akun, ID akun yang terkait dengan kredensial yang Anda berikan untuk menandatangani permintaan akan diambil. Untuk informasi selengkapnya, lihat Menggunakan AWS SDK for .NET dengan Amazon Glacier.
3. Jalankan metode `UploadArchive` dengan menyediakan objek permintaan sebagai parameter.

   Sebagai tanggapan, Amazon Glacier mengembalikan ID arsip dari arsip yang baru diunggah.

Contoh: Mengunggah Arsip dalam Operasi Tunggal Menggunakan API Tingkat Rendah AWS SDK for .NET

Contoh kode #C berikut menjelaskan langkah sebelumnya. Contoh menggunakan file AWS SDK for .NET untuk mengunggah arsip ke vault (`examplevault`).

> **Note**
>
> Untuk informasi tentang REST API yang mendasarinya guna mengunggah arsip dalam satu permintaan, lihat Mengunggah Arsip (POST archive).

Untuk step-by-step petunjuk tentang cara menjalankan contoh ini, lihatMenjalankan Contoh Kode. Anda perlu memperbarui kode seperti yang ditunjukkan dengan nama file yang ingin Anda unggah.

Example

```
using System;
using System.IO;
using Amazon.Glacier;
using Amazon.Glacier.Model;
using Amazon.Runtime;

namespace glacier.amazon.com.docsamples
{
  class ArchiveUploadSingleOpLowLevel
  {
    static string vaultName       = "examplevault";
    static string archiveToUpload = "*** Provide file name (with full path) to upload
 ***";

    public static void Main(string[] args)
    {
      AmazonGlacierClient client;
      try
      {
         using (client = new AmazonGlacierClient(Amazon.RegionEndpoint.USWest2))
        {
          Console.WriteLine("Uploading an archive.");
          string archiveId = UploadAnArchive(client);
          Console.WriteLine("Archive ID: {0}", archiveId);
        }
```

```
      }
      catch (AmazonGlacierException e) { Console.WriteLine(e.Message); }
      catch (AmazonServiceException e) { Console.WriteLine(e.Message); }
      catch (Exception e) { Console.WriteLine(e.Message); }
      Console.WriteLine("To continue, press Enter");
      Console.ReadKey();
    }

    static string UploadAnArchive(AmazonGlacierClient client)
    {
      using (FileStream fileStream = new FileStream(archiveToUpload, FileMode.Open,
 FileAccess.Read))
      {
        string treeHash = TreeHashGenerator.CalculateTreeHash(fileStream);
        UploadArchiveRequest request = new UploadArchiveRequest()
        {
          VaultName = vaultName,
          Body = fileStream,
          Checksum = treeHash
        };
        UploadArchiveResponse response = client.UploadArchive(request);
        string archiveID = response.ArchiveId;
        return archiveID;
      }
    }
  }
}
```

Mengunggah Arsip dalam Satu Operasi Menggunakan REST API

Anda dapat menggunakan panggilan API Unggah Arsip untuk mengunggah arsip dalam satu operasi. Lihat informasi yang lebih lengkap di Mengunggah Arsip (POST archive).

Mengunggah Arsip Besar dalam Beberapa Bagian (Unggahan Multipart)

Topik

- Proses Unggahan Multipart
- Fakta Cepat
- Mengunggah Arsip Besar dengan Menggunakan AWS CLI
- Mengunggah Arsip Besar dalam Beberapa Bagian Menggunakan Amazon SDK for Java
- Mengunggah Arsip Besar Menggunakan AWS SDK for .NET

- Mengunggah Arsip Besar dalam Beberapa Bagian Menggunakan REST API

Proses Unggahan Multipart

Seperti dijelaskan dalamMengunggah Arsip di Amazon Glacier, kami mendorong pelanggan Amazon Glacier (Amazon Glacier) untuk menggunakan Multipart Upload untuk mengunggah arsip lebih dari 100 mebibytes (MiB).

1. Mulai Unggahan Multibagian

   Saat Anda mengirim permintaan untuk memulai unggahan multipart, Amazon Glacier mengembalikan ID unggahan multipart, yang merupakan pengenal unik untuk unggahan multipart Anda. Operasi unggahan multipart berikutnya memerlukan ID ini. ID ini tidak kedaluwarsa setidaknya 24 jam setelah Amazon Glacier menyelesaikan pekerjaan.

   Dalam permintaan Anda untuk memulai unggahan multipart, tentukan ukuran bagian dalam jumlah byte. Setiap bagian yang Anda unggah, kecuali bagian terakhir, harus seukuran ini.

   > **Note**
   >
   > Anda tidak perlu mengetahui ukuran arsip secara keseluruhan saat menggunakan unggahan multipart. Ini berarti Anda dapat menggunakan unggahan multipart jika Anda tidak mengetahui ukuran arsip ketika Anda mulai mengunggah arsip. Anda hanya perlu memutuskan ukuran bagian pada saat Anda memulai unggahan multipart.

   Dalam permintaan mulai unggahan multipart, Anda juga dapat memberikan deskripsi arsip opsional.

2. Unggah Bagian

   Untuk setiap permintaan unggah bagian, Anda harus menyertakan ID unggahan multipart yang Anda peroleh pada langkah 1. Dalam permintaan tersebut, Anda juga harus menentukan rentang konten, dalam byte, yang mengidentifikasi posisi bagian dalam arsip final. Amazon Glacier kemudian menggunakan informasi rentang konten untuk mengumpulkan arsip dalam urutan yang tepat. Karena Anda menyediakan berbagai konten untuk setiap bagian yang Anda unggah, hal ini akan menentukan posisi bagian dalam penggabungan akhir final, dan karena itu, Anda dapat mengunggah bagian dalam urutan apa pun. Anda juga dapat mengunggah bagian secara

paralel. Jika Anda mengunggah bagian baru menggunakan rentang konten yang sama dengan bagian yang diunggah sebelumnya, bagian yang diunggah sebelumnya akan ditimpa.

3. Selesaikan (atau hentikan) Unggahan Multipart

   Setelah mengunggah semua bagian arsip, Anda menggunakan operasi selesaikan. Sekali lagi, Anda harus menentukan ID unggahan dalam permintaan Anda. Amazon Glacier membuat arsip dengan menggabungkan bagian-bagian dalam urutan menaik berdasarkan rentang konten yang Anda berikan. Respons Amazon Glacier terhadap permintaan Unggahan Multibagian Lengkap menyertakan ID arsip untuk arsip yang baru dibuat. Jika Anda memberikan deskripsi arsip opsional dalam permintaan Inisiate Multipart Upload, Amazon Glacier mengaitkannya dengan arsip rakitan. Setelah Anda berhasil menyelesaikan unggahan multipart, Anda tidak dapat merujuk ke ID unggahan multipart. Itu berarti Anda tidak dapat mengakses bagian yang terkait dengan ID unggahan multipart.

   Setelah menghentikan unggahan multipart, Anda tidak dapat mengunggah bagian apa pun menggunakan ID unggahan multipart tersebut. Semua penyimpanan yang digunakan oleh setiap bagian yang terkait dengan unggahan multipart yang dihentikan akan dikosongkan. Jika ada unggahan bagian yang sedang berlangsung, unggahan masih dapat berhasil atau gagal meskipun sudah Anda hentikan.

Operasi Unggahan Multipart Tambahan

Amazon Glacier (Amazon Glacier) menyediakan panggilan API unggahan multibagian tambahan berikut.

- Cantumkan Bagian—Dengan menggunakan operasi ini, Anda dapat mencantumkan bagian-bagian dari unggahan multipart tertentu. Operasi ini mengembalikan informasi bagian yang sudah Anda unggah untuk unggahan multipart. Untuk setiap permintaan bagian daftar, Amazon Glacier mengembalikan informasi hingga 1.000 bagian. Jika ada lebih banyak bagian yang dicantumkan untuk unggahan multipart, hasilnya adalah pemberian nomor halaman dan penanda dikembalikan dalam respons di tempat daftar dilanjutkan. Anda perlu mengirim permintaan tambahan untuk mengambil bagian berikutnya. Perhatikan bahwa daftar bagian yang telah ditampilkan tidak akan mencakup bagian yang belum selesai diunggah.
- Cantumkan Unggahan Multipart—Dengan menggunakan operasi ini, Anda bisa mendapatkan daftar unggahan multipart yang sedang diproses. Unggahan multipart yang sedang berlangsung adalah unggahan yang telah Anda mulai, tetapi belum selesai atau dihentikan. Untuk setiap permintaan unggahan multipart daftar, Amazon Glacier mengembalikan hingga 1.000 unggahan

multipart. Jika ada lebih banyak unggahan multipart yang dicantumkan, hasilnya adalah pemberian nomor halaman dan penanda dikembalikan dalam respons di tempat daftar dilanjutkan. Anda harus mengirim permintaan tambahan untuk mengambil unggahan multipart yang tersisa.

Fakta Cepat

Tabel berikut menyediakan spesifikasi inti unggahan multipart.

Item	Spesifikasi
Ukuran arsip maksimum	10.000 x 4 gibibyte (GiB)
Jumlah maksimum bagian per unggahan	10.000
Ukuran bagian	1 MiB sampai 4 GiB, bagian terakhir bisa < 1 MiB. Anda menentukan nilai ukuran dalam byte. Ukuran bagian harus mebibyte (1024 kibibytes [KiB]) dikalikan dengan kekuatan 2. Misalnya, `1048576` (1 MiB), `2097152` (2 MiB), (`41943044` MiB), (8 MiB). `8388608`
Jumlah maksimum bagian yang ditampilkan untuk permintaan daftar bagian	1.000
Jumlah maksimum unggahan multipart yang ditampilkan dalam sebuah permintaan daftar unggahan multipart	1.000

Mengunggah Arsip Besar dengan Menggunakan AWS CLI

Anda dapat mengunggah arsip di Amazon Glacier (Amazon Glacier) dengan menggunakan (). AWS Command Line Interface AWS CLI Untuk meningkatkan pengalaman mengunggah arsip yang lebih besar, Amazon Glacier menyediakan beberapa operasi API untuk mendukung unggahan multipart. Dengan menggunakan operasi API ini, Anda dapat mengunggah arsip di beberapa bagian. Bagian-bagian ini dapat diunggah secara independen, dalam urutan apa pun, dan secara paralel. Jika

unggahan bagian gagal, Anda hanya perlu mengunggah bagian itu lagi, bukan seluruh arsip. Anda dapat menggunakan unggahan multipart untuk arsip dari 1 byte hingga sekitar 40.000 gibibytes (GiB) dalam ukuran.

Untuk informasi selengkapnya tentang unggahan multipart Amazon Glacier, lihat. Mengunggah Arsip Besar dalam Beberapa Bagian (Unggahan Multipart)

Topik

- (Prasyarat) Menyiapkan AWS CLI
- (Prasyarat) Instal Python
- (Prasyarat) Buat Amazon Glacier Vault
- Contoh: Mengunggah Arsip Besar di Bagian dengan Menggunakan AWS CLI

(Prasyarat) Menyiapkan AWS CLI

1. Unduh dan konfigurasikan AWS CLI. Untuk melakukannya, lihat topik berikut di Panduan Pengguna AWS Command Line Interface :

   Instalasi AWS Command Line Interface

   Mengkonfigurasi AWS Command Line Interface
2. Verifikasi AWS CLI pengaturan Anda dengan memasukkan perintah berikut pada prompt perintah. Perintah ini tidak memberikan kredensial secara eksplisit, sehingga kredensial profil default digunakan.

   - Coba gunakan perintah bantuan.

     ```
     aws help
     ```

   - Untuk mendapatkan daftar brankas Amazon Glacier pada akun yang dikonfigurasi, gunakan perintah. `list-vaults` Ganti *123456789012* dengan Akun AWS ID Anda.

     ```
     aws glacier list-vaults --account-id 123456789012
     ```

   - Untuk melihat data konfigurasi saat ini untuk AWS CLI, gunakan `aws configure list` perintah.

     ```
     aws configure list
     ```

(Prasyarat) Instal Python

Untuk menyelesaikan unggahan multipart, Anda harus menghitung hash SHA256 pohon dari arsip yang Anda unggah. Melakukannya berbeda dengan menghitung hash SHA256 pohon dari file yang ingin Anda unggah. Untuk menghitung hash SHA256 pohon dari arsip yang Anda unggah, Anda dapat menggunakanJava, C# (dengan.NET), atau. Python Dalam contoh ini, Anda akan menggunakanPython. Untuk petunjuk tentang penggunaan Java atau C #, lihatChecksum Komputasi.

Untuk informasi selengkapnya tentang menginstalPython, lihat Menginstal atau memperbarui Python di Panduan Pengembang Boto3.

(Prasyarat) Buat Amazon Glacier Vault

Untuk menggunakan contoh berikut, Anda harus memiliki setidaknya satu lemari besi Amazon Glacier yang dibuat. Untuk informasi selengkapnya tentang membuat brankas, lihat. Membuat Vault di Amazon Glacier

Contoh: Mengunggah Arsip Besar di Bagian dengan Menggunakan AWS CLI

Dalam contoh ini, Anda akan membuat file dan menggunakan operasi API unggahan multibagian untuk mengunggah file ini, sebagian, ke Amazon Glacier.

> **Important**
>
> Sebelum memulai prosedur ini, pastikan Anda telah melakukan semua langkah prasyarat. Untuk mengunggah arsip, Anda harus memiliki vault yang dibuat, AWS CLI dikonfigurasi, dan siap untuk menggunakanJava, C #, atau Python untuk menghitung hash SHA256 pohon.

Prosedur berikut menggunakan`initiate-multipart-upload`,`upload-multipart-part`, dan `complete-multipart-upload` AWS CLI perintah.

Untuk informasi lebih rinci tentang masing-masing perintah ini, lihat, initiate-multipart-uploadupload-multipart-part, dan complete-multipart-uploaddi AWS CLI Command Reference.

1. Gunakan initiate-multipart-uploadperintah untuk membuat sumber daya unggahan multibagian. Dalam permintaan Anda, tentukan ukuran bagian dalam jumlah byte. Setiap bagian yang Anda unggah, kecuali bagian terakhir, akan berukuran ini. Anda tidak perlu mengetahui ukuran arsip keseluruhan saat memulai unggahan. Namun, Anda akan memerlukan ukuran total, dalam byte, dari setiap bagian saat menyelesaikan unggahan pada langkah terakhir.

Dalam perintah berikut, ganti nilai untuk `--account-ID` parameter `--vault-name` dan dengan informasi Anda sendiri. Perintah ini menentukan bahwa Anda akan mengunggah arsip dengan ukuran bagian 1 mebibyte (MiB) (1024 x 1024 byte) per file. Ganti nilai `--part-size` parameter ini jika diperlukan.

```
aws glacier initiate-multipart-upload --vault-name awsexamplevault --part-
size 1048576 --account-id 123456789012
```

Keluaran yang diharapkan

```
{
"location": "/123456789012/vaults/awsexamplevault/multipart-uploads/uploadId",
"uploadId": "uploadId"
}
```

Setelah selesai, perintah akan menampilkan ID unggahan dan lokasi sumber daya unggahan multibagian di Amazon Glacier. Anda akan menggunakan ID unggahan ini pada langkah selanjutnya.

2. Untuk contoh ini, Anda dapat menggunakan perintah berikut untuk membuat file 4.4 MiB, membaginya menjadi potongan 1 MiB, dan mengunggah setiap potongan. Untuk mengunggah file Anda sendiri, Anda dapat mengikuti prosedur serupa untuk membagi data Anda menjadi beberapa bagian dan mengunggah setiap bagian.

   Linux atau macOS

   Perintah berikut membuat file 4.4 MiB, bernama`file_to_upload`, di Linux atau macOS.

   ```
   mkfile -n 9000b file_to_upload
   ```

   Windows

   Perintah berikut membuat file 4.4 MiB, bernama`file_to_upload`, pada Windows.

   ```
   fsutil file createnew file_to_upload 4608000
   ```

3. Selanjutnya, Anda akan membagi file ini menjadi potongan 1 MiB.

```
split -b 1048576 file_to_upload chunk
```

Anda sekarang memiliki lima potongan berikut. Empat yang pertama adalah 1 MiB, dan yang terakhir adalah sekitar 400 kibibytes (KiB).

```
chunkaa
chunkab
chunkac
chunkad
chunkae
```

4. Gunakan upload-multipart-partperintah untuk mengunggah bagian dari arsip. Anda dapat mengunggah bagian arsip dalam urutan apa pun. Anda juga dapat mengunggahnya secara paralel. Anda dapat mengunggah hingga 10.000 bagian untuk unggahan multipart.

   Dalam perintah berikut, ganti nilai untuk`--vault-name`,`--account-ID`, dan `--upload-id` parameter. ID upload harus cocok dengan ID yang diberikan sebagai output dari `initiate-multipart-upload` perintah. `--range`Parameter menentukan bahwa Anda akan mengunggah bagian dengan ukuran 1 MiB (1024 x 1024 byte). Ukuran ini harus sesuai dengan apa yang Anda tentukan dalam `initiate-multipart-upload` perintah. Sesuaikan nilai ukuran ini jika diperlukan. `--body`Parameter menentukan nama bagian yang Anda upload.

```
aws glacier upload-multipart-part --body chunkaa --range='bytes 0-1048575/*' --
vault-name awsexamplevault --account-id 123456789012 --upload-id upload_ID
```

   Jika berhasil, perintah akan menghasilkan output yang berisi checksum untuk bagian yang diunggah.

5. Jalankan `upload-multipart-part` perintah lagi untuk mengunggah bagian yang tersisa dari unggahan multipart Anda. Perbarui nilai `--range` dan `--body` parameter untuk setiap perintah agar sesuai dengan bagian yang Anda unggah.

```
aws glacier upload-multipart-part --body chunkab --range='bytes 1048576-2097151/*'
 --vault-name awsexamplevault --account-id 123456789012 --upload-id upload_ID
```

```
aws glacier upload-multipart-part --body chunkac --range='bytes 2097152-3145727/*'
 --vault-name awsexamplevault --account-id 123456789012 --upload-id upload_ID
```

```
aws glacier upload-multipart-part --body chunkad --range='bytes 3145728-4194303/*'
 --vault-name awsexamplevault --account-id 123456789012 --upload-id upload_ID
```

```
aws glacier upload-multipart-part --body chunkae --range='bytes 4194304-4607999/*'
 --vault-name awsexamplevault --account-id 123456789012 --upload-id upload_ID
```

> **Note**
>
> Nilai `--range` parameter perintah akhir lebih kecil karena bagian akhir dari unggahan kami kurang dari 1 MiB. Jika berhasil, setiap perintah akan menghasilkan output yang berisi checksum untuk setiap bagian yang diunggah.

6. Selanjutnya, Anda akan merakit arsip dan menyelesaikan unggahan. Anda harus menyertakan ukuran total dan hash SHA256 pohon arsip.

   Untuk menghitung hash SHA256 pohon arsip, Anda dapat menggunakanJava, C #, atau. Python Dalam contoh ini, Anda akan menggunakanPython. Untuk petunjuk tentang penggunaan Java atau C #, lihatChecksum Komputasi.

   Buat Python file `checksum.py` dan masukkan kode berikut. Jika perlu, ganti nama file asli.

   ```
   from botocore.utils import calculate_tree_hash

   checksum = calculate_tree_hash(open('file_to_upload', 'rb'))
   print(checksum)
   ```

7. Jalankan `checksum.py` untuk menghitung hash SHA256 pohon. Hash berikut mungkin tidak cocok dengan output Anda.

   ```
   $ python3 checksum.py
   $ 3d760edb291bfc9d90d35809243de092aea4c47b308290ad12d084f69988ae0c
   ```

8. Gunakan complete-multipart-uploadperintah untuk menyelesaikan unggahan arsip. Ganti nilai untuk`--vault-name`,`--account-ID`,`--upload-ID`, dan `--checksum` parameter. Nilai `--archive` parameter menentukan ukuran total, dalam byte, dari arsip. Nilai ini harus merupakan jumlah dari semua ukuran masing-masing bagian yang Anda unggah. Ganti nilai ini jika diperlukan.

```
aws glacier complete-multipart-upload --archive-size 4608000 --vault-
name awsexamplevault --account-id 123456789012 --upload-id upload_ID --
checksum checksum
```

Setelah selesai, perintah akan menampilkan ID arsip, checksum, dan lokasi di Amazon Glacier.

Mengunggah Arsip Besar dalam Beberapa Bagian Menggunakan Amazon SDK for Java

Baik tingkat tinggi maupun tingkat rendah yang APIs disediakan oleh Amazon SDK for Java menyediakan metode untuk mengunggah arsip besar (lihat). Mengunggah Arsip di Amazon Glacier

- API tingkat tinggi menyediakan metode yang dapat Anda gunakan untuk mengunggah arsip dalam ukuran apa pun. Bergantung pada file yang Anda unggah, metode ini mengunggah arsip dalam satu operasi atau menggunakan dukungan unggahan multibagian di Amazon Glacier (Amazon Glacier) untuk mengunggah arsip menjadi beberapa bagian.
- API tingkat rendah hampir memetakan ke implementasi REST yang mendasarinya. Dengan demikian, ini menyediakan metode untuk mengunggah arsip yang lebih kecil dalam satu operasi dan grup metode yang mendukung unggahan multipart untuk arsip yang lebih besar. Bagian ini menjelaskan mengunggah arsip besar dalam beberapa bagian menggunakan API tingkat rendah.

Untuk informasi lebih lanjut tentang tingkat tinggi dan tingkat rendah APIs, lihat. Menggunakan AWS SDK for Java dengan Amazon Glacier

Topik

- Mengunggah Arsip Besar di Bagian Menggunakan API Tingkat Tinggi AWS SDK for Java
- Unggah Arsip Besar di Bagian Menggunakan API Tingkat Rendah AWS SDK for Java

Mengunggah Arsip Besar di Bagian Menggunakan API Tingkat Tinggi AWS SDK for Java

Anda menggunakan metode API tingkat tinggi yang sama untuk mengunggah arsip kecil atau besar. Berdasarkan ukuran arsip, metode API tingkat tinggi memutuskan apakah akan mengunggah arsip dalam satu operasi atau menggunakan API unggahan multibagian yang disediakan oleh Amazon Glacier. Untuk informasi selengkapnya, lihat Mengunggah Arsip Menggunakan API Tingkat Tinggi AWS SDK for Java.

Unggah Arsip Besar di Bagian Menggunakan API Tingkat Rendah AWS SDK for Java

Untuk kontrol unggahan granular, Anda dapat menggunakan API tingkat rendah tempat Anda dapat mengonfigurasi permintaan dan memproses respons. Berikut adalah langkah-langkah untuk mengunggah arsip besar dalam beberapa bagian menggunakan AWS SDK for Java.

1. Buat instans dari kelas `AmazonGlacierClient` (klien).

   Anda perlu menentukan AWS Wilayah tempat Anda ingin menyimpan arsip. Semua operasi yang Anda lakukan menggunakan klien ini berlaku untuk AWS Wilayah tersebut.
2. Mulai unggahan multipart dengan memanggil metode `initiateMultipartUpload`.

   Anda perlu memberi nama vault tempat Anda ingin mengunggah arsip, ukuran bagian yang ingin Anda gunakan untuk mengunggah bagian arsip, dan deskripsi opsional. Anda memberikan informasi ini dengan membuat instans kelas `InitiateMultipartUploadRequest`. Sebagai tanggapan, Amazon Glacier mengembalikan ID unggahan.
3. Unggah bagian dengan memanggil metode `uploadMultipartPart`.

   Untuk setiap bagian yang Anda unggah, Anda perlu memberi nama vault, rentang byte dalam gabungan arsip final yang akan diunggah di bagian ini, checksum dari data bagian, dan ID unggahan.
4. Selesaikan unggahan multipart dengan memanggil metode `completeMultipartUpload`.

   Anda perlu memberikan ID unggahan, checksum dari seluruh arsip, ukuran arsip (ukuran gabungan dari semua bagian yang Anda unggah), dan nama vault. Amazon Glacier membuat arsip dari bagian yang diunggah dan mengembalikan ID arsip.

Contoh: Mengunggah Arsip Besar di Bagian Menggunakan AWS SDK for Java

Contoh kode Java berikut menggunakan AWS SDK for Java untuk meng-upload arsip ke vault (`examplevault`). Untuk step-by-step petunjuk tentang cara menjalankan contoh ini, lihatMenjalankan Contoh Java untuk Amazon Glacier Menggunakan Eclipse. Anda perlu memperbarui kode seperti yang ditunjukkan dengan nama file yang ingin Anda unggah.

> **Note**
>
> Contoh ini berlaku untuk ukuran bagian mulai dari 1 MB hingga 1 GB. Namun, Amazon Glacier mendukung ukuran bagian hingga 4 GB.

Example

```
import java.io.ByteArrayInputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.security.NoSuchAlgorithmException;
import java.util.Arrays;
import java.util.Date;
import java.util.LinkedList;
import java.util.List;

import com.amazonaws.AmazonClientException;
import com.amazonaws.AmazonServiceException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.glacier.AmazonGlacierClient;
import com.amazonaws.services.glacier.TreeHashGenerator;
import com.amazonaws.services.glacier.model.CompleteMultipartUploadRequest;
import com.amazonaws.services.glacier.model.CompleteMultipartUploadResult;
import com.amazonaws.services.glacier.model.InitiateMultipartUploadRequest;
import com.amazonaws.services.glacier.model.InitiateMultipartUploadResult;
import com.amazonaws.services.glacier.model.UploadMultipartPartRequest;
import com.amazonaws.services.glacier.model.UploadMultipartPartResult;
import com.amazonaws.util.BinaryUtils;

public class ArchiveMPU {

    public static String vaultName = "examplevault";
    // This example works for part sizes up to 1 GB.
    public static String partSize = "1048576"; // 1 MB.
    public static String archiveFilePath = "*** provide archive file path ***";
    public static AmazonGlacierClient client;

    public static void main(String[] args) throws IOException {

     ProfileCredentialsProvider credentials = new ProfileCredentialsProvider();
```

```
        client = new AmazonGlacierClient(credentials);
        client.setEndpoint("https://glacier.us-west-2.amazonaws.com/");

        try {
            System.out.println("Uploading an archive.");
            String uploadId = initiateMultipartUpload();
            String checksum = uploadParts(uploadId);
            String archiveId = CompleteMultiPartUpload(uploadId, checksum);
            System.out.println("Completed an archive. ArchiveId: " + archiveId);

        } catch (Exception e) {
            System.err.println(e);
        }

    }

    private static String initiateMultipartUpload() {
        // Initiate
        InitiateMultipartUploadRequest request = new InitiateMultipartUploadRequest()
            .withVaultName(vaultName)
            .withArchiveDescription("my archive " + (new Date()))
            .withPartSize(partSize);

        InitiateMultipartUploadResult result = client.initiateMultipartUpload(request);

        System.out.println("ArchiveID: " + result.getUploadId());
        return result.getUploadId();
    }

    private static String uploadParts(String uploadId) throws AmazonServiceException,
NoSuchAlgorithmException, AmazonClientException, IOException {

        int filePosition = 0;
        long currentPosition = 0;
        byte[] buffer = new byte[Integer.valueOf(partSize)];
        List<byte[]> binaryChecksums = new LinkedList<byte[]>();

        File file = new File(archiveFilePath);
        FileInputStream fileToUpload = new FileInputStream(file);
        String contentRange;
        int read = 0;
        while (currentPosition < file.length())
        {
            read = fileToUpload.read(buffer, filePosition, buffer.length);
```

```
            if (read == -1) { break; }
            byte[] bytesRead = Arrays.copyOf(buffer, read);

            contentRange = String.format("bytes %s-%s/*", currentPosition,
currentPosition + read - 1);
            String checksum = TreeHashGenerator.calculateTreeHash(new
ByteArrayInputStream(bytesRead));
            byte[] binaryChecksum = BinaryUtils.fromHex(checksum);
            binaryChecksums.add(binaryChecksum);
            System.out.println(contentRange);

            //Upload part.
            UploadMultipartPartRequest partRequest = new UploadMultipartPartRequest()
            .withVaultName(vaultName)
            .withBody(new ByteArrayInputStream(bytesRead))
            .withChecksum(checksum)
            .withRange(contentRange)
            .withUploadId(uploadId);

            UploadMultipartPartResult partResult =
client.uploadMultipartPart(partRequest);
            System.out.println("Part uploaded, checksum: " + partResult.getChecksum());

            currentPosition = currentPosition + read;
        }
        fileToUpload.close();
        String checksum = TreeHashGenerator.calculateTreeHash(binaryChecksums);
        return checksum;
    }

    private static String CompleteMultiPartUpload(String uploadId, String checksum)
throws NoSuchAlgorithmException, IOException {

        File file = new File(archiveFilePath);

        CompleteMultipartUploadRequest compRequest = new
CompleteMultipartUploadRequest()
            .withVaultName(vaultName)
            .withUploadId(uploadId)
            .withChecksum(checksum)
            .withArchiveSize(String.valueOf(file.length()));

        CompleteMultipartUploadResult compResult =
client.completeMultipartUpload(compRequest);
```

```
        return compResult.getLocation();
    }
}
```

Mengunggah Arsip Besar Menggunakan AWS SDK for .NET

Baik tingkat tinggi maupun tingkat rendah yang APIs disediakan oleh Amazon SDK for .NET menyediakan metode untuk mengunggah arsip besar di beberapa bagian (lihat). Mengunggah Arsip di Amazon Glacier

- API tingkat tinggi menyediakan metode yang dapat Anda gunakan untuk mengunggah arsip dalam ukuran apa pun. Bergantung pada file yang Anda unggah, metode ini mengunggah arsip dalam satu operasi atau menggunakan dukungan unggahan multibagian di Amazon Glacier (Amazon Glacier) untuk mengunggah arsip menjadi beberapa bagian.
- API tingkat rendah hampir memetakan ke implementasi REST yang mendasarinya. Dengan demikian, ini menyediakan metode untuk mengunggah arsip yang lebih kecil dalam satu operasi dan grup metode yang mendukung unggahan multipart untuk arsip yang lebih besar. Bagian ini menjelaskan mengunggah arsip besar dalam beberapa bagian menggunakan API tingkat rendah.

Untuk informasi lebih lanjut tentang tingkat tinggi dan tingkat rendah APIs, lihat. Menggunakan AWS SDK for .NET dengan Amazon Glacier

Topik

- Mengunggah Arsip Besar di Bagian Menggunakan API Tingkat Tinggi AWS SDK for .NET
- Mengunggah Arsip Besar di Bagian Menggunakan API Tingkat Rendah AWS SDK for .NET

Mengunggah Arsip Besar di Bagian Menggunakan API Tingkat Tinggi AWS SDK for .NET

Anda menggunakan metode API tingkat tinggi yang sama untuk mengunggah arsip kecil atau besar. Berdasarkan ukuran arsip, metode API tingkat tinggi memutuskan apakah akan mengunggah arsip dalam satu operasi atau menggunakan API unggahan multibagian yang disediakan oleh Amazon Glacier. Untuk informasi selengkapnya, lihat Mengunggah Arsip Menggunakan API Tingkat Tinggi AWS SDK for .NET.

Mengunggah Arsip Besar di Bagian Menggunakan API Tingkat Rendah AWS SDK for .NET

Untuk kontrol unggahan granular, Anda dapat menggunakan API tingkat rendah, tempat Anda dapat mengonfigurasi permintaan dan memproses respons. Berikut adalah langkah-langkah untuk mengunggah arsip besar dalam beberapa bagian menggunakan AWS SDK for .NET.

1. Buat instans dari kelas `AmazonGlacierClient` (klien).

   Anda perlu menentukan AWS Wilayah tempat Anda ingin menyimpan arsip. Semua operasi yang Anda lakukan menggunakan klien ini berlaku untuk AWS Wilayah tersebut.
2. Mulai unggahan multipart dengan memanggil metode `InitiateMultipartUpload`.

   Anda perlu memberi nama vault tempat Anda ingin mengunggah arsip, ukuran bagian yang ingin Anda gunakan untuk mengunggah bagian arsip, dan deskripsi opsional. Anda memberikan informasi ini dengan membuat instans kelas `InitiateMultipartUploadRequest`. Sebagai tanggapan, Amazon Glacier mengembalikan ID unggahan.
3. Unggah bagian dengan memanggil metode `UploadMultipartPart`.

   Untuk setiap bagian yang Anda unggah, Anda perlu memberi nama vault, rentang byte dalam gabungan arsip final yang akan diunggah di bagian ini, checksum dari data bagian, dan ID unggahan.
4. Selesaikan unggahan multipart dengan memanggil metode `CompleteMultipartUpload`.

   Anda perlu memberikan ID unggahan, checksum dari seluruh arsip, ukuran arsip (ukuran gabungan dari semua bagian yang Anda unggah), dan nama vault. Amazon Glacier membuat arsip dari bagian yang diunggah dan mengembalikan ID arsip.

Contoh: Mengunggah Arsip Besar dalam Beberapa Bagian Menggunakan Amazon SDK for .NET

Contoh kode C # berikut menggunakan AWS SDK for .NET untuk mengunggah arsip ke vault ()`examplevault`. Untuk step-by-step petunjuk tentang cara menjalankan contoh ini, lihatMenjalankan Contoh Kode. Anda perlu memperbarui kode seperti yang ditunjukkan dengan nama file yang ingin Anda unggah.

Example

```
using System;
using System.Collections.Generic;
using System.IO;
```

```
using Amazon.Glacier;
using Amazon.Glacier.Model;
using Amazon.Runtime;

namespace glacier.amazon.com.docsamples
{
  class ArchiveUploadMPU
  {
    static string vaultName       = "examplevault";
    static string archiveToUpload = "*** Provide file name (with full path) to upload
 ***";
    static long partSize          = 4194304; // 4 MB.

    public static void Main(string[] args)
    {
      AmazonGlacierClient client;
      List<string> partChecksumList = new List<string>();
      try
      {
         using (client = new AmazonGlacierClient(Amazon.RegionEndpoint.USWest2))
        {
          Console.WriteLine("Uploading an archive.");
          string uploadId  = InitiateMultipartUpload(client);
          partChecksumList = UploadParts(uploadId, client);
          string archiveId = CompleteMPU(uploadId, client, partChecksumList);
          Console.WriteLine("Archive ID: {0}", archiveId);
        }
        Console.WriteLine("Operations successful. To continue, press Enter");
        Console.ReadKey();
      }
      catch (AmazonGlacierException e) { Console.WriteLine(e.Message); }
      catch (AmazonServiceException e) { Console.WriteLine(e.Message); }
      catch (Exception e) { Console.WriteLine(e.Message); }
      Console.WriteLine("To continue, press Enter");
      Console.ReadKey();
    }

    static string InitiateMultipartUpload(AmazonGlacierClient client)
    {
      InitiateMultipartUploadRequest initiateMPUrequest = new
 InitiateMultipartUploadRequest()
      {

        VaultName = vaultName,
```

```
        PartSize = partSize,
        ArchiveDescription = "Test doc uploaded using MPU."
      };

      InitiateMultipartUploadResponse initiateMPUresponse =
client.InitiateMultipartUpload(initiateMPUrequest);

      return initiateMPUresponse.UploadId;
    }

    static List<string> UploadParts(string uploadID, AmazonGlacierClient client)
    {
      List<string> partChecksumList = new List<string>();
      long currentPosition = 0;
      var buffer = new byte[Convert.ToInt32(partSize)];

      long fileLength = new FileInfo(archiveToUpload).Length;
      using (FileStream fileToUpload = new FileStream(archiveToUpload, FileMode.Open,
FileAccess.Read))
      {
        while (fileToUpload.Position < fileLength)
        {
          Stream uploadPartStream = GlacierUtils.CreatePartStream(fileToUpload,
partSize);
          string checksum = TreeHashGenerator.CalculateTreeHash(uploadPartStream);
          partChecksumList.Add(checksum);
          // Upload part.
          UploadMultipartPartRequest uploadMPUrequest = new
UploadMultipartPartRequest()
          {

            VaultName = vaultName,
            Body = uploadPartStream,
            Checksum = checksum,
            UploadId = uploadID
          };
          uploadMPUrequest.SetRange(currentPosition, currentPosition +
uploadPartStream.Length - 1);
          client.UploadMultipartPart(uploadMPUrequest);

          currentPosition = currentPosition + uploadPartStream.Length;
        }
      }
      return partChecksumList;
```

```
    }

    static string CompleteMPU(string uploadID, AmazonGlacierClient client, List<string>
 partChecksumList)
    {
      long fileLength = new FileInfo(archiveToUpload).Length;
      CompleteMultipartUploadRequest completeMPUrequest = new
 CompleteMultipartUploadRequest()
      {
        UploadId = uploadID,
        ArchiveSize = fileLength.ToString(),
        Checksum = TreeHashGenerator.CalculateTreeHash(partChecksumList),
        VaultName = vaultName
      };

      CompleteMultipartUploadResponse completeMPUresponse =
 client.CompleteMultipartUpload(completeMPUrequest);
      return completeMPUresponse.ArchiveId;
    }
  }
}
```

Mengunggah Arsip Besar dalam Beberapa Bagian Menggunakan REST API

Seperti yang dijelaskan dalam Mengunggah Arsip Besar dalam Beberapa Bagian (Unggahan Multipart), unggahan multipart mengacu pada serangkaian operasi yang memungkinkan Anda mengunggah arsip dalam beberapa bagian dan melakukan operasi terkait. Untuk informasi selengkapnya tentang operasi ini, lihat topik referensi API berikut:

- Memulai Unggahan Multipart (POST multipart-uploads)
- Mengunggah Bagian (PUT UploadID)
- Menyelesaikan Unggahan Multipart (POST uploadID)
- Membatalkan Unggahan Multipart (DELETE uploadID)
- Mencantumkan Bagian (GET uploadID)
- Mencantumkan Unggahan Multipart (GET multipart-uploads)

Mengunduh Arsip di Amazon Glacier

Amazon Glacier menyediakan konsol manajemen, yang dapat Anda gunakan untuk membuat dan menghapus brankas. Namun, Anda tidak dapat mengunduh arsip dari Amazon Glacier dengan menggunakan konsol manajemen. Untuk mengunduh data, seperti foto, video, dan dokumen lainnya, Anda harus menggunakan AWS Command Line Interface (AWS CLI) atau menulis kode untuk membuat permintaan, dengan menggunakan REST API secara langsung atau dengan menggunakan AWS SDKs.

Untuk informasi tentang penggunaan Amazon Glacier dengan, lihat AWS CLI Referensi AWS CLI untuk Amazon Glacier. Untuk menginstal AWS CLI, lihat AWS Command Line Interface. Topik berikut menjelaskan cara mengunduh arsip ke Amazon Glacier dengan menggunakan AWS SDK for Java, API REST Amazon Glacier, AWS SDK for .NET dan Amazon Glacier.

Topik

- Mengambil Arsip Amazon Glacier
- Mengunduh Arsip di Amazon Glacier Menggunakan AWS SDK for Java
- Mengunduh Arsip di Amazon Glacier Menggunakan AWS SDK for .NET
- Mengunduh Arsip Besar Menggunakan Pemrosesan Paralel dengan Python
- Mengunduh Arsip dengan Menggunakan REST API
- Mengunduh Arsip di Amazon Glacier Menggunakan AWS CLI

Mengambil Arsip Amazon Glacier

Mengambil arsip dari Amazon Glacier adalah operasi asinkron di mana Anda pertama kali memulai pekerjaan, dan kemudian mengunduh output setelah pekerjaan selesai. Untuk memulai pekerjaan pengambilan arsip, Anda menggunakan operasi Memulai Tugas (POST jobs) REST API atau yang setara di AWS CLI, atau. AWS SDKs

Topik

- Opsi Pengambilan Arsip
- Pengambilan Arsip Berkisar

Mengambil arsip dari Amazon Glacier adalah proses dua langkah. Berikut ini adalah ikhtisar dari proses ini.

Untuk mengambil arsip

1. Mulai tugas pengambilan arsip.

   a. Dapatkan ID arsip yang ingin Anda ambil. Anda bisa mendapatkan ID arsip dari inventaris vault. Anda bisa mendapatkan ID arsip dengan REST API, AWS CLI, atau AWS SDKs. Untuk informasi selengkapnya, lihat Mengunduh Inventaris Vault di Amazon Glacier.

   b. Memulai pekerjaan yang meminta Amazon Glacier untuk menyiapkan seluruh arsip atau sebagian arsip untuk diunduh berikutnya dengan menggunakan operasi. Memulai Tugas (POST jobs)

   Saat Anda memulai pekerjaan, Amazon Glacier mengembalikan ID pekerjaan dalam respons dan menjalankan pekerjaan secara asinkron. (Anda tidak dapat mengunduh hasil pekerjaan sampai pekerjaan selesai, seperti yang dijelaskan pada Langkah 2.)

   > **Important**
   >
   > Hanya untuk pengambilan Standar, kebijakan pengambilan data dapat menyebabkan `Initiate Job` permintaan Anda gagal dengan pengecualian. `PolicyEnforcedException` Untuk informasi selengkapnya tentang kebijakan pengambilan data, lihat Kebijakan Pengambilan Data Amazon Glacier. Untuk informasi selengkapnya tentang pengecualian `PolicyEnforcedException`, lihat Respons Kesalahan.

   Bila diperlukan, Anda dapat memulihkan segmen besar data yang disimpan di Amazon Glacier. Untuk informasi selengkapnya tentang memulihkan data dari kelas penyimpanan Amazon Glacier, lihat Kelas Penyimpanan untuk Mengarsipkan Objek di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

2. Setelah pekerjaan selesai, unduh byte dengan menggunakan Mendapatkan Output Tugas (GET output) operasi.

   Anda dapat mengunduh semua byte atau menentukan rentang byte untuk hanya mengunduh sebagian output tugas. Untuk output yang lebih besar, mengunduh output dalam potongan membantu dalam hal kegagalan unduhan, seperti kegagalan jaringan. Jika Anda mendapatkan output tugas dalam satu permintaan dan terjadi kegagalan jaringan, Anda harus memulai ulang pengunduhan output dari awal. Namun, jika Anda mengunduh output dalam potongan, jika

terjadi kegagalan apa pun, Anda hanya perlu memulai ulang unduhan bagian yang lebih kecil dan bukan keseluruhan output.

Amazon Glacier harus menyelesaikan pekerjaan sebelum Anda bisa mendapatkan hasilnya. Setelah selesai, pekerjaan tidak kedaluwarsa setidaknya 24 jam, yang berarti Anda dapat mengunduh output dalam periode 24 jam setelah pekerjaan selesai. Pemulihan dapat kedaluwarsa kapan saja setelah 24 jam sejak selesainya pekerjaan. Untuk menentukan apakah tugas Anda selesai, periksa statusnya menggunakan salah satu opsi berikut:

- Tunggu pemberitahuan penyelesaian pekerjaan - Anda dapat menentukan topik Amazon Simple Notification Service (Amazon SNS) tempat Amazon Glacier dapat memposting pemberitahuan setelah pekerjaan selesai. Amazon Glacier mengirimkan pemberitahuan hanya setelah menyelesaikan pekerjaan.

  Anda dapat menentukan topik Amazon SNS untuk tugas ketika Anda memulai tugas. Selain menentukan topik Amazon SNS dalam permintaan pekerjaan Anda, jika vault Anda memiliki pemberitahuan yang disetel untuk peristiwa pengambilan arsip, Amazon Glacier juga menerbitkan pemberitahuan ke topik SNS tersebut. Untuk informasi selengkapnya, lihat Mengkonfigurasi Pemberitahuan Vault di Amazon Glacier.
- Meminta informasi pekerjaan secara eksplisit — Anda juga dapat menggunakan operasi Amazon `Describe Job` Glacier API Mendeskripsikan Tugas (GET JobID) () untuk melakukan polling secara berkala untuk informasi pekerjaan. Namun, sebaiknya gunakan notifikasi Amazon SNS.

> **Note**
>
> Informasi yang Anda dapatkan dengan menggunakan notifikasi Amazon SNS sama dengan apa yang Anda dapatkan dengan memanggil operasi `Describe Job` API.

Opsi Pengambilan Arsip

Saat memulai pekerjaan untuk mengambil arsip, Anda dapat menentukan salah satu opsi pengambilan berikut, berdasarkan waktu akses dan persyaratan biaya Anda. Untuk informasi tentang harga pengambilan, lihat Harga Amazon Glacier.

- Dipercepat - Pengambilan yang dipercepat memungkinkan Anda mengakses data dengan cepat yang disimpan di kelas penyimpanan S3 Glacier Flexible Retrieval atau tingkat Akses Arsip Tingkat

Cerdas S3 ketika permintaan mendesak sesekali untuk memulihkan arsip diperlukan. Untuk semua kecuali arsip terbesar (lebih dari 250 MB), data yang diakses dengan menggunakan Expedited retrievals biasanya tersedia dalam 1-5 menit. Kapasitas yang disediakan memastikan bahwa kapasitas pengambilan untuk pengambilan yang dipercepat tersedia saat Anda membutuhkannya. Untuk informasi selengkapnya, lihat Kapasitas yang Disediakan.

- Standar — Pengambilan standar memungkinkan Anda mengakses arsip Anda dalam beberapa jam. Pengambilan standar biasanya diselesaikan dalam waktu 3-5 jam. Standar adalah opsi default untuk permintaan pengambilan yang tidak menentukan opsi pengambilan.
- Massal — Pengambilan massal adalah opsi pengambilan Amazon Glacier dengan biaya terendah, yang dapat Anda gunakan untuk mengambil data dalam jumlah besar, bahkan petabyte, dengan harga murah dalam sehari. Pengambilan massal biasanya diselesaikan dalam waktu 5-12 jam.

Tabel berikut merangkum opsi pengambilan arsip. Untuk informasi lebih lanjut tenngenai harga, lihat harga Amazon Glacier.

Untuk membuat`Expedited`,`Standard`, atau `Bulk` pengambilan, setel elemen `Tier` permintaan dalam permintaan operasi RestoreObjectREST API ke opsi yang Anda inginkan, atau yang setara dalam 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 yang Dipercepat 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 memerlukan akses yang sangat andal dan dapat diprediksi ke subset data Anda dalam hitungan menit, sebaiknya Anda membeli kapasitas pengambilan yang disediakan. Tanpa kapasitas yang disediakan, pengambilan yang dipercepat biasanya diterima, kecuali untuk situasi langka dengan permintaan yang luar biasa tinggi. Namun, jika Anda memerlukan akses ke Pengambilan yang dipercepat dalam segala situasi, Anda harus membeli kapasitas pengambilan yang disediakan.

Membeli Kapasitas yang Disediakan

Anda dapat membeli unit kapasitas yang disediakan dengan menggunakan konsol Amazon Glacier, operasi REST API, Membeli Kapasitas yang Disediakan (GET provisioned-capacity) atau. AWS SDKs AWS CLI Untuk informasi harga kapasitas yang disediakan, lihat Harga Amazon Glacier.

Unit kapasitas yang disediakan berlangsung selama satu bulan, mulai dari tanggal dan waktu pembelian.

Jika tanggal mulai adalah tanggal 31, tanggal kedaluwarsa adalah hari terakhir bulan berikutnya. Misalnya, jika tanggal mulai adalah 31 Agustus, tanggal kedaluwarsa adalah 30 September. Jika tanggal mulai 31 Januari, tanggal kedaluwarsa adalah 28 Februari.

Untuk membeli kapasitas yang disediakan dengan menggunakan konsol Amazon Glacier

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Glacier di rumah. https://console.aws.amazon.com/glacier/
2. Di panel navigasi kiri, pilih Pengaturan pengambilan data.
3. Di bawah Unit kapasitas yang disediakan (PCUs), pilih Beli PCU. Kotak dialog Pembelian PCU muncul.
4. Jika Anda ingin membeli kapasitas yang disediakan, masukkan kotak **`confirm`** Untuk mengonfirmasi pembelian.
5. Pilih Beli PCU.

Pengambilan Arsip Berkisar

Ketika Anda mengambil arsip dari Amazon Glacier, Anda dapat secara opsional menentukan rentang, atau bagian, dari arsip yang diambil. Default-nya adalah mengambil seluruh arsip. Menentukan rentang byte dapat membantu ketika Anda ingin melakukan hal berikut:

- Kelola unduhan data Anda — Amazon Glacier memungkinkan data yang diambil untuk diunduh selama 24 jam setelah permintaan pengambilan selesai. Oleh karena itu, Anda mungkin hanya ingin mengambil bagian dari arsip sehingga Anda dapat mengelola jadwal unduhan dalam jendela unduhan yang diberikan.
- Mengambil bagian yang ditargetkan dari arsip besar – Sebagai contoh, misalkan Anda sebelumnya sudah mengumpulkan banyak file dan mengunggahnya sebagai arsip tunggal, dan sekarang Anda ingin mengambil beberapa file. Dalam hal ini, Anda dapat menentukan rentang arsip yang berisi file

yang Anda minati dengan menggunakan satu permintaan pengambilan. Atau, Anda dapat memulai beberapa permintaan pengambilan, masing-masing dengan rentang untuk satu atau beberapa file.

Ketika memulai tugas pengambilan menggunakan pengambilan rentang, Anda harus memberikan rentang yang selaras dengan megabyte. Dengan kata lain, rentang byte dapat dimulai dari nol (awal arsip Anda), atau pada interval 1-MB sesudahnya (1 MB, 2 MB, 3 MB, dan seterusnya).

Akhir rentang dapat menjadi akhir arsip Anda atau interval 1 MB lebih besar dari awal rentang Anda. Selanjutnya, jika Anda ingin mendapatkan nilai checksum saat mengunduh data (setelah pekerjaan pengambilan selesai), rentang yang Anda minta dalam inisiasi pekerjaan juga harus disejajarkan dengan hash pohon. Anda dapat menggunakan checksum untuk membantu memastikan bahwa data Anda tidak rusak selama transmisi. Untuk informasi selengkapnya tentang penyelarasan megabyte dan penyelarasan hash pohon, lihatMenerima Checksum Saat Mengunduh Data.

Mengunduh Arsip di Amazon Glacier Menggunakan AWS SDK for Java

Baik tingkat tinggi dan tingkat rendah yang APIs disediakan oleh Amazon SDK for Java menyediakan metode untuk mengunduh arsip.

Topik

- Mengunduh Arsip Menggunakan API Tingkat Tinggi AWS SDK for Java
- Mengunduh Arsip Menggunakan API Tingkat Rendah AWS SDK for Java

Mengunduh Arsip Menggunakan API Tingkat Tinggi AWS SDK for Java

Kelas `ArchiveTransferManager` dari API tingkat tinggi menyediakan metode `download` yang dapat Anda gunakan untuk mengunduh arsip.

> **Important**
>
> Kelas `ArchiveTransferManager` membuat topik Amazon Simple Notification Service (Amazon SNS) dan antrean Amazon Simple Queue Service (Amazon SQS) yang berlangganan topik tersebut. Kelas tersebut selanjutnya memulai tugas pengambilan arsip dan melakukan polling antrean untuk arsip yang akan tersedia. Ketika arsip tersedia, unduhan dimulai. Untuk informasi tentang waktu pengambilan, lihat Opsi Pengambilan Arsip.

Contoh: Mengunduh Arsip Menggunakan API Tingkat Tinggi AWS SDK for Java

Contoh kode Java berikut mengunduh arsip dari vault (`examplevault`) di Wilayah US West (Oregon) (`us-west-2`).

Untuk step-by-step instruksi untuk menjalankan sampel ini, lihatMenjalankan Contoh Java untuk Amazon Glacier Menggunakan Eclipse. Anda perlu memperbarui kode seperti yang ditunjukkan dengan ID arsip yang ada dan jalur file lokal tempat Anda ingin menyimpan arsip yang diunduh.

Example

```
import java.io.File;
import java.io.IOException;

import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.glacier.AmazonGlacierClient;
import com.amazonaws.services.glacier.transfer.ArchiveTransferManager;
import com.amazonaws.services.sns.AmazonSNSClient;
import com.amazonaws.services.sqs.AmazonSQSClient;


public class ArchiveDownloadHighLevel {
    public static String vaultName = "examplevault";
    public static String archiveId = "*** provide archive ID ***";
    public static String downloadFilePath  = "*** provide location to download archive
 ***";

    public static AmazonGlacierClient glacierClient;
    public static AmazonSQSClient sqsClient;
    public static AmazonSNSClient snsClient;

    public static void main(String[] args) throws IOException {

     ProfileCredentialsProvider credentials = new ProfileCredentialsProvider();

        glacierClient = new AmazonGlacierClient(credentials);

        sqsClient = new AmazonSQSClient(credentials);
        snsClient = new AmazonSNSClient(credentials);
        glacierClient.setEndpoint("glacier.us-west-2.amazonaws.com");
        sqsClient.setEndpoint("sqs.us-west-2.amazonaws.com");
        snsClient.setEndpoint("sns.us-west-2.amazonaws.com");
```

```
        try {
            ArchiveTransferManager atm = new ArchiveTransferManager(glacierClient,
 sqsClient, snsClient);

            atm.download(vaultName, archiveId, new File(downloadFilePath));
            System.out.println("Downloaded file to " + downloadFilePath);

        } catch (Exception e)
        {
            System.err.println(e);
        }
    }
}
```

Mengunduh Arsip Menggunakan API Tingkat Rendah AWS SDK for Java

Berikut adalah langkah-langkah untuk mengambil inventaris vault menggunakan API tingkat rendah AWS SDK for Java .

1. Buat instans dari kelas `AmazonGlacierClient` (klien).

   Anda perlu menentukan AWS Wilayah dari mana Anda ingin mengunduh arsip. Semua operasi yang Anda lakukan menggunakan klien ini berlaku untuk AWS Wilayah tersebut.
2. Mulai tugas `archive-retrieval` dengan menjalankan metode `initiateJob`.

   Anda memberikan informasi pekerjaan, seperti ID arsip arsip yang ingin Anda unduh dan topik Amazon SNS opsional yang Anda inginkan Amazon Glacier (Amazon Glacier) untuk memposting pesan penyelesaian pekerjaan, dengan membuat instance kelas. `InitiateJobRequest` Amazon Glacier mengembalikan ID pekerjaan sebagai tanggapan. Respons tersedia dalam instans dari kelas `InitiateJobResult`.

```
JobParameters jobParameters = new JobParameters()
    .withArchiveId("*** provide an archive id ***")
    .withDescription("archive retrieval")
    .withRetrievalByteRange("*** provide a retrieval range***") // optional
    .withType("archive-retrieval");

InitiateJobResult initiateJobResult = client.initiateJob(new InitiateJobRequest()
    .withJobParameters(jobParameters)
    .withVaultName(vaultName));
```

```
String jobId = initiateJobResult.getJobId();
```

Anda dapat secara opsional menentukan rentang byte untuk meminta Amazon Glacier untuk menyiapkan hanya sebagian dari arsip. Misalnya, Anda dapat memperbarui permintaan sebelumnya dengan menambahkan pernyataan berikut untuk meminta Amazon Glacier untuk menyiapkan hanya 1 MB hingga 2 MB bagian arsip.

```
int ONE_MEG = 1048576;
String retrievalByteRange = String.format("%s-%s", ONE_MEG, 2*ONE_MEG -1);

JobParameters jobParameters = new JobParameters()
    .withType("archive-retrieval")
    .withArchiveId(archiveId)
    .withRetrievalByteRange(retrievalByteRange)
    .withSNSTopic(snsTopicARN);

InitiateJobResult initiateJobResult = client.initiateJob(new InitiateJobRequest()
    .withJobParameters(jobParameters)
    .withVaultName(vaultName));

String jobId = initiateJobResult.getJobId();
```

3. Tunggu hingga tugas selesai.

   Anda harus menunggu hingga output tugas siap diunduh. Jika Anda telah menyetel konfigurasi notifikasi di vault yang mengidentifikasi topik Amazon Simple Notification Service (Amazon SNS) atau menetapkan topik Amazon SNS saat memulai pekerjaan, Amazon Glacier akan mengirimkan pesan ke topik tersebut setelah menyelesaikan pekerjaan.

   Anda juga dapat melakukan polling Amazon Glacier dengan memanggil `describeJob` metode untuk menentukan status penyelesaian pekerjaan. Meskipun, menggunakan topik Amazon SNS untuk notifikasi adalah pendekatan yang disarankan.

4. Unduh output tugas (data arsip) dengan menjalankan metode `getJobOutput`.

   Anda memberikan informasi permintaan seperti ID tugas dan nama vault dengan membuat instans dari kelas `GetJobOutputRequest`. Output yang dikembalikan Amazon Glacier tersedia di objek. `GetJobOutputResult`

```
GetJobOutputRequest jobOutputRequest = new GetJobOutputRequest()
        .withJobId("*** provide a job ID ***")
        .withVaultName("*** provide a vault name ****");
GetJobOutputResult jobOutputResult = client.getJobOutput(jobOutputRequest);

// jobOutputResult.getBody() // Provides the input stream.
```

Potongan kode sebelumnya mengunduh seluruh output tugas. Anda secara opsional dapat mengambil hanya sebagian output, atau mengunduh seluruh output dalam potongan yang lebih kecil dengan menentukan rentang byte di `GetJobOutputRequest` Anda.

```
GetJobOutputRequest jobOutputRequest = new GetJobOutputRequest()
        .withJobId("*** provide a job ID ***")
        .withRange("bytes=0-1048575")   // Download only the first 1 MB of the
 output.
        .withVaultName("*** provide a vault name ****");
```

Menanggapi `GetJobOutput` panggilan Anda, Amazon Glacier mengembalikan checksum bagian data yang Anda unduh, jika kondisi tertentu terpenuhi. Untuk informasi selengkapnya, lihat Menerima Checksum Saat Mengunduh Data.

Untuk memverifikasi tidak ada kesalahan dalam unduhan, Anda kemudian dapat menghitung checksum di sisi klien dan membandingkannya dengan checksum Amazon Glacier yang dikirim dalam respons.

Untuk pekerjaan pengambilan arsip dengan rentang opsional yang ditentukan, ketika Anda mendapatkan deskripsi pekerjaan, itu termasuk checksum dari rentang yang Anda ambil (). SHA256 TreeHash Anda dapat menggunakan nilai ini untuk memverifikasi lebih lanjut akurasi seluruh rentang byte yang Anda unduh selanjutnya. Misalnya, jika Anda memulai tugas untuk mengambil rentang arsip hash pohon selaras, lalu mengunduh output dalam potongan sehingga masing-masing permintaan `GetJobOutput` Anda mengembalikan checksum, Anda dapat menghitung checksum setiap bagian yang Anda unduh di sisi klien, lalu menghitung hash pohon. Anda dapat membandingkannya dengan pengembalian Amazon Glacier checksum sebagai tanggapan atas permintaan pekerjaan deskripsikan Anda untuk memverifikasi bahwa seluruh rentang byte yang telah Anda unduh sama dengan rentang byte yang disimpan di Amazon Glacier.

Untuk contoh pekerjaan, lihat Contoh 2: Mengambil Arsip Menggunakan API Tingkat Rendah dari AWS SDK for Java—Download Output in Chunks.

Contoh 1: Mengambil Arsip Menggunakan API Tingkat Rendah AWS SDK for Java

Contoh kode Java berikut mengunduh arsip dari vault yang ditentukan. Setelah tugas selesai, contoh mengunduh seluruh output dalam satu panggilan `getJobOutput`. Untuk contoh mengunduh output dalam potongan, lihat Contoh 2: Mengambil Arsip Menggunakan API Tingkat Rendah dari AWS SDK for Java—Download Output in Chunks.

Contoh tersebut melakukan tugas-tugas berikut:

- Membuat topik Amazon Simple Notification Service (Amazon SNS).

  Amazon Glacier mengirimkan pemberitahuan ke topik ini setelah menyelesaikan pekerjaan.
- Membuat antrean Amazon Simple Queue Service (Amazon SQS).

  Contoh melampirkan kebijakan ke antrean untuk mengizinkan topik Amazon SNS mengirim pesan ke antrean.
- Memulai tugas untuk mengunduh arsip yang ditentukan.

  Dalam permintaan pekerjaan, topik Amazon SNS yang dibuat ditentukan sehingga Amazon Glacier dapat mempublikasikan pemberitahuan ke topik setelah menyelesaikan pekerjaan.
- Memeriksa antrean Amazon SQS secara berkala untuk pesan yang berisi ID tugas.

  Jika ada pesan, urai JSON dan periksa apakah tugas berhasil diselesaikan. Jika ya, unduh arsipnya.
- Membersihkan dengan menghapus topik Amazon SNS dan antrean Amazon SQS yang dibuat.

```
import java.io.BufferedInputStream;
import java.io.BufferedOutputStream;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileOutputStream;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStream;
```

```
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import org.codehaus.jackson.JsonFactory;
import org.codehaus.jackson.JsonNode;
import org.codehaus.jackson.JsonParseException;
import org.codehaus.jackson.JsonParser;
import org.codehaus.jackson.map.ObjectMapper;

import com.amazonaws.AmazonClientException;
import com.amazonaws.auth.policy.Policy;
import com.amazonaws.auth.policy.Principal;
import com.amazonaws.auth.policy.Resource;
import com.amazonaws.auth.policy.Statement;
import com.amazonaws.auth.policy.Statement.Effect;
import com.amazonaws.auth.policy.actions.SQSActions;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.glacier.AmazonGlacierClient;
import com.amazonaws.services.glacier.model.GetJobOutputRequest;
import com.amazonaws.services.glacier.model.GetJobOutputResult;
import com.amazonaws.services.glacier.model.InitiateJobRequest;
import com.amazonaws.services.glacier.model.InitiateJobResult;
import com.amazonaws.services.glacier.model.JobParameters;
import com.amazonaws.services.sns.AmazonSNSClient;
import com.amazonaws.services.sns.model.CreateTopicRequest;
import com.amazonaws.services.sns.model.CreateTopicResult;
import com.amazonaws.services.sns.model.DeleteTopicRequest;
import com.amazonaws.services.sns.model.SubscribeRequest;
import com.amazonaws.services.sns.model.SubscribeResult;
import com.amazonaws.services.sns.model.UnsubscribeRequest;
import com.amazonaws.services.sqs.AmazonSQSClient;
import com.amazonaws.services.sqs.model.CreateQueueRequest;
import com.amazonaws.services.sqs.model.CreateQueueResult;
import com.amazonaws.services.sqs.model.DeleteQueueRequest;
import com.amazonaws.services.sqs.model.GetQueueAttributesRequest;
import com.amazonaws.services.sqs.model.GetQueueAttributesResult;
import com.amazonaws.services.sqs.model.Message;
import com.amazonaws.services.sqs.model.ReceiveMessageRequest;
import com.amazonaws.services.sqs.model.SetQueueAttributesRequest;
```

```
public class AmazonGlacierDownloadArchiveWithSQSPolling {

    public static String archiveId = "*** provide archive ID ****";
    public static String vaultName = "*** provide vault name ***";
    public static String snsTopicName = "*** provide topic name ***";
    public static String sqsQueueName = "*** provide queue name ***";
    public static String sqsQueueARN;
    public static String sqsQueueURL;
    public static String snsTopicARN;
    public static String snsSubscriptionARN;
    public static String fileName = "*** provide file name ***";
    public static String region = "*** region ***";
    public static long sleepTime = 600;
    public static AmazonGlacierClient client;
    public static AmazonSQSClient sqsClient;
    public static AmazonSNSClient snsClient;

    public static void main(String[] args) throws IOException {

     ProfileCredentialsProvider credentials = new ProfileCredentialsProvider();

        client = new AmazonGlacierClient(credentials);
        client.setEndpoint("https://glacier." + region + ".amazonaws.com");
        sqsClient = new AmazonSQSClient(credentials);
        sqsClient.setEndpoint("https://sqs." + region + ".amazonaws.com");
        snsClient = new AmazonSNSClient(credentials);
        snsClient.setEndpoint("https://sns." + region + ".amazonaws.com");

        try {
            setupSQS();

            setupSNS();

            String jobId = initiateJobRequest();
            System.out.println("Jobid = " + jobId);

            Boolean success = waitForJobToComplete(jobId, sqsQueueURL);
            if (!success) { throw new Exception("Job did not complete
 successfully."); }

            downloadJobOutput(jobId);

            cleanUp();
```

```
        } catch (Exception e) {
            System.err.println("Archive retrieval failed.");
            System.err.println(e);
        }
    }

    private static void setupSQS() {
        CreateQueueRequest request = new CreateQueueRequest()
            .withQueueName(sqsQueueName);
        CreateQueueResult result = sqsClient.createQueue(request);
        sqsQueueURL = result.getQueueUrl();

        GetQueueAttributesRequest qRequest = new GetQueueAttributesRequest()
            .withQueueUrl(sqsQueueURL)
            .withAttributeNames("QueueArn");

        GetQueueAttributesResult qResult = sqsClient.getQueueAttributes(qRequest);
        sqsQueueARN = qResult.getAttributes().get("QueueArn");

        Policy sqsPolicy =
            new Policy().withStatements(
                    new Statement(Effect.Allow)
                    .withPrincipals(Principal.AllUsers)
                    .withActions(SQSActions.SendMessage)
                    .withResources(new Resource(sqsQueueARN)));
        Map<String, String> queueAttributes = new HashMap<String, String>();
        queueAttributes.put("Policy", sqsPolicy.toJson());
        sqsClient.setQueueAttributes(new SetQueueAttributesRequest(sqsQueueURL,
queueAttributes));

    }
    private static void setupSNS() {
        CreateTopicRequest request = new CreateTopicRequest()
            .withName(snsTopicName);
        CreateTopicResult result = snsClient.createTopic(request);
        snsTopicARN = result.getTopicArn();

        SubscribeRequest request2 = new SubscribeRequest()
            .withTopicArn(snsTopicARN)
            .withEndpoint(sqsQueueARN)
            .withProtocol("sqs");
        SubscribeResult result2 = snsClient.subscribe(request2);

        snsSubscriptionARN = result2.getSubscriptionArn();
```

```
    }
    private static String initiateJobRequest() {

        JobParameters jobParameters = new JobParameters()
            .withType("archive-retrieval")
            .withArchiveId(archiveId)
            .withSNSTopic(snsTopicARN);

        InitiateJobRequest request = new InitiateJobRequest()
            .withVaultName(vaultName)
            .withJobParameters(jobParameters);

        InitiateJobResult response = client.initiateJob(request);

        return response.getJobId();
    }

    private static Boolean waitForJobToComplete(String jobId, String sqsQueueUrl)
throws InterruptedException, JsonParseException, IOException {

        Boolean messageFound = false;
        Boolean jobSuccessful = false;
        ObjectMapper mapper = new ObjectMapper();
        JsonFactory factory = mapper.getJsonFactory();

        while (!messageFound) {
            List<Message> msgs = sqsClient.receiveMessage(
               new
ReceiveMessageRequest(sqsQueueUrl).withMaxNumberOfMessages(10)).getMessages();

            if (msgs.size() > 0) {
                for (Message m : msgs) {
                    JsonParser jpMessage = factory.createJsonParser(m.getBody());
                    JsonNode jobMessageNode = mapper.readTree(jpMessage);
                    String jobMessage = jobMessageNode.get("Message").getTextValue();

                    JsonParser jpDesc = factory.createJsonParser(jobMessage);
                    JsonNode jobDescNode = mapper.readTree(jpDesc);
                    String retrievedJobId = jobDescNode.get("JobId").getTextValue();
                    String statusCode = jobDescNode.get("StatusCode").getTextValue();
                    if (retrievedJobId.equals(jobId)) {
                        messageFound = true;
                        if (statusCode.equals("Succeeded")) {
                            jobSuccessful = true;
```

```
                        }
                    }
                }

            } else {
              Thread.sleep(sleepTime * 1000);
            }
          }
        return (messageFound && jobSuccessful);
    }

    private static void downloadJobOutput(String jobId) throws IOException {

        GetJobOutputRequest getJobOutputRequest = new GetJobOutputRequest()
            .withVaultName(vaultName)
            .withJobId(jobId);
        GetJobOutputResult getJobOutputResult =
client.getJobOutput(getJobOutputRequest);

        InputStream input = new BufferedInputStream(getJobOutputResult.getBody());
        OutputStream output = null;
        try {
            output = new BufferedOutputStream(new FileOutputStream(fileName));

            byte[] buffer = new byte[1024 * 1024];

            int bytesRead = 0;
            do {
                bytesRead = input.read(buffer);
                if (bytesRead <= 0) break;
                output.write(buffer, 0, bytesRead);
            } while (bytesRead > 0);
        } catch (IOException e) {
            throw new AmazonClientException("Unable to save archive", e);
        } finally {
            try {input.close();}  catch (Exception e) {}
            try {output.close();} catch (Exception e) {}
        }
        System.out.println("Retrieved archive to " + fileName);
    }

    private static void cleanUp() {
        snsClient.unsubscribe(new UnsubscribeRequest(snsSubscriptionARN));
        snsClient.deleteTopic(new DeleteTopicRequest(snsTopicARN));
```

```
        sqsClient.deleteQueue(new DeleteQueueRequest(sqsQueueURL));
    }
}
```

Contoh 2: Mengambil Arsip Menggunakan API Tingkat Rendah dari AWS SDK for Java—Download Output in Chunks

Contoh kode Java berikut mengambil arsip dari Amazon Glacier. Contoh kode mengunduh output tugas dalam potongan dengan menentukan rentang byte dalam objek `GetJobOutputRequest`.

```
import java.io.BufferedInputStream;
import java.io.ByteArrayInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import com.fasterxml.jackson.core.JsonFactory;
import com.fasterxml.jackson.core.JsonParseException;
import com.fasterxml.jackson.core.JsonParser;
import com.fasterxml.jackson.databind.JsonNode;
import com.fasterxml.jackson.databind.ObjectMapper;

import com.amazonaws.auth.policy.Policy;
import com.amazonaws.auth.policy.Principal;
import com.amazonaws.auth.policy.Resource;
import com.amazonaws.auth.policy.Statement;
import com.amazonaws.auth.policy.Statement.Effect;
import com.amazonaws.auth.policy.actions.SQSActions;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.glacier.AmazonGlacierClient;
import com.amazonaws.services.glacier.TreeHashGenerator;
import com.amazonaws.services.glacier.model.GetJobOutputRequest;
import com.amazonaws.services.glacier.model.GetJobOutputResult;
import com.amazonaws.services.glacier.model.InitiateJobRequest;
import com.amazonaws.services.glacier.model.InitiateJobResult;
import com.amazonaws.services.glacier.model.JobParameters;
import com.amazonaws.services.sns.AmazonSNSClient;
import com.amazonaws.services.sns.model.CreateTopicRequest;
import com.amazonaws.services.sns.model.CreateTopicResult;
import com.amazonaws.services.sns.model.DeleteTopicRequest;
```

```
import com.amazonaws.services.sns.model.SubscribeRequest;
import com.amazonaws.services.sns.model.SubscribeResult;
import com.amazonaws.services.sns.model.UnsubscribeRequest;
import com.amazonaws.services.sqs.AmazonSQSClient;
import com.amazonaws.services.sqs.model.CreateQueueRequest;
import com.amazonaws.services.sqs.model.CreateQueueResult;
import com.amazonaws.services.sqs.model.DeleteQueueRequest;
import com.amazonaws.services.sqs.model.GetQueueAttributesRequest;
import com.amazonaws.services.sqs.model.GetQueueAttributesResult;
import com.amazonaws.services.sqs.model.Message;
import com.amazonaws.services.sqs.model.ReceiveMessageRequest;
import com.amazonaws.services.sqs.model.SetQueueAttributesRequest;


public class ArchiveDownloadLowLevelWithRange {

    public static String vaultName = "*** provide vault name ***";
    public static String archiveId = "*** provide archive id ***";
    public static String snsTopicName = "glacier-temp-sns-topic";
    public static String sqsQueueName = "glacier-temp-sqs-queue";
    public static long downloadChunkSize = 4194304; // 4 MB
    public static String sqsQueueARN;
    public static String sqsQueueURL;
    public static String snsTopicARN;
    public static String snsSubscriptionARN;
    public static String fileName = "*** provide file name to save archive to ***";
    public static String region   = "*** region ***";
    public static long sleepTime  = 600;

    public static AmazonGlacierClient client;
    public static AmazonSQSClient sqsClient;
    public static AmazonSNSClient snsClient;

    public static void main(String[] args) throws IOException {

     ProfileCredentialsProvider credentials = new ProfileCredentialsProvider();

        client = new AmazonGlacierClient(credentials);
        client.setEndpoint("https://glacier." + region + ".amazonaws.com");
        sqsClient = new AmazonSQSClient(credentials);
        sqsClient.setEndpoint("https://sqs." + region + ".amazonaws.com");
        snsClient = new AmazonSNSClient(credentials);
        snsClient.setEndpoint("https://sns." + region + ".amazonaws.com");
```

```
        try {
            setupSQS();

            setupSNS();

            String jobId = initiateJobRequest();
            System.out.println("Jobid = " + jobId);

            long archiveSizeInBytes = waitForJobToComplete(jobId, sqsQueueURL);
            if (archiveSizeInBytes==-1) { throw new Exception("Job did not complete
successfully."); }

            downloadJobOutput(jobId, archiveSizeInBytes);

            cleanUp();

        } catch (Exception e) {
            System.err.println("Archive retrieval failed.");
            System.err.println(e);
        }
    }

    private static void setupSQS() {
        CreateQueueRequest request = new CreateQueueRequest()
            .withQueueName(sqsQueueName);
        CreateQueueResult result = sqsClient.createQueue(request);
        sqsQueueURL = result.getQueueUrl();

        GetQueueAttributesRequest qRequest = new GetQueueAttributesRequest()
            .withQueueUrl(sqsQueueURL)
            .withAttributeNames("QueueArn");

        GetQueueAttributesResult qResult = sqsClient.getQueueAttributes(qRequest);
        sqsQueueARN = qResult.getAttributes().get("QueueArn");

        Policy sqsPolicy =
            new Policy().withStatements(
                    new Statement(Effect.Allow)
                    .withPrincipals(Principal.AllUsers)
                    .withActions(SQSActions.SendMessage)
                    .withResources(new Resource(sqsQueueARN)));
        Map<String, String> queueAttributes = new HashMap<String, String>();
        queueAttributes.put("Policy", sqsPolicy.toJson());
```

```
        sqsClient.setQueueAttributes(new SetQueueAttributesRequest(sqsQueueURL,
queueAttributes));

    }
    private static void setupSNS() {
        CreateTopicRequest request = new CreateTopicRequest()
            .withName(snsTopicName);
        CreateTopicResult result = snsClient.createTopic(request);
        snsTopicARN = result.getTopicArn();

        SubscribeRequest request2 = new SubscribeRequest()
            .withTopicArn(snsTopicARN)
            .withEndpoint(sqsQueueARN)
            .withProtocol("sqs");
        SubscribeResult result2 = snsClient.subscribe(request2);

        snsSubscriptionARN = result2.getSubscriptionArn();
    }
    private static String initiateJobRequest() {

        JobParameters jobParameters = new JobParameters()
            .withType("archive-retrieval")
            .withArchiveId(archiveId)
            .withSNSTopic(snsTopicARN);

        InitiateJobRequest request = new InitiateJobRequest()
            .withVaultName(vaultName)
            .withJobParameters(jobParameters);

        InitiateJobResult response = client.initiateJob(request);

        return response.getJobId();
    }

    private static long waitForJobToComplete(String jobId, String sqsQueueUrl) throws
InterruptedException, JsonParseException, IOException {

        Boolean messageFound = false;
        Boolean jobSuccessful = false;
        long archiveSizeInBytes = -1;
        ObjectMapper mapper = new ObjectMapper();
        JsonFactory factory = mapper.getFactory();

        while (!messageFound) {
```

```
            List<Message> msgs = sqsClient.receiveMessage(
               new
ReceiveMessageRequest(sqsQueueUrl).withMaxNumberOfMessages(10)).getMessages();

            if (msgs.size() > 0) {
                for (Message m : msgs) {
                    JsonParser jpMessage = factory.createJsonParser(m.getBody());
                    JsonNode jobMessageNode = mapper.readTree(jpMessage);
                    String jobMessage = jobMessageNode.get("Message").textValue();

                    JsonParser jpDesc = factory.createJsonParser(jobMessage);
                    JsonNode jobDescNode = mapper.readTree(jpDesc);
                    String retrievedJobId = jobDescNode.get("JobId").textValue();
                    String statusCode = jobDescNode.get("StatusCode").textValue();
                    archiveSizeInBytes =
jobDescNode.get("ArchiveSizeInBytes").longValue();
                    if (retrievedJobId.equals(jobId)) {
                        messageFound = true;
                        if (statusCode.equals("Succeeded")) {
                            jobSuccessful = true;
                        }
                    }
                }

            } else {
              Thread.sleep(sleepTime * 1000);
            }
          }
        return (messageFound && jobSuccessful) ? archiveSizeInBytes : -1;
    }

    private static void downloadJobOutput(String jobId, long archiveSizeInBytes) throws
IOException {

        if (archiveSizeInBytes < 0) {
            System.err.println("Nothing to download.");
            return;
        }

        System.out.println("archiveSizeInBytes: " + archiveSizeInBytes);
        FileOutputStream fstream = new FileOutputStream(fileName);
        long startRange = 0;
        long endRange = (downloadChunkSize > archiveSizeInBytes) ? archiveSizeInBytes
-1 : downloadChunkSize - 1;
```

```
        do {

            GetJobOutputRequest getJobOutputRequest = new GetJobOutputRequest()
                .withVaultName(vaultName)
                .withRange("bytes=" + startRange + "-" + endRange)
                .withJobId(jobId);
            GetJobOutputResult getJobOutputResult =
client.getJobOutput(getJobOutputRequest);

            BufferedInputStream is = new
BufferedInputStream(getJobOutputResult.getBody());
            byte[] buffer = new byte[(int)(endRange - startRange + 1)];

            System.out.println("Checksum received: " +
getJobOutputResult.getChecksum());
            System.out.println("Content range " +
getJobOutputResult.getContentRange());


            int totalRead = 0;
            while (totalRead < buffer.length) {
                int bytesRemaining = buffer.length - totalRead;
                int read = is.read(buffer, totalRead, bytesRemaining);
                if (read > 0) {
                    totalRead = totalRead + read;
                } else {
                    break;
                }

            }
            System.out.println("Calculated checksum: " +
TreeHashGenerator.calculateTreeHash(new ByteArrayInputStream(buffer)));
            System.out.println("read = " + totalRead);
            fstream.write(buffer);

            startRange = startRange + (long)totalRead;
            endRange = ((endRange + downloadChunkSize) >  archiveSizeInBytes) ?
archiveSizeInBytes : (endRange + downloadChunkSize);
            is.close();
        } while (endRange <= archiveSizeInBytes  && startRange < archiveSizeInBytes);

        fstream.close();
        System.out.println("Retrieved file to " + fileName);
```

```
    }

    private static void cleanUp() {
        snsClient.unsubscribe(new UnsubscribeRequest(snsSubscriptionARN));
        snsClient.deleteTopic(new DeleteTopicRequest(snsTopicARN));
        sqsClient.deleteQueue(new DeleteQueueRequest(sqsQueueURL));
    }
}
```

Mengunduh Arsip di Amazon Glacier Menggunakan AWS SDK for .NET

Baik tingkat tinggi dan tingkat rendah yang APIs disediakan oleh Amazon SDK untuk.NET menyediakan metode untuk mengunduh arsip.

Topik

- Mengunduh Arsip Menggunakan API Tingkat Tinggi dari AWS SDK for .NET
- Mengunduh Arsip Menggunakan API Tingkat Rendah dari AWS SDK for .NET

Mengunduh Arsip Menggunakan API Tingkat Tinggi dari AWS SDK for .NET

Kelas `ArchiveTransferManager` dari API tingkat tinggi menyediakan metode `Download` yang dapat Anda gunakan untuk mengunduh arsip.

> **Important**
>
> Kelas `ArchiveTransferManager` membuat topik Amazon Simple Notification Service (Amazon SNS) dan antrean Amazon Simple Queue Service (Amazon SQS) yang berlangganan topik tersebut. Kelas tersebut selanjutnya memulai tugas pengambilan arsip dan melakukan polling antrean untuk arsip yang akan tersedia. Ketika arsip tersedia, unduhan dimulai. Untuk informasi tentang waktu pengambilan, lihat Opsi Pengambilan Arsip

Contoh: Mengunduh Arsip Menggunakan API Tingkat Tinggi AWS SDK for .NET

Contoh kode C# berikut mengunduh arsip dari vault (`examplevault`) di Wilayah US West (Oregon).

Untuk step-by-step petunjuk tentang cara menjalankan contoh ini, lihatMenjalankan Contoh Kode. Anda perlu memperbarui kode seperti yang ditunjukkan dengan ID arsip yang ada dan jalur file lokal tempat Anda ingin menyimpan arsip yang diunduh.

```
using System;
using Amazon.Glacier;
using Amazon.Glacier.Transfer;
using Amazon.Runtime;

namespace glacier.amazon.com.docsamples
{
  class ArchiveDownloadHighLevel
  {
    static string vaultName        = "examplevault";
    static string archiveId        = "*** Provide archive ID ***";
    static string downloadFilePath = "*** Provide the file name and path to where to
 store the download ***";

    public static void Main(string[] args)
    {
      try
      {
        var manager = new ArchiveTransferManager(Amazon.RegionEndpoint.USWest2);

        var options = new DownloadOptions();
        options.StreamTransferProgress += ArchiveDownloadHighLevel.progress;
        // Download an archive.
        Console.WriteLine("Intiating the archive retrieval job and then polling SQS
 queue for the archive to be available.");
        Console.WriteLine("Once the archive is available, downloading will begin.");
        manager.Download(vaultName, archiveId, downloadFilePath, options);
        Console.WriteLine("To continue, press Enter");
        Console.ReadKey();
      }
      catch (AmazonGlacierException e) { Console.WriteLine(e.Message); }
      catch (AmazonServiceException e) { Console.WriteLine(e.Message); }
      catch (Exception e) { Console.WriteLine(e.Message); }
      Console.WriteLine("To continue, press Enter");
      Console.ReadKey();
    }

    static int currentPercentage = -1;
    static void progress(object sender, StreamTransferProgressArgs args)
    {
      if (args.PercentDone != currentPercentage)
      {
        currentPercentage = args.PercentDone;
```

```
        Console.WriteLine("Downloaded {0}%", args.PercentDone);
      }
    }
  }
}
```

Mengunduh Arsip Menggunakan API Tingkat Rendah dari AWS SDK for .NET

Berikut ini adalah langkah-langkah untuk mengunduh arsip Amazon Glacier (Amazon Glacier) menggunakan API tingkat rendah. AWS SDK for .NET

1. Buat instans dari kelas `AmazonGlacierClient` (klien).

   Anda perlu menentukan AWS Wilayah dari mana Anda ingin mengunduh arsip. Semua operasi yang Anda lakukan menggunakan klien ini berlaku untuk AWS Wilayah tersebut.

2. Mulai tugas `archive-retrieval` dengan menjalankan metode `InitiateJob`.

   Anda memberikan informasi pekerjaan, seperti ID arsip arsip yang ingin Anda unduh dan topik Amazon SNS opsional yang Anda inginkan Amazon Glacier untuk memposting pesan penyelesaian pekerjaan, dengan membuat instance kelas. `InitiateJobRequest` Amazon Glacier mengembalikan ID pekerjaan sebagai tanggapan. Respons tersedia dalam instans dari kelas `InitiateJobResponse`.

   ```
   AmazonGlacierClient client;
   client = new AmazonGlacierClient(Amazon.RegionEndpoint.USWest2);

   InitiateJobRequest initJobRequest = new InitiateJobRequest()
   {
     VaultName = vaultName,
     JobParameters = new JobParameters()
     {
       Type      = "archive-retrieval",
       ArchiveId = "*** Provide archive id ***",
       SNSTopic  = "*** Provide Amazon SNS topic ARN ***",
     }
   };

   InitiateJobResponse initJobResponse = client.InitiateJob(initJobRequest);
   string jobId = initJobResponse.JobId;
   ```

Anda dapat secara opsional menentukan rentang byte untuk meminta Amazon Glacier untuk menyiapkan hanya sebagian dari arsip seperti yang ditunjukkan dalam permintaan berikut. Permintaan menentukan Amazon Glacier untuk menyiapkan hanya 1 MB hingga 2 MB bagian arsip.

```
AmazonGlacierClient client;
client = new AmazonGlacierClient(Amazon.RegionEndpoint.USWest2);


InitiateJobRequest initJobRequest = new InitiateJobRequest()
{
  VaultName = vaultName,
  JobParameters = new JobParameters()
  {
    Type      = "archive-retrieval",
    ArchiveId = "*** Provide archive id ***",
    SNSTopic  = "*** Provide Amazon SNS topic ARN ***",
  }
};
// Specify byte range.
int ONE_MEG = 1048576;
initJobRequest.JobParameters.RetrievalByteRange = string.Format("{0}-{1}", ONE_MEG, 2
 * ONE_MEG -1);

InitiateJobResponse initJobResponse = client.InitiateJob(initJobRequest);
string jobId = initJobResponse.JobId;
```

3. Tunggu hingga tugas selesai.

   Anda harus menunggu hingga output tugas siap diunduh. Jika Anda telah menyetel konfigurasi notifikasi di vault yang mengidentifikasi topik Amazon Simple Notification Service (Amazon SNS) atau menetapkan topik Amazon SNS saat memulai pekerjaan, Amazon Glacier akan mengirimkan pesan ke topik tersebut setelah menyelesaikan pekerjaan. Contoh kode yang diberikan di bagian berikut menggunakan Amazon SNS untuk Amazon Glacier untuk mempublikasikan pesan.

   Anda juga dapat melakukan polling Amazon Glacier dengan memanggil `DescribeJob` metode untuk menentukan status penyelesaian pekerjaan. Meskipun, menggunakan topik Amazon SNS untuk notifikasi adalah pendekatan yang disarankan.

4. Unduh output tugas (data arsip) dengan menjalankan metode `GetJobOutput`.

Anda memberikan informasi permintaan seperti ID tugas dan nama vault dengan membuat instans dari kelas `GetJobOutputRequest`. Output yang dikembalikan Amazon Glacier tersedia di objek. `GetJobOutputResponse`

```
GetJobOutputRequest getJobOutputRequest = new GetJobOutputRequest()
{
  JobId = jobId,
  VaultName = vaultName
};

GetJobOutputResponse getJobOutputResponse = client.GetJobOutput(getJobOutputRequest);
using (Stream webStream = getJobOutputResponse.Body)
{
  using (Stream fileToSave = File.OpenWrite(fileName))
  {
     CopyStream(webStream, fileToSave);
  }
}
```

Potongan kode sebelumnya mengunduh seluruh output tugas. Anda secara opsional dapat mengambil hanya sebagian output, atau mengunduh seluruh output dalam potongan yang lebih kecil dengan menentukan rentang byte di `GetJobOutputRequest` Anda.

```
GetJobOutputRequest getJobOutputRequest = new GetJobOutputRequest()
{
  JobId = jobId,
  VaultName = vaultName
};
getJobOutputRequest.SetRange(0, 1048575); // Download only the first 1 MB chunk of
 the output.
```

Menanggapi `GetJobOutput` panggilan Anda, Amazon Glacier mengembalikan checksum bagian data yang Anda unduh, jika kondisi tertentu terpenuhi. Untuk informasi selengkapnya, lihat Menerima Checksum Saat Mengunduh Data.

Untuk memverifikasi tidak ada kesalahan dalam unduhan, Anda kemudian dapat menghitung checksum di sisi klien dan membandingkannya dengan checksum Amazon Glacier yang dikirim dalam respons.

Untuk pekerjaan pengambilan arsip dengan rentang opsional yang ditentukan, ketika Anda mendapatkan deskripsi pekerjaan, itu termasuk checksum dari rentang yang Anda ambil (SHA256TreeHash) .Anda dapat menggunakan nilai ini untuk memverifikasi lebih lanjut keakuratan seluruh rentang byte yang kemudian Anda unduh. Misalnya, jika Anda memulai tugas untuk mengambil rentang arsip hash pohon selaras, lalu mengunduh output dalam potongan sehingga masing-masing permintaan `GetJobOutput` Anda mengembalikan checksum, Anda dapat menghitung checksum setiap bagian yang Anda unduh di sisi klien, lalu menghitung hash pohon. Anda dapat membandingkannya dengan pengembalian Amazon Glacier checksum sebagai tanggapan atas permintaan pekerjaan deskripsikan Anda untuk memverifikasi bahwa seluruh rentang byte yang telah Anda unduh sama dengan rentang byte yang disimpan di Amazon Glacier.

Untuk contoh pekerjaan, lihat Contoh 2: Mengambil Arsip Menggunakan API Tingkat Rendah dari AWS SDK for .NET—Download Output in Chunks.

Contoh 1: Mengambil Arsip Menggunakan API Tingkat Rendah AWS SDK for .NET

Contoh kode C# berikut mengunduh arsip dari vault yang ditentukan. Setelah tugas selesai, contoh mengunduh seluruh output dalam satu panggilan `GetJobOutput`. Untuk contoh mengunduh output dalam potongan, lihat Contoh 2: Mengambil Arsip Menggunakan API Tingkat Rendah dari AWS SDK for .NET—Download Output in Chunks.

Contoh tersebut melakukan tugas-tugas berikut:

- Menyiapkan topik Amazon Simple Notification Service (Amazon SNS)

  Amazon Glacier mengirimkan pemberitahuan ke topik ini setelah menyelesaikan pekerjaan.
- Menyiapkan antrean Amazon Simple Queue Service (Amazon SQS).

  Contoh melampirkan kebijakan ke antrean untuk mengizinkan topik Amazon SNS mengirim pesan.
- Memulai tugas untuk mengunduh arsip yang ditentukan.

  Dalam permintaan pekerjaan, contoh menentukan topik Amazon SNS sehingga Amazon Glacier dapat mengirim pesan setelah menyelesaikan pekerjaan.
- Memeriksa antrean Amazon SQS secara berkala untuk pesan.

  Jika ada pesan, urai JSON dan periksa apakah tugas berhasil diselesaikan. Jika ya, unduh arsipnya. Contoh kode menggunakan pustaka JSON.NET (lihat JSON.NET) untuk mengurai JSON.

- Membersihkan dengan menghapus topik Amazon SNS dan antrean Amazon SQS yang dibuat.

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Threading;
using Amazon.Glacier;
using Amazon.Glacier.Model;
using Amazon.Runtime;
using Amazon.SimpleNotificationService;
using Amazon.SimpleNotificationService.Model;
using Amazon.SQS;
using Amazon.SQS.Model;
using Newtonsoft.Json;

namespace glacier.amazon.com.docsamples
{
  class ArchiveDownloadLowLevelUsingSNSSQS
  {
    static string topicArn;
    static string queueUrl;
    static string queueArn;
    static string vaultName = "*** Provide vault name ***";
    static string archiveID = "*** Provide archive ID ***";
    static string fileName  = "*** Provide the file name and path to where to store
 downloaded archive ***";
    static AmazonSimpleNotificationServiceClient snsClient;
    static AmazonSQSClient sqsClient;
    const string SQS_POLICY =
        "{" +
        "    \"Version\" : \"2012-10-17\",&TCX5-2025-waiver;" +
        "    \"Statement\" : [" +
        "        {" +
        "            \"Sid\" : \"sns-rule\"," +
        "            \"Effect\" : \"Allow\"," +
        "            \"Principal\" : {\"Service\" : \"sns.amazonaws.com\" }," +
        "            \"Action\"    : \"sqs:SendMessage\"," +
        "            \"Resource\"  : \"{QueueArn}\"," +
        "            \"Condition\" : {" +
        "                \"ArnLike\" : {" +
        "                    \"aws:SourceArn\" : \"{TopicArn}\"" +
        "                }" +
        "            }" +
```

```
        "        }" +
        "    ]" +
        "}";

    public static void Main(string[] args)
    {
      AmazonGlacierClient client;
      try
      {
        using (client = new AmazonGlacierClient(Amazon.RegionEndpoint.USWest2))
        {
          Console.WriteLine("Setup SNS topic and SQS queue.");
          SetupTopicAndQueue();
          Console.WriteLine("To continue, press Enter"); Console.ReadKey();
          Console.WriteLine("Retrieving...");
          RetrieveArchive(client);
        }
        Console.WriteLine("Operations successful. To continue, press Enter");
        Console.ReadKey();
      }
      catch (AmazonGlacierException e) { Console.WriteLine(e.Message); }
      catch (AmazonServiceException e) { Console.WriteLine(e.Message); }
      catch (Exception e) { Console.WriteLine(e.Message); }
      finally
      {
        // Delete SNS topic and SQS queue.
        snsClient.DeleteTopic(new DeleteTopicRequest() { TopicArn = topicArn });
        sqsClient.DeleteQueue(new DeleteQueueRequest() { QueueUrl = queueUrl });
      }
    }

    static void SetupTopicAndQueue()
    {
      snsClient = new
AmazonSimpleNotificationServiceClient(Amazon.RegionEndpoint.USWest2);
      sqsClient = new AmazonSQSClient(Amazon.RegionEndpoint.USWest2);

      long ticks = DateTime.Now.Ticks;
      topicArn = snsClient.CreateTopic(new CreateTopicRequest { Name =
"GlacierDownload-" + ticks }).TopicArn;
      Console.Write("topicArn: "); Console.WriteLine(topicArn);

      CreateQueueRequest createQueueRequest = new CreateQueueRequest();
      createQueueRequest.QueueName = "GlacierDownload-" + ticks;
```

```
      CreateQueueResponse createQueueResponse =
sqsClient.CreateQueue(createQueueRequest);
      queueUrl = createQueueResponse.QueueUrl;
      Console.Write("QueueURL: "); Console.WriteLine(queueUrl);

      GetQueueAttributesRequest getQueueAttributesRequest = new
GetQueueAttributesRequest();
      getQueueAttributesRequest.AttributeNames = new List<string> { "QueueArn" };
      getQueueAttributesRequest.QueueUrl = queueUrl;
      GetQueueAttributesResponse response =
sqsClient.GetQueueAttributes(getQueueAttributesRequest);
      queueArn = response.QueueARN;
      Console.Write("QueueArn: "); Console.WriteLine(queueArn);

      // Setup the Amazon SNS topic to publish to the SQS queue.
      snsClient.Subscribe(new SubscribeRequest()
      {
        Protocol = "sqs",
        Endpoint = queueArn,
        TopicArn = topicArn
      });

      // Add policy to the queue so SNS can send messages to the queue.
      var policy = SQS_POLICY.Replace("{TopicArn}", topicArn).Replace("{QueueArn}",
queueArn);

      sqsClient.SetQueueAttributes(new SetQueueAttributesRequest()
      {
          QueueUrl = queueUrl,
          Attributes = new Dictionary<string, string>
          {
              { QueueAttributeName.Policy, policy }
          }
      });
    }

    static void RetrieveArchive(AmazonGlacierClient client)
    {
      // Initiate job.
      InitiateJobRequest initJobRequest = new InitiateJobRequest()
      {
        VaultName = vaultName,
        JobParameters = new JobParameters()
        {
```

```
          Type = "archive-retrieval",
          ArchiveId = archiveID,
          Description = "This job is to download archive.",
          SNSTopic = topicArn,
        }
      };
      InitiateJobResponse initJobResponse = client.InitiateJob(initJobRequest);
      string jobId = initJobResponse.JobId;

      // Check queue for a message and if job completed successfully, download archive.
      ProcessQueue(jobId, client);
    }

    private static void ProcessQueue(string jobId, AmazonGlacierClient client)
    {
      ReceiveMessageRequest receiveMessageRequest = new ReceiveMessageRequest()
{ QueueUrl = queueUrl, MaxNumberOfMessages = 1 };
      bool jobDone = false;
      while (!jobDone)
      {
        Console.WriteLine("Poll SQS queue");
        ReceiveMessageResponse receiveMessageResponse =
sqsClient.ReceiveMessage(receiveMessageRequest);
        if (receiveMessageResponse.Messages.Count == 0)
        {
          Thread.Sleep(10000 * 60);
          continue;
        }
        Console.WriteLine("Got message");
        Message message = receiveMessageResponse.Messages[0];
        Dictionary<string, string> outerLayer =
JsonConvert.DeserializeObject<Dictionary<string, string>>(message.Body);
        Dictionary<string, object> fields =
JsonConvert.DeserializeObject<Dictionary<string, object>>(outerLayer["Message"]);
        string statusCode = fields["StatusCode"] as string;

        if (string.Equals(statusCode, GlacierUtils.JOB_STATUS_SUCCEEDED,
StringComparison.InvariantCultureIgnoreCase))
        {
          Console.WriteLine("Downloading job output");
          DownloadOutput(jobId, client); // Save job output to the specified file
location.
        }
```

```
        else if (string.Equals(statusCode, GlacierUtils.JOB_STATUS_FAILED,
 StringComparison.InvariantCultureIgnoreCase))
          Console.WriteLine("Job failed... cannot download the archive.");

        jobDone = true;
        sqsClient.DeleteMessage(new DeleteMessageRequest() { QueueUrl = queueUrl,
 ReceiptHandle = message.ReceiptHandle });
      }
    }

    private static void DownloadOutput(string jobId, AmazonGlacierClient client)
    {
      GetJobOutputRequest getJobOutputRequest = new GetJobOutputRequest()
      {
        JobId = jobId,
        VaultName = vaultName
      };

      GetJobOutputResponse getJobOutputResponse =
 client.GetJobOutput(getJobOutputRequest);
      using (Stream webStream = getJobOutputResponse.Body)
      {
          using (Stream fileToSave = File.OpenWrite(fileName))
          {
              CopyStream(webStream, fileToSave);
          }
      }
    }

    public static void CopyStream(Stream input, Stream output)
    {
      byte[] buffer = new byte[65536];
      int length;
      while ((length = input.Read(buffer, 0, buffer.Length)) > 0)
      {
        output.Write(buffer, 0, length);
      }
    }
  }
}
```

Contoh 2: Mengambil Arsip Menggunakan API Tingkat Rendah dari AWS SDK for .NET—Download Output in Chunks

Contoh kode C # berikut mengambil arsip dari Amazon Glacier. Contoh kode mengunduh output tugas dalam potongan dengan menentukan rentang byte dalam objek `GetJobOutputRequest`.

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Threading;
using Amazon.Glacier;
using Amazon.Glacier.Model;
using Amazon.Glacier.Transfer;
using Amazon.Runtime;
using Amazon.SimpleNotificationService;
using Amazon.SimpleNotificationService.Model;
using Amazon.SQS;
using Amazon.SQS.Model;
using Newtonsoft.Json;
using System.Collections.Specialized;

namespace glacier.amazon.com.docsamples
{
  class ArchiveDownloadLowLevelUsingSQLSNSOutputUsingRange
  {
    static string topicArn;
    static string queueUrl;
    static string queueArn;
    static string vaultName = "*** Provide vault name ***";
    static string archiveId = "*** Provide archive ID ***";
    static string fileName  = "*** Provide the file name and path to where to store
 downloaded archive ***";
    static AmazonSimpleNotificationServiceClient snsClient;
    static AmazonSQSClient sqsClient;
    const string SQS_POLICY =
        "{" +
        "    \"Version\" : \"2012-10-17\",&TCX5-2025-waiver;" +
        "    \"Statement\" : [" +
        "        {" +
        "            \"Sid\" : \"sns-rule\"," +
        "            \"Effect\" : \"Allow\"," +
        "            \"Principal\" : {\"AWS\" : \"arn:aws:iam::123456789012:root\" },"
 +
```

```
        "            \"Action\"  : \"sqs:SendMessage\"," +
        "            \"Resource\"  : \"{QuernArn}\"," +
        "            \"Condition\" : {" +
        "                \"ArnLike\" : {" +
        "                    \"aws:SourceArn\" : \"{TopicArn}\"" +
        "                }" +
        "            }" +
        "        }" +
        "    ]" +
        "}";

    public static void Main(string[] args)
    {
      AmazonGlacierClient client;

      try
      {
          using (client = new AmazonGlacierClient(Amazon.RegionEndpoint.USWest2))
          {
              Console.WriteLine("Setup SNS topic and SQS queue.");
              SetupTopicAndQueue();
              Console.WriteLine("To continue, press Enter"); Console.ReadKey();

              Console.WriteLine("Download archive");
              DownloadAnArchive(archiveId, client);
        }
        Console.WriteLine("Operations successful. To continue, press Enter");
        Console.ReadKey();
      }
      catch (AmazonGlacierException e) { Console.WriteLine(e.Message); }
      catch (AmazonServiceException e) { Console.WriteLine(e.Message); }
      catch (Exception e) { Console.WriteLine(e.Message); }
      finally
      {
        // Delete SNS topic and SQS queue.
        snsClient.DeleteTopic(new DeleteTopicRequest() { TopicArn = topicArn });
        sqsClient.DeleteQueue(new DeleteQueueRequest() { QueueUrl = queueUrl });
      }
    }

       static void SetupTopicAndQueue()
    {
      long ticks = DateTime.Now.Ticks;
```

```
      // Setup SNS topic.
      snsClient = new
AmazonSimpleNotificationServiceClient(Amazon.RegionEndpoint.USWest2);
      sqsClient = new AmazonSQSClient(Amazon.RegionEndpoint.USWest2);

      topicArn = snsClient.CreateTopic(new CreateTopicRequest { Name =
"GlacierDownload-" + ticks }).TopicArn;
      Console.Write("topicArn: "); Console.WriteLine(topicArn);

      CreateQueueRequest createQueueRequest = new CreateQueueRequest();
      createQueueRequest.QueueName = "GlacierDownload-" + ticks;
      CreateQueueResponse createQueueResponse =
sqsClient.CreateQueue(createQueueRequest);
      queueUrl = createQueueResponse.QueueUrl;
      Console.Write("QueueURL: "); Console.WriteLine(queueUrl);

      GetQueueAttributesRequest getQueueAttributesRequest = new
GetQueueAttributesRequest();
      getQueueAttributesRequest.AttributeNames = new List<string> { "QueueArn" };
      getQueueAttributesRequest.QueueUrl = queueUrl;
      GetQueueAttributesResponse response =
sqsClient.GetQueueAttributes(getQueueAttributesRequest);
      queueArn = response.QueueARN;
      Console.Write("QueueArn: "); Console.WriteLine(queueArn);

      // Setup the Amazon SNS topic to publish to the SQS queue.
      snsClient.Subscribe(new SubscribeRequest()
      {
        Protocol = "sqs",
        Endpoint = queueArn,
        TopicArn = topicArn
      });

      // Add the policy to the queue so SNS can send messages to the queue.
      var policy = SQS_POLICY.Replace("{TopicArn}", topicArn).Replace("{QuernArn}",
queueArn);

      sqsClient.SetQueueAttributes(new SetQueueAttributesRequest()
      {
          QueueUrl = queueUrl,
          Attributes = new Dictionary<string, string>
          {
              { QueueAttributeName.Policy, policy }
          }
```

```
      });
    }

    static void DownloadAnArchive(string archiveId, AmazonGlacierClient client)
    {
      // Initiate job.
      InitiateJobRequest initJobRequest = new InitiateJobRequest()
      {

        VaultName = vaultName,
        JobParameters = new JobParameters()
        {
          Type = "archive-retrieval",
          ArchiveId = archiveId,
          Description = "This job is to download the archive.",
          SNSTopic = topicArn,
        }
      };
      InitiateJobResponse initJobResponse = client.InitiateJob(initJobRequest);
      string jobId = initJobResponse.JobId;

      // Check queue for a message and if job completed successfully, download archive.
      ProcessQueue(jobId, client);
    }

    private static void ProcessQueue(string jobId, AmazonGlacierClient client)
    {
        var receiveMessageRequest = new ReceiveMessageRequest() { QueueUrl = queueUrl,
MaxNumberOfMessages = 1 };
        bool jobDone = false;
        while (!jobDone)
        {
            Console.WriteLine("Poll SQS queue");
            ReceiveMessageResponse receiveMessageResponse =
sqsClient.ReceiveMessage(receiveMessageRequest);
            if (receiveMessageResponse.Messages.Count == 0)
            {
                Thread.Sleep(10000 * 60);
                continue;
            }
            Console.WriteLine("Got message");
            Message message = receiveMessageResponse.Messages[0];
            Dictionary<string, string> outerLayer =
JsonConvert.DeserializeObject<Dictionary<string, string>>(message.Body);
```

```
            Dictionary<string, object> fields =
JsonConvert.DeserializeObject<Dictionary<string, object>>(outerLayer["Message"]);
            string statusCode = fields["StatusCode"] as string;
            if (string.Equals(statusCode, GlacierUtils.JOB_STATUS_SUCCEEDED,
StringComparison.InvariantCultureIgnoreCase))
            {
                long archiveSize = Convert.ToInt64(fields["ArchiveSizeInBytes"]);
                Console.WriteLine("Downloading job output");
                DownloadOutput(jobId, archiveSize, client); // This where we save job
output to the specified file location.
            }
            else if (string.Equals(statusCode, GlacierUtils.JOB_STATUS_FAILED,
StringComparison.InvariantCultureIgnoreCase))
                Console.WriteLine("Job failed... cannot download the archive.");
            jobDone = true;
            sqsClient.DeleteMessage(new DeleteMessageRequest() { QueueUrl = queueUrl,
ReceiptHandle = message.ReceiptHandle });
        }
    }

    private static void DownloadOutput(string jobId, long archiveSize,
AmazonGlacierClient client)
    {
      long partSize = 4 * (long)Math.Pow(2, 20);  // 4 MB.
      using (Stream fileToSave = new FileStream(fileName, FileMode.Create,
FileAccess.Write))
      {

        long currentPosition = 0;
        do
        {
          GetJobOutputRequest getJobOutputRequest = new GetJobOutputRequest()
          {
            JobId = jobId,
            VaultName = vaultName
          };

          long endPosition = currentPosition + partSize - 1;
          if (endPosition > archiveSize)
            endPosition = archiveSize;

          getJobOutputRequest.SetRange(currentPosition, endPosition);
          GetJobOutputResponse getJobOutputResponse =
client.GetJobOutput(getJobOutputRequest);
```

```
          using (Stream webStream = getJobOutputResponse.Body)
          {
            CopyStream(webStream, fileToSave);
          }
          currentPosition += partSize;
        } while (currentPosition < archiveSize);
      }
    }

    public static void CopyStream(Stream input, Stream output)
    {
      byte[] buffer = new byte[65536];
      int length;
      while ((length = input.Read(buffer, 0, buffer.Length)) > 0)
      {
        output.Write(buffer, 0, length);
      }
    }
  }
}
```

Mengunduh Arsip Besar Menggunakan Pemrosesan Paralel dengan Python

Topik ini menjelaskan cara mengunduh arsip besar dari Amazon S3 Glacier (S3 Glacier) menggunakan pemrosesan paralel dengan Python. Pendekatan ini memungkinkan Anda mengunduh arsip dengan ukuran berapa pun dengan memecahnya menjadi potongan-potongan kecil yang dapat diproses secara independen.

Ikhtisar

Skrip Python yang disediakan dalam contoh ini melakukan tugas-tugas berikut:

1. Menetapkan AWS sumber daya yang diperlukan (topik Amazon SNS dan antrian Amazon SQS) untuk pemberitahuan
2. Memulai pekerjaan pengambilan arsip dengan Amazon Glacier
3. Memantau antrian Amazon SQS untuk pemberitahuan penyelesaian pekerjaan
4. Membagi arsip besar menjadi potongan-potongan yang dapat dikelola
5. Mengunduh potongan secara paralel menggunakan beberapa utas pekerja
6. Menyimpan setiap potongan ke disk untuk dipasang kembali nanti

Prasyarat

Sebelum memulai, pastikan Anda memiliki:

- Python 3.6 atau yang lebih baru diinstal
- AWS SDK untuk Python (Boto3) diinstal
- AWS kredensional yang dikonfigurasi dengan izin yang sesuai untuk Amazon Glacier, Amazon SNS, dan Amazon SQS
- Ruang disk yang cukup untuk menyimpan potongan arsip yang diunduh

Contoh: Mengunduh Arsip Menggunakan Pemrosesan Paralel dengan Python

Skrip Python berikut menunjukkan cara mengunduh arsip besar dari Amazon Glacier menggunakan pemrosesan paralel:

```
import boto3
import time
import json
import jmespath
import re
import concurrent.futures
import os

output_file_path = "output_directory_path"
vault_name = "vault_name"

chunk_size = 1000000000 #1gb - size of chunks for parallel download.
notify_queue_name = 'GlacierJobCompleteNotifyQueue' # SQS queue for Glacier recall
 notification
chunk_download_queue_name='GlacierChunkReadyNotifyQueue' # SQS queue for chunks
sns_topic_name = 'GlacierRecallJobCompleted' # the SNS topic to be notified when
 Glacier archive is restored.
chunk_queue_visibility_timeout = 7200 # 2 hours - this may need to be adjusted.
region = 'us-east-1'
archive_id = "archive_id_to_restore"
retrieve_archive = True # set to false if you do not want to restore from Glacier -
 useful for restarting or parallel processing of the chunk queue.
workers = 12 # the number of parallel worker threads for downloading chunks.

def setup_queues_and_topic():
```

```
    sqs = boto3.client('sqs')
    sns = boto3.client('sns')

    # Create the SNS topic
    topic_response = sns.create_topic(
        Name=sns_topic_name
    )
    topic_arn = topic_response['TopicArn']
    print("Creating the SNS topic " + topic_arn)

    # Create the notification queue
    notify_queue_response = sqs.create_queue(
        QueueName=notify_queue_name,
        Attributes={
            'VisibilityTimeout': '300',  # 5 minutes
            'ReceiveMessageWaitTimeSeconds': '20'  # Enable long polling
        }
    )
    notify_queue_url = notify_queue_response['QueueUrl']
    print("Creating the archive-retrieval notification queue " + notify_queue_url)

    # Create the chunk download queue
    chunk_queue_response = sqs.create_queue(
        QueueName=chunk_download_queue_name,
        Attributes={
            'VisibilityTimeout': str(chunk_queue_visibility_timeout),  # 5 minutes
            'ReceiveMessageWaitTimeSeconds': '0'
        }
    )
    chunk_queue_url = chunk_queue_response['QueueUrl']

    print("Creating the chunk ready notification queue " + chunk_queue_url)


   # Get the ARN for the notification queue
    notify_queue_attributes = sqs.get_queue_attributes(
        QueueUrl=notify_queue_url,
        AttributeNames=['QueueArn']
    )
    notify_queue_arn = notify_queue_attributes['Attributes']['QueueArn']

    # Set up the SNS topic policy on the notification queue
    queue_policy = {
        "Version": "2012-10-17",
```

```
        "Statement": [{
            "Sid": "allow-sns-messages",
            "Effect": "Allow",
            "Principal": {"AWS": "*"},
            "Action": "SQS:SendMessage",
            "Resource": notify_queue_arn,
            "Condition": {
                "ArnEquals": {
                    "aws:SourceArn": topic_arn
                }
            }
        }]
    }

    # Set the queue policy
    sqs.set_queue_attributes(
        QueueUrl=notify_queue_url,
        Attributes={
            'Policy': json.dumps(queue_policy)
        }
    )

    # Subscribe the notification queue to the SNS topic
    sns.subscribe(
        TopicArn=topic_arn,
        Protocol='sqs',
        Endpoint=notify_queue_arn
    )

    return {
        'topic_arn': topic_arn,
        'notify_queue_url': notify_queue_url,
        'chunk_queue_url': chunk_queue_url
    }


def split_and_send_chunks(archive_size, job_id,chunk_queue_url):
    ranges = []
    current = 0
    chunk_number = 0

    while current < archive_size:
        chunk_number += 1
        next_range = min(current + chunk_size - 1, archive_size - 1)
```

```
        ranges.append((current, next_range, chunk_number))
        current = next_range + 1

    # Send messages to SQS queue
    for start, end, chunk_number in ranges:
        body = {"start": start, "end": end, "job_id": job_id, "chunk_number":
 chunk_number}
        body = json.dumps(body)
        print("Sending SQS message for range:" + str(body))
        response = sqs.send_message(
            QueueUrl=chunk_queue_url,
            MessageBody=str(body)
        )

def GetJobOutputChunks(job_id, byterange, chunk_number):
    glacier = boto3.client('glacier')
    response = glacier.get_job_output(
        vaultName=vault_name,
        jobId=job_id,
        range=byterange,

    )

    with open(os.path.join(output_file_path,str(chunk_number)+".chunk"), 'wb') as
 output_file:
        output_file.write(response['body'].read())

    return response

def ReceiveArchiveReadyMessages(notify_queue_url,chunk_queue_url):

    response = sqs.receive_message(
        QueueUrl=notify_queue_url,
        AttributeNames=['All'],
        MaxNumberOfMessages=1,
        WaitTimeSeconds=20,
        MessageAttributeNames=['Message']
    )
    print("Polling archive retrieval job ready queue...")
    # Checking that there is a Messages key before proceeding. No 'Messages' key likely
 means the queue is empty

    if 'Messages' in response:
        print("Received a message from the archive retrieval job queue")
```

```
        jsonresponse = response
        # Loading the string into JSON and checking that ArchiveSizeInBytes key is
 present before continuing.
        jsonresponse=json.loads(jsonresponse['Messages'][0]['Body'])
        jsonresponse=json.loads(jsonresponse['Message'])
        if 'ArchiveSizeInBytes' in jsonresponse:
            receipt_handle = response['Messages'][0]['ReceiptHandle']
            if jsonresponse['ArchiveSizeInBytes']:
                archive_size = jsonresponse['ArchiveSizeInBytes']

                print(f'Received message: {response}')
                if archive_size > chunk_size:
                    split_and_send_chunks(archive_size,
 jsonresponse['JobId'],chunk_queue_url)

                    sqs.delete_message(
                    QueueUrl=notify_queue_url,
                    ReceiptHandle=receipt_handle)

            else:
                print("No ArchiveSizeInBytes value found in message")
                print(response)

    else:
        print('No messages available in the queue at this time.')

    time.sleep(1)

def ReceiveArchiveChunkMessages(chunk_queue_url):
    response = sqs.receive_message(
        QueueUrl=chunk_queue_url,
        AttributeNames=['All'],
        MaxNumberOfMessages=1,
        WaitTimeSeconds=0,
        MessageAttributeNames=['Message']
    )
    print("Polling archive chunk queue...")
    print(response)
    # Checking that there is a Messages key before proceeding. No 'Messages' key likely
 means the queue is empty
    if 'Messages' in response:
        jsonresponse = response
        # Loading the string into JSON and checking that ArchiveSizeInBytes key is
 present before continuing.
```

```
        jsonresponse=json.loads(jsonresponse['Messages'][0]['Body'])
        if 'job_id' in jsonresponse: #checking that there is a job id before continuing
            job_id = jsonresponse['job_id']
            byterange = "bytes="+str(jsonresponse['start']) + '-' +
 str(jsonresponse['end'])
            chunk_number = jsonresponse['chunk_number']
            receipt_handle = response['Messages'][0]['ReceiptHandle']
            if jsonresponse['job_id']:
                print(f'Received message: {response}')
                GetJobOutputChunks(job_id,byterange,chunk_number)
                sqs.delete_message(
                QueueUrl=chunk_queue_url,
                ReceiptHandle=receipt_handle)
    else:
        print('No messages available in the chunk queue at this time.')

def initiate_archive_retrieval(archive_id, topic_arn):
    glacier = boto3.client('glacier')

    job_parameters = {
        "Type": "archive-retrieval",
        "ArchiveId": archive_id,
        "Description": "Archive retrieval job",
        "SNSTopic": topic_arn,
        "Tier": "Bulk"  # You can change this to "Standard" or "Expedited" based on
 your needs
    }

    try:
        response = glacier.initiate_job(
            vaultName=vault_name,
            jobParameters=job_parameters
        )

        print("Archive retrieval job initiated:")
        print(f"Job ID: {response['jobId']}")
        print(f"Job parameters: {job_parameters}")
        print(f"Complete response: {json.dumps(response, indent=2)}")

        return response['jobId']

    except Exception as e:
        print(f"Error initiating archive retrieval job: {str(e)}")
        raise
```

```
def run_async_tasks(chunk_queue_url, workers):
    max_workers = workers  # Set the desired maximum number of concurrent tasks
    with concurrent.futures.ThreadPoolExecutor(max_workers=max_workers) as executor:
        for _ in range(max_workers):
            executor.submit(ReceiveArchiveChunkMessages, chunk_queue_url)

# One time setup of the necessary queues and topics.
queue_and_topic_atts = setup_queues_and_topic()

topic_arn = queue_and_topic_atts['topic_arn']
notify_queue_url = queue_and_topic_atts['notify_queue_url']
chunk_queue_url = queue_and_topic_atts['chunk_queue_url']

if retrieve_archive:
    print("Retrieving the defined archive... The topic arn we will notify when
 recalling the archive is: "+topic_arn)
    job_id = initiate_archive_retrieval(archive_id, topic_arn)
else:
    print("Retrieve archive is false, polling queues and downloading only.")

while True:
   ReceiveArchiveReadyMessages(notify_queue_url,chunk_queue_url)
   run_async_tasks(chunk_queue_url,workers)
```

Menggunakan Script

Untuk menggunakan skrip ini, ikuti langkah-langkah berikut:

1. Ganti nilai placeholder dalam skrip dengan informasi spesifik Anda:
   - *output_file_path*: Direktori tempat file chunk akan disimpan
   - *vault_name*: Nama lemari besi S3 Glacier Anda
   - *notify_queue_name*: Nama untuk antrian pemberitahuan pekerjaan
   - *chunk_download_queue_name*: Nama untuk antrian unduhan potongan
   - *sns_topic_name*: Nama untuk topik SNS
   - *region*: AWS wilayah tempat lemari besi Anda berada
   - *archive_id*: ID arsip untuk mengambil
2. Jalankan skrip .

```
python download_large_archive.py
```

3. Setelah semua potongan diunduh, Anda dapat menggabungkannya menjadi satu file menggunakan perintah seperti:

```
cat /path/to/chunks/*.chunk > complete_archive.file
```

Pertimbangan Penting

Saat menggunakan skrip ini, ingatlah hal berikut:

- Pengambilan arsip dari S3 Glacier dapat memakan waktu beberapa jam untuk diselesaikan, tergantung pada tingkat pengambilan yang dipilih.
- Skrip berjalan tanpa batas waktu, terus melakukan polling antrian. Anda mungkin ingin menambahkan kondisi penghentian berdasarkan persyaratan spesifik Anda.
- Pastikan Anda memiliki ruang disk yang cukup untuk menyimpan semua potongan arsip Anda.
- Jika skrip terputus, Anda dapat memulai ulang dengan `retrieve_archive=False` untuk terus mengunduh potongan tanpa memulai pekerjaan pengambilan baru.
- Sesuaikan *`chunk_size`* dan *`workers`* parameter berdasarkan bandwidth jaringan dan sumber daya sistem Anda.
- AWS Biaya standar berlaku untuk pengambilan Amazon S3, Amazon SNS, dan penggunaan Amazon SQS.

Mengunduh Arsip dengan Menggunakan REST API

Untuk mengunduh arsip menggunakan REST API

Mengunduh arsip adalah proses dua langkah.

1. Mulai tugas dari tipe `archive-retrieval`. Untuk informasi selengkapnya, lihat Memulai Tugas (POST jobs).
2. Setelah pekerjaan selesai, unduh data arsip. Lihat informasi yang lebih lengkap di Mendapatkan Output Tugas (GET output).

Mengunduh Arsip di Amazon Glacier Menggunakan AWS CLI

Anda dapat mengunduh arsip di Amazon Glacier (Amazon Glacier) menggunakan (). AWS Command Line Interface AWS CLI

Topik

- (Prasyarat) Menyiapkan AWS CLI
- Contoh: Unduh Arsip Menggunakan AWS CLI

(Prasyarat) Menyiapkan AWS CLI

1. Unduh dan konfigurasikan AWS CLI. Untuk melakukannya, lihat topik berikut di Panduan Pengguna AWS Command Line Interface :

   Memasang AWS Command Line Interface

   Mengkonfigurasi AWS Command Line Interface

2. Verifikasi AWS CLI pengaturan Anda dengan memasukkan perintah berikut pada prompt perintah. Perintah ini tidak memberikan kredensial secara eksplisit, sehingga kredensial profil default digunakan.

   - Coba gunakan perintah bantuan.

     ```
     aws help
     ```

   - Untuk mendapatkan daftar brankas Amazon Glacier pada akun yang dikonfigurasi, gunakan perintah. `list-vaults` Ganti *123456789012* dengan Akun AWS ID Anda.

     ```
     aws glacier list-vaults --account-id 123456789012
     ```

   - Untuk melihat data konfigurasi saat ini untuk AWS CLI, gunakan `aws configure list` perintah.

     ```
     aws configure list
     ```

Contoh: Unduh Arsip Menggunakan AWS CLI

> **Note**
>
> Untuk mengunduh arsip Anda, Anda harus mengetahui id arsip Anda. Langkah 1-4 akan mengambil id arsip Anda. Jika Anda sudah mengetahui id arsip yang ingin Anda unduh, lewati ke langkah 5.

1. Gunakan `initiate-job` perintah untuk memulai pekerjaan pengambilan inventaris. Laporan inventaris akan mencantumkan id arsip Anda.

   ```
   aws glacier initiate-job --vault-name awsexamplevault --account-id 111122223333 --
   job-parameters="{\"Type\":\"inventory-retrieval\"}"
   ```

   Keluaran yang diharapkan

   ```
   {
       "location": "/111122223333/vaults/awsexamplevault/jobs/*** jobid ***",
       "jobId": "*** jobid ***"
   }
   ```

2. Gunakan `describe-job` perintah untuk memeriksa status perintah pekerjaan sebelumnya.

   ```
   aws glacier describe-job --vault-name awsexamplevault --account-id 111122223333 --
   job-id *** jobid ***
   ```

   Keluaran yang diharapkan

   ```
   {
       "InventoryRetrievalParameters": {
           "Format": "JSON"
       },
       "VaultARN": "*** vault arn ***",
       "Completed": false,
       "JobId": "*** jobid ***",
       "Action": "InventoryRetrieval",
       "CreationDate": "*** job creation date ***",
   ```

```
    "StatusCode": "InProgress"
}
```

3. Tunggu hingga tugas selesai.

   Anda harus menunggu hingga output tugas siap diunduh. Jika Anda menyetel konfigurasi notifikasi di vault atau menetapkan topik Simple Notification Service Amazon (Amazon SNS) saat memulai pekerjaan, Amazon Glacier akan mengirimkan pesan ke topik tersebut setelah menyelesaikan pekerjaan.

   Anda dapat menetapkan konfigurasi notifikasi untuk peristiwa tertentu di vault. Untuk informasi selengkapnya, lihat Mengkonfigurasi Pemberitahuan Vault di Amazon Glacier. Amazon Glacier mengirim pesan ke topik SNS yang ditentukan kapan saja peristiwa tertentu terjadi.

4. Setelah selesai, gunakan perintah `get-job-output` untuk mengunduh tugas pengambilan ke file `output.json`. File ini akan berisi id arsip Anda.

   ```
   aws glacier get-job-output --vault-name awsexamplevault --account-id 111122223333
    --job-id *** jobid *** output.json
   ```

   Perintah ini menghasilkan file dengan bidang berikut.

   ```
   {
   "VaultARN":"arn:aws:glacier:region:111122223333:vaults/awsexamplevault",
   "InventoryDate":"*** job completion date ***",
   "ArchiveList":[
   {"ArchiveId":"*** archiveid ***",
   "ArchiveDescription":*** archive description (if set) ***,
   "CreationDate":"*** archive creation date ***",
   "Size":"*** archive size (in bytes) ***",
   "SHA256TreeHash":"*** archive hash ***"
   }
   {"ArchiveId":
   ...
   ]}
   ```

5. Gunakan `initiate-job` perintah untuk memulai proses pengambilan setiap arsip dari brankas. Anda harus menentukan parameter pekerjaan `archive-retrieval` seperti yang terlihat di bawah ini.

```
aws glacier initiate-job --vault-name awsexamplevault --account-id 111122223333
 --job-parameters="{\"Type\":\"archive-retrieval\",\"ArchiveId\":\"*** archiveId
 ***\"}"
```

6. Tunggu sampai `archive-retrieval` pekerjaan selesai. Gunakan `describe-job` perintah untuk memeriksa status perintah sebelumnya.

```
aws glacier describe-job --vault-name awsexamplevault --account-id 111122223333 --
job-id *** jobid ***
```

7. Ketika pekerjaan di atas selesai gunakan `get-job-output` perintah untuk men-download arsip Anda.

```
aws glacier get-job-output --vault-name awsexamplevault --account-id 111122223333
 --job-id *** jobid *** output_file_name
```

Menghapus Arsip di Amazon Glacier

Anda tidak dapat menghapus arsip menggunakan konsol manajemen Amazon Glacier (Amazon Glacier). Untuk menghapus arsip, Anda harus menggunakan AWS Command Line Interface (CLI) atau menulis kode untuk membuat permintaan penghapusan menggunakan REST API secara langsung atau pustaka pembungkus AWS SDK for Java dan .NET. Topik berikut menjelaskan cara menggunakan library wrapper AWS SDK for Java dan .NET, REST API, dan. AWS CLI

Topik

- Menghapus Arsip di Amazon Glacier Menggunakan AWS SDK for Java
- Menghapus Arsip di Amazon Glacier Menggunakan AWS SDK for .NET
- Menghapus Arsip Amazon Glacier Menggunakan REST API
- Menghapus Arsip di Amazon Glacier Menggunakan AWS Command Line Interface

Anda dapat menghapus satu arsip sekaligus dari vault. Untuk menghapus arsip, Anda harus memberikan ID arsip dalam permintaan hapus Anda. Anda bisa mendapatkan ID arsip dengan mengunduh inventaris vault untuk vault yang berisi arsip. Untuk informasi selengkapnya tentang mengunduh inventaris vault, lihat Mengunduh Inventaris Vault di Amazon Glacier.

Setelah menghapus arsip, Anda mungkin masih dapat membuat permintaan yang sukses untuk memulai tugas guna mengambil arsip yang dihapus, tetapi tugas pengambilan arsip akan gagal.

Pengambilan arsip yang sedang berlangsung untuk ID arsip ketika Anda menghapus arsip mungkin atau mungkin tidak berhasil menurut skenario berikut:

- Jika tugas pengambilan arsip secara aktif menyiapkan data untuk diunduh saat Amazon Glacier menerima permintaan hapus arsip, maka operasi pengambilan arsip mungkin gagal.
- Jika pekerjaan pengambilan arsip telah berhasil menyiapkan arsip untuk diunduh saat Amazon Glacier menerima permintaan arsip hapus, maka Anda akan dapat mengunduh hasilnya.

Untuk informasi selengkapnya tentang pengambilan arsip, lihat Mengunduh Arsip di Amazon Glacier.

Operasi ini bersifat idempoten. Menghapus arsip yang sudah dihapus tidak mengakibatkan kesalahan.

Setelah Anda menghapus arsip, jika Anda segera mengunduh inventaris vault, mungkin akan menyertakan arsip yang dihapus dalam daftar karena Amazon Glacier menyiapkan inventaris vault hanya sekitar sekali sehari.

> **Note**
>
> Untuk penghapusan otomatis arsip vault, lihat Penghapusan otomatis arsip vault di Amazon S3 Glacier.

Menghapus Arsip di Amazon Glacier Menggunakan AWS SDK for Java

Berikut ini adalah langkah-langkah untuk menghapus arsip menggunakan API AWS SDK for Java tingkat rendah.

1. Buat instans dari kelas `AmazonGlacierClient` (klien).

   Anda perlu menentukan AWS Wilayah tempat arsip yang ingin Anda hapus disimpan. Semua operasi yang Anda lakukan menggunakan klien ini berlaku untuk AWS Wilayah tersebut.
2. Berikan informasi permintaan dengan membuat instans kelas `DeleteArchiveRequest`.

Anda harus memberikan ID arsip, nama vault, dan ID akun Anda. Jika Anda tidak memberikan ID akun, ID akun yang terkait dengan kredensial yang Anda berikan untuk menandatangani permintaan akan diambil. Untuk informasi selengkapnya, lihat Menggunakan AWS SDK for Java dengan Amazon Glacier.

3. Jalankan metode `deleteArchive` dengan menyediakan objek permintaan sebagai parameter.

Potongan kode Java berikut menggambarkan langkah-langkah sebelumnya.

```
AmazonGlacierClient client;

DeleteArchiveRequest request = new DeleteArchiveRequest()
    .withVaultName("*** provide a vault name ***")
    .withArchiveId("*** provide an archive ID ***");

client.deleteArchive(request);
```

> **Note**
>
> Untuk informasi tentang REST API yang mendasarinya, lihat Menghapus Arsip (DELETE archive).

Contoh: Menghapus Arsip Menggunakan AWS SDK for Java

Contoh kode Java berikut menggunakan AWS SDK for Java untuk menghapus arsip. Untuk step-by-step petunjuk tentang cara menjalankan contoh ini, lihatMenjalankan Contoh Java untuk Amazon Glacier Menggunakan Eclipse. Anda perlu memperbarui kode seperti yang ditunjukkan dengan nama vault dan ID arsip dari arsip yang ingin Anda hapus.

Example

```
import java.io.IOException;

import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.glacier.AmazonGlacierClient;
import com.amazonaws.services.glacier.model.DeleteArchiveRequest;
```

```
public class ArchiveDelete {

    public static String vaultName = "*** provide vault name ****";
    public static String archiveId = "*** provide archive ID***";
    public static AmazonGlacierClient client;

    public static void main(String[] args) throws IOException {

     ProfileCredentialsProvider credentials = new ProfileCredentialsProvider();

        client = new AmazonGlacierClient(credentials);
        client.setEndpoint("https://glacier.us-east-1.amazonaws.com/");

        try {

            // Delete the archive.
            client.deleteArchive(new DeleteArchiveRequest()
                .withVaultName(vaultName)
                .withArchiveId(archiveId));

            System.out.println("Deleted archive successfully.");

        } catch (Exception e) {
            System.err.println("Archive not deleted.");
            System.err.println(e);
        }
    }
}
```

Menghapus Arsip di Amazon Glacier Menggunakan AWS SDK for .NET

Baik tingkat tinggi dan tingkat rendah yang APIs disediakan oleh Amazon SDK untuk.NET menyediakan metode untuk menghapus arsip.

Topik

- Menghapus Arsip Menggunakan API Tingkat Tinggi AWS SDK for .NET
- Menghapus Arsip Menggunakan API Tingkat Rendah AWS SDK for .NET

Menghapus Arsip Menggunakan API Tingkat Tinggi AWS SDK for .NET

Kelas `ArchiveTransferManager` dari API tingkat tinggi menyediakan metode `DeleteArchive` yang dapat Anda gunakan untuk menghapus arsip.

Contoh: Menghapus Arsip Menggunakan API Tingkat Tinggi AWS SDK for .NET

Contoh kode C # berikut menggunakan API tingkat tinggi AWS SDK for .NET untuk menghapus arsip. Untuk step-by-step petunjuk tentang cara menjalankan contoh ini, lihatMenjalankan Contoh Kode. Anda perlu memperbarui kode seperti yang ditunjukkan dengan ID arsip dari arsip yang ingin Anda hapus.

Example

```
using System;
using Amazon.Glacier;
using Amazon.Glacier.Transfer;
using Amazon.Runtime;

namespace glacier.amazon.com.docsamples
{
  class ArchiveDeleteHighLevel
  {
    static string vaultName = "examplevault";
    static string archiveId = "*** Provide archive ID ***";

    public static void Main(string[] args)
    {
      try
      {
        var manager = new ArchiveTransferManager(Amazon.RegionEndpoint.USWest2);
        manager.DeleteArchive(vaultName, archiveId);
        Console.ReadKey();
      }
      catch (AmazonGlacierException e) { Console.WriteLine(e.Message); }
      catch (AmazonServiceException e) { Console.WriteLine(e.Message); }
      catch (Exception e) { Console.WriteLine(e.Message); }
      Console.WriteLine("To continue, press Enter");
      Console.ReadKey();
    }
  }
}
```

Menghapus Arsip Menggunakan API Tingkat Rendah AWS SDK for .NET

Berikut adalah langkah-langkah untuk menghapus menggunakan AWS SDK for .NET.

1. Buat instans dari kelas `AmazonGlacierClient` (klien).

   Anda perlu menentukan AWS Wilayah tempat arsip yang ingin Anda hapus disimpan. Semua operasi yang Anda lakukan menggunakan klien ini berlaku untuk AWS Wilayah tersebut.
2. Berikan informasi permintaan dengan membuat instans kelas `DeleteArchiveRequest`.

   Anda harus memberikan ID arsip, nama vault, dan ID akun Anda. Jika Anda tidak memberikan ID akun, ID akun yang terkait dengan kredensial yang Anda berikan untuk menandatangani permintaan akan diambil. Untuk informasi selengkapnya, lihat Menggunakan AWS SDKs dengan Amazon Glacier.
3. Jalankan metode `DeleteArchive` dengan menyediakan objek permintaan sebagai parameter.

Contoh: Menghapus Arsip Menggunakan API Tingkat Rendah AWS SDK for .NET

Contoh C# berikut menjelaskan langkah sebelumnya. Contoh menggunakan API tingkat rendah AWS SDK for .NET untuk menghapus arsip.

> **Note**
>
> Untuk informasi tentang REST API yang mendasarinya, lihat Menghapus Arsip (DELETE archive).

Untuk step-by-step petunjuk tentang cara menjalankan contoh ini, lihatMenjalankan Contoh Kode. Anda perlu memperbarui kode seperti yang ditunjukkan dengan ID arsip dari arsip yang ingin Anda hapus.

Example

```
using System;
using Amazon.Glacier;
using Amazon.Glacier.Model;
using Amazon.Runtime;

namespace glacier.amazon.com.docsamples
```

```
{
  class ArchiveDeleteLowLevel
  {
    static string vaultName = "examplevault";
    static string archiveId = "*** Provide archive ID ***";

    public static void Main(string[] args)
    {
      AmazonGlacierClient client;
      try
      {
        using (client = new AmazonGlacierClient(Amazon.RegionEndpoint.USWest2))
        {
          Console.WriteLine("Deleting the archive");
          DeleteAnArchive(client);
        }
        Console.WriteLine("Operations successful. To continue, press Enter");
        Console.ReadKey();
      }
      catch (AmazonGlacierException e) { Console.WriteLine(e.Message); }
      catch (AmazonServiceException e) { Console.WriteLine(e.Message); }
      catch (Exception e) { Console.WriteLine(e.Message); }
      Console.WriteLine("To continue, press Enter");
      Console.ReadKey();
    }

    static void DeleteAnArchive(AmazonGlacierClient client)
    {
      DeleteArchiveRequest request = new DeleteArchiveRequest()
      {
        VaultName = vaultName,
        ArchiveId = archiveId
      };
      DeleteArchiveResponse response = client.DeleteArchive(request);
    }
  }
}
```

Menghapus Arsip Amazon Glacier Menggunakan REST API

Anda dapat menggunakan API Hapus Arsip untuk menghapus arsip.

- Untuk informasi tentang API Hapus Arsip, lihat Menghapus Arsip (DELETE archive).

- Untuk informasi selengkapnya tentang menggunakan REST API, lihat Referensi API untuk Amazon Glacier.

Menghapus Arsip di Amazon Glacier Menggunakan AWS Command Line Interface

Anda dapat menghapus arsip di Amazon Glacier (Amazon Glacier) menggunakan (). AWS Command Line Interface AWS CLI

Topik

- (Prasyarat) Menyiapkan AWS CLI
- Contoh: Menghapus Arsip Menggunakan AWS CLI

(Prasyarat) Menyiapkan AWS CLI

1. Unduh dan konfigurasikan AWS CLI. Untuk melakukannya, lihat topik berikut di Panduan Pengguna AWS Command Line Interface :

   Instalasi AWS Command Line Interface

   Mengkonfigurasi AWS Command Line Interface

2. Verifikasi AWS CLI pengaturan Anda dengan memasukkan perintah berikut pada prompt perintah. Perintah ini tidak memberikan kredensial secara eksplisit, sehingga kredensial profil default digunakan.

   - Coba gunakan perintah bantuan.

     ```
     aws help
     ```

   - Untuk mendapatkan daftar brankas Amazon Glacier pada akun yang dikonfigurasi, gunakan perintah. `list-vaults` Ganti *123456789012* dengan Akun AWS ID Anda.

     ```
     aws glacier list-vaults --account-id 123456789012
     ```

   - Untuk melihat data konfigurasi saat ini untuk AWS CLI, gunakan `aws configure list` perintah.

```
aws configure list
```

Contoh: Menghapus Arsip Menggunakan AWS CLI

1. Gunakan perintah initiate-job untuk memulai tugas pengambilan inventaris.

```
aws glacier initiate-job --vault-name awsexamplevault --account-id 111122223333 --
job-parameters="{\"Type\":\"inventory-retrieval\"}"
```

Output yang diharapkan:

```
{
    "location": "/111122223333/vaults/awsexamplevault/jobs/*** jobid ***",
    "jobId": "*** jobid ***"
}
```

2. Gunakan perintah describe-job untuk memeriksa status tugas pengambilan sebelumnya.

```
aws glacier describe-job --vault-name awsexamplevault --account-id 111122223333 --
job-id *** jobid ***
```

Output yang diharapkan:

```
{
    "InventoryRetrievalParameters": {
        "Format": "JSON"
    },
    "VaultARN": "*** vault arn ***",
    "Completed": false,
    "JobId": "*** jobid ***",
    "Action": "InventoryRetrieval",
    "CreationDate": "*** job creation date ***",
    "StatusCode": "InProgress"
}
```

3. Tunggu hingga tugas selesai.

   Anda harus menunggu hingga output tugas siap diunduh. Jika Anda menyetel konfigurasi notifikasi di vault atau menetapkan topik Simple Notification Service Amazon (Amazon SNS) saat memulai pekerjaan, Amazon Glacier akan mengirimkan pesan ke topik tersebut setelah menyelesaikan pekerjaan.

   Anda dapat menetapkan konfigurasi notifikasi untuk peristiwa tertentu di vault. Untuk informasi selengkapnya, lihat Mengkonfigurasi Pemberitahuan Vault di Amazon Glacier. Amazon Glacier mengirim pesan ke topik SNS yang ditentukan kapan saja peristiwa tertentu terjadi.

4. Setelah selesai, gunakan perintah get-job-output untuk mengunduh tugas pengambilan ke file `output.json`.

```
aws glacier get-job-output --vault-name awsexamplevault --account-id 111122223333
 --job-id *** jobid *** output.json
```

   Perintah ini menghasilkan file dengan bidang berikut.

```
{
"VaultARN":"arn:aws:glacier:region:111122223333:vaults/awsexamplevault",
"InventoryDate":"*** job completion date ***",
"ArchiveList":[
{"ArchiveId":"*** archiveid ***",
"ArchiveDescription":*** archive description (if set) ***,
"CreationDate":"*** archive creation date ***",
"Size":"*** archive size (in bytes) ***",
"SHA256TreeHash":"*** archive hash ***"
}
{"ArchiveId":
...
]}
```

5. Gunakan perintah `delete-archive` untuk menghapus setiap arsip dari vault sampai tidak ada yang tersisa.

```
aws glacier delete-archive --vault-name awsexamplevault --account-id 111122223333
 --archive-id *** archiveid ***
```

Menggunakan AWS SDKs dengan Amazon Glacier

AWS menyediakan SDKs bagi Anda untuk mengembangkan aplikasi untuk Amazon Glacier. Pustaka SDK membungkus API Amazon Glacier yang mendasarinya, menyederhanakan tugas pemrograman Anda. Misalnya, untuk setiap permintaan yang dikirim ke Amazon Glacier, Anda harus menyertakan tanda tangan untuk mengautentikasi permintaan Anda. Saat Anda menggunakan pustaka SDK, Anda hanya perlu memberikan kredensyal AWS keamanan dalam kode Anda dan pustaka menghitung tanda tangan yang diperlukan dan menyertakannya dalam permintaan yang dikirim ke Amazon Glacier. Ini AWS SDKs menyediakan pustaka yang memetakan ke REST API yang mendasarinya dan menyediakan objek yang dapat Anda gunakan untuk membuat permintaan dan respons proses dengan mudah.

Topik

- AWS Perpustakaan SDK untuk Java dan .NET
- Menggunakan Amazon Glacier dengan SDK AWS
- Menggunakan AWS SDK for Java dengan Amazon Glacier
- Menggunakan AWS SDK for .NET dengan Amazon Glacier

The AWS Command Line Interface (AWS CLI) adalah alat terpadu untuk mengelola Anda Layanan AWS, termasuk Amazon Glacier. Untuk informasi tentang mengunduh AWS CLI, lihat AWS Command Line Interface. Untuk daftar perintah Amazon Glacier CLI, lihat Referensi Perintah.AWS CLI

AWS Perpustakaan SDK untuk Java dan .NET

The AWS SDKs for Java dan .NET menawarkan pustaka pembungkus tingkat tinggi dan tingkat rendah.

Anda dapat menemukan contoh bekerja dengan Amazon Glacier dengan menggunakan dan AWS SDK for Java seluruh panduan pengembang AWS SDK for .NET ini.

Apa itu API Tingkat Rendah?

Pustaka pembungkus tingkat rendah memetakan erat REST API (Referensi API untuk Amazon Glacier) yang mendasari yang didukung oleh Amazon Glacier. Untuk setiap operasi Amazon Glacier

REST, API tingkat rendah menyediakan metode yang sesuai, objek permintaan agar Anda dapat memberikan informasi permintaan, dan objek respons bagi Anda untuk memproses respons Amazon Glacier. Pustaka pembungkus tingkat rendah adalah implementasi paling lengkap dari operasi Amazon Glacier yang mendasarinya.

Untuk informasi tentang pustaka SDK ini, lihat Menggunakan AWS SDK for Java dengan Amazon Glacier dan Menggunakan AWS SDK for .NET dengan Amazon Glacier.

Apa itu API Tingkat Tinggi?

Untuk lebih menyederhanakan pengembangan aplikasi, pustaka ini menawarkan abstraksi tingkat yang lebih tinggi untuk beberapa operasi. Contoh:

- Mengunggah arsip—Untuk mengunggah arsip menggunakan API tingkat rendah selain nama file dan nama vault tempat Anda ingin menyimpan arsip, Anda perlu memberikan checksum (hash pohon SHA-256) dari muatan. Namun, API tingkat tinggi menghitung checksum untuk Anda.
- Mengunduh arsip atau inventaris vault—Untuk mengunduh arsip menggunakan API tingkat rendah, Anda terlebih dulu memulai tugas, menunggu tugas selesai, lalu mendapatkan output tugas. Anda perlu menulis kode tambahan untuk menyiapkan topik Amazon Simple Notification Service (Amazon SNS) untuk Amazon Glacier untuk memberi tahu Anda saat pekerjaan selesai. Anda juga memerlukan beberapa mekanisme polling untuk memeriksa apakah pesan penyelesaian tugas sudah di-post ke topik. API tingkat tinggi menyediakan metode untuk mengunduh arsip yang menangani semua langkah ini. Anda hanya menentukan ID arsip dan jalur folder tempat Anda ingin menyimpan data yang diunduh.

Untuk informasi tentang pustaka SDK ini, lihat Menggunakan AWS SDK for Java dengan Amazon Glacier dan Menggunakan AWS SDK for .NET dengan Amazon Glacier.

Kapan Menggunakan API Tingkat Tinggi dan Tingkat Rendah

Secara umum, jika API tingkat tinggi menyediakan metode yang Anda perlukan untuk melakukan operasi, Anda harus menggunakan API tingkat tinggi karena kesederhanaan yang disediakan. Namun, jika API tingkat tinggi tidak menawarkan fungsionalitas, Anda dapat menggunakan API tingkat rendah. Selain itu, API tingkat rendah memungkinkan kontrol granular operasi seperti logika coba lagi jika terjadi kegagalan. Misalnya, ketika mengunggah arsip, API tingkat tinggi menggunakan ukuran file untuk menentukan apakah akan mengunggah arsip dalam satu operasi atau menggunakan API unggahan multipart. API juga memiliki logika coba lagi bawaan jika unggahan

gagal. Namun, aplikasi Anda mungkin memerlukan kontrol granular atas keputusan ini, dalam hal ini Anda dapat menggunakan API tingkat rendah.

Menggunakan Amazon Glacier dengan SDK AWS

AWS kit pengembangan perangkat lunak (SDKs) tersedia untuk banyak bahasa pemrograman populer. Setiap SDK menyediakan API, contoh kode, dan dokumentasi yang memudahkan developer untuk membangun aplikasi dalam bahasa pilihan mereka.

Dokumentasi SDK	Contoh kode
AWS SDK for C++	AWS SDK for C++ contoh kode
AWS CLI	AWS CLI contoh kode
AWS SDK for Go	AWS SDK for Go contoh kode
AWS SDK for Java	AWS SDK for Java contoh kode
AWS SDK for JavaScript	AWS SDK for JavaScript contoh kode
AWS SDK for Kotlin	AWS SDK for Kotlin contoh kode
AWS SDK for .NET	AWS SDK for .NET contoh kode
AWS SDK for PHP	AWS SDK for PHP contoh kode
AWS Tools for PowerShell	AWS Tools for PowerShell contoh kode
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) contoh kode
AWS SDK for Ruby	AWS SDK for Ruby contoh kode
AWS SDK for Rust	AWS SDK for Rust contoh kode
AWS SDK for SAP ABAP	AWS SDK for SAP ABAP contoh kode
AWS SDK for Swift	AWS SDK for Swift contoh kode

Untuk contoh khusus untuk Amazon Glacier, lihat. Contoh kode untuk Amazon Glacier menggunakan AWS SDKs

> **Ketersediaan contoh**
>
> Tidak dapat menemukan apa yang Anda butuhkan? Minta contoh kode menggunakan tautan Berikan umpan balik di bagian bawah halaman ini.

Menggunakan AWS SDK for Java dengan Amazon Glacier

AWS SDK for Java Ini menyediakan tingkat tinggi dan tingkat rendah APIs untuk Amazon Glacier (Amazon Glacier) seperti yang dijelaskan dalam. Menggunakan AWS SDKs dengan Amazon Glacier Untuk informasi tentang mengunduh AWS SDK for Java, lihat Amazon SDK for Java.

> **Note**
>
> AWS SDK for Java Ini menyediakan klien yang aman untuk mengakses Amazon Glacier. Sebagai praktik terbaik, aplikasi Anda harus membuat satu klien dan menggunakan kembali klien di antara thread.

Topik

- Menggunakan API Tingkat Rendah
- Menggunakan API Tingkat Tinggi
- Menjalankan Contoh Java untuk Amazon Glacier Menggunakan Eclipse
- Mengatur Titik Akhir

Menggunakan API Tingkat Rendah

`AmazonGlacierClient`Kelas tingkat rendah menyediakan semua metode yang memetakan ke operasi REST yang mendasari Amazon Referensi API untuk Amazon Glacier Glacier (). Saat memanggil salah satu metode ini, Anda harus membuat objek permintaan yang sesuai dan memberikan objek respons di mana metode tersebut dapat mengembalikan respons Amazon Glacier ke operasi.

Misalnya, kelas `AmazonGlacierClient` menyediakan metode `createVault` untuk membuat vault. Metode ini memetakan ke operasi Buat REST Vault yang mendasarinya (lihat Membuat Vault (PUT vault)). Untuk menggunakan metode ini, Anda harus membuat instance `CreateVaultResult` objek yang menerima respons Amazon Glacier seperti yang ditunjukkan pada cuplikan kode Java berikut:

```
AmazonGlacierClient client = new AmazonGlacierClient(credentials);
client.setEndpoint("https://glacier.us-west-2.amazonaws.com/");

CreateVaultRequest request = new CreateVaultRequest()
    .withAccountId("-")
    .withVaultName(vaultName);
CreateVaultResult result = client.createVault(createVaultRequest);
```

Semua sampel tingkat rendah dalam panduan menggunakan pola ini.

> **Note**
>
> Segmen kode sebelumnya menentukan `AccountID` ketika membuat permintaan. Namun, saat menggunakan AWS SDK for Java, permintaan `AccountId` dalam adalah opsional, dan oleh karena itu semua contoh tingkat rendah dalam panduan ini tidak menetapkan nilai ini. `AccountId`Ini adalah Akun AWS ID. Nilai ini harus cocok dengan Akun AWS ID yang terkait dengan kredensional yang digunakan untuk menandatangani permintaan. Anda dapat menentukan Akun AWS ID atau opsional '-', dalam hal ini Amazon Glacier Akun AWS menggunakan ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menentukan ID Akun, jangan sertakan tanda hubung di dalamnya. Saat menggunakan AWS SDK for Java, jika Anda tidak memberikan ID akun, pustaka akan menyetel ID akun menjadi '-'.

Menggunakan API Tingkat Tinggi

Untuk lebih menyederhanakan pengembangan aplikasi Anda, AWS SDK for Java menyediakan `ArchiveTransferManager` kelas yang mengimplementasikan abstraksi tingkat yang lebih tinggi untuk beberapa metode di API tingkat rendah. Ini menyediakan metode yang berguna, seperti metode `upload` dan `download`, untuk operasi arsip.

Sebagai contoh, potongan kode Java berikut menggunakan metode tingkat tinggi `upload` untuk mengunggah arsip.

```
String vaultName = "examplevault";
String archiveToUpload = "c:/folder/exampleArchive.zip";

ArchiveTransferManager atm = new ArchiveTransferManager(client, credentials);
String archiveId = atm.upload(vaultName, "Tax 2012 documents", new
 File(archiveToUpload)).getArchiveId();
```

Perhatikan bahwa operasi apa pun yang Anda lakukan berlaku untuk AWS Wilayah yang Anda tentukan saat membuat `ArchiveTransferManager` objek. Jika Anda tidak menentukan AWS Region apapun, AWS SDK for Java set `us-east-1` sebagai AWS Region default.

Semua contoh tingkat tinggi dalam panduan ini menggunakan pola ini.

> **Note**
>
> Kelas `ArchiveTransferManager` tingkat tinggi dapat dibangun dengan instans `AmazonGlacierClient` atau instans `AWSCredentials`.

Menjalankan Contoh Java untuk Amazon Glacier Menggunakan Eclipse

Cara termudah untuk memulai dengan contoh kode Java adalah dengan menginstal AWS Toolkit for Eclipse terbaru. Untuk informasi tentang menginstal atau memperbarui ke kit alat terbaru, buka http://aws.amazon.com/eclipse. Tugas-tugas berikut ini memandu Anda dalam pembuatan dan pengujian contoh kode Java yang disediakan dalam bagian ini.

Proses Umum Membuat Contoh Kode Java

1	Buat profil kredensial default untuk kredensional Anda seperti yang dijelaskan dalam AWS SDK for Java topik Menyediakan AWSAWS Kredensial di Amazon SDK for Java.
2	Buat proyek AWS Java baru di Eclipse. Proyek ini sudah dikonfigurasi sebelumnya dengan AWS SDK for Java.
3	Salin kode dari bagian yang sedang Anda baca ke proyek Anda.

4	Perbarui kode dengan menyediakan data yang diperlukan. Misalnya, jika mengunggah file, berikan alur file dan nama bucket.
5	Jalankan kode. Verifikasi bahwa objek dibuat menggunakan Konsol Manajemen AWS. Untuk informasi lebih lanjut tentang Konsol Manajemen AWS, kunjungi http://aws.amazon.com/console/.

Mengatur Titik Akhir

Secara default, AWS SDK for Java menggunakan titik akhir`https://glacier.us-east-1.amazonaws.com`. Anda dapat mengatur titik akhir secara eksplisit seperti ditunjukkan dalam potongan kode Java berikut.

Potongan berikut menunjukkan cara mengatur titik akhir ke Wilayah US West (Oregon) (`us-west-2`) di API tingkat rendah.

Example

```
client = new AmazonGlacierClient(credentials);
client.setEndpoint("glacier.us-west-2.amazonaws.com");
```

Potongan berikut menunjukkan cara mengatur titik akhir ke Wilayah US West (Oregon) di API tingkat tinggi.

```
glacierClient = new AmazonGlacierClient(credentials);
sqsClient = new AmazonSQSClient(credentials);
snsClient = new AmazonSNSClient(credentials);

glacierClient.setEndpoint("glacier.us-west-2.amazonaws.com");
sqsClient.setEndpoint("sqs.us-west-2.amazonaws.com");
snsClient.setEndpoint("sns.us-west-2.amazonaws.com");

ArchiveTransferManager atm = new ArchiveTransferManager(glacierClient, sqsClient,
 snsClient);
```

Untuk daftar AWS Wilayah dan titik akhir yang didukung, lihatMengakses Amazon Glacier.

Menggunakan AWS SDK for .NET dengan Amazon Glacier

AWS SDK for .NET API tersedia di`AWSSDK.dll`. Untuk informasi tentang mengunduh AWS SDK for .NET, buka Perpustakaan Kode Sampel. Seperti dijelaskan dalamMenggunakan AWS SDKs dengan Amazon Glacier, AWS SDK for .NET menyediakan tingkat tinggi dan tingkat rendah APIs.

> **Note**
>
> API tingkat rendah dan API tingkat tinggi menyediakan klien yang aman untuk mengakses Amazon Glacier. Sebagai praktik terbaik, aplikasi Anda harus membuat satu klien dan menggunakan kembali klien di antara thread.

Topik

- Menggunakan API Tingkat Rendah
- Menggunakan API Tingkat Tinggi
- Menjalankan Contoh Kode
- Mengatur Titik Akhir

Menggunakan API Tingkat Rendah

`AmazonGlacierClient`Kelas tingkat rendah menyediakan semua metode yang memetakan ke operasi REST yang mendasari Amazon Glacier (Amazon Glacier) (). Referensi API untuk Amazon Glacier Saat memanggil salah satu metode ini, Anda harus membuat objek permintaan yang sesuai dan memberikan objek respons di mana metode tersebut dapat mengembalikan respons Amazon Glacier ke operasi.

Misalnya, kelas `AmazonGlacierClient` menyediakan metode `CreateVault` untuk membuat vault. Metode ini memetakan ke operasi Buat REST Vault yang mendasarinya (lihat Membuat Vault (PUT vault)). Untuk menggunakan metode ini, Anda harus membuat instance `CreateVaultRequest` dan `CreateVaultResponse` kelas untuk memberikan informasi permintaan dan menerima respons Amazon Glacier seperti yang ditunjukkan pada cuplikan kode C # berikut:

```
AmazonGlacierClient client;
```

```
client = new AmazonGlacierClient(Amazon.RegionEndpoint.USEast1);

CreateVaultRequest request = new CreateVaultRequest()
{
  AccountId = "-",
  VaultName = "*** Provide vault name ***"
};

CreateVaultResponse response = client.CreateVault(request);
```

Semua sampel tingkat rendah dalam panduan menggunakan pola ini.

> **Note**
>
> Segmen kode sebelumnya menentukan `AccountId` ketika membuat permintaan. Namun, saat menggunakan AWS SDK for .NET, permintaan `AccountId` dalam adalah opsional, dan oleh karena itu semua contoh tingkat rendah dalam panduan ini tidak menetapkan nilai ini. `AccountId`Ini adalah Akun AWS ID. Nilai ini harus cocok dengan Akun AWS ID yang terkait dengan kredensional yang digunakan untuk menandatangani permintaan. Anda dapat menentukan Akun AWS ID atau opsional '-', dalam hal ini Amazon Glacier Akun AWS menggunakan ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menentukan ID Akun, jangan sertakan tanda hubung di dalamnya. Saat menggunakan AWS SDK for .NET, jika Anda tidak memberikan ID akun, pustaka akan menyetel ID akun menjadi '-'.

Menggunakan API Tingkat Tinggi

Untuk lebih menyederhanakan pengembangan aplikasi Anda, AWS SDK for .NET menyediakan `ArchiveTransferManager` kelas yang mengimplementasikan abstraksi tingkat yang lebih tinggi untuk beberapa metode di API tingkat rendah. Ini menyediakan metode yang berguna, seperti `Upload` dan `Download`, untuk operasi arsip.

Sebagai contoh, potongan kode C# berikut menggunakan metode tingkat tinggi `Upload` untuk mengunggah arsip.

```
string vaultName = "examplevault";
```

```
string archiveToUpload = "c:\folder\exampleArchive.zip";

var manager = new ArchiveTransferManager(Amazon.RegionEndpoint.USEast1);
string archiveId = manager.Upload(vaultName, "archive description",
 archiveToUpload).ArchiveId;
```

Perhatikan bahwa operasi apa pun yang Anda lakukan berlaku untuk AWS Wilayah yang Anda tentukan saat membuat `ArchiveTransferManager` objek. Semua contoh tingkat tinggi dalam panduan ini menggunakan pola ini.

> **Note**
>
> Kelas `ArchiveTransferManager` tingkat tinggi masih memerlukan klien `AmazonGlacierClient` tingkat rendah, yang Anda dapat teruskan secara eksplisit atau `ArchiveTransferManager` membuat klien.

Menjalankan Contoh Kode

Cara termudah untuk memulai dengan contoh kode .NET adalah dengan menginstal AWS SDK for .NET. Untuk informasi selengkapnya, buka Amazon SDK for .NET.

Prosedur berikut menguraikan langkah-langkah bagi Anda untuk menguji contoh kode yang diberikan dalam panduan ini.

Proses Umum Dalam Membuat Contoh Kode .NET (Menggunakan Visual Studio)

1	Buat profil kredensial untuk kredensional Anda seperti yang AWS dijelaskan dalam topik Amazon SDK for .NET Mengonfigurasi kredensial. AWS
2	Buat proyek Visual Studio baru menggunakan templat Proyek Kosong AWS .
3	Ganti kode dalam file proyek, `Program.cs` , dengan kode di bagian yang sedang Anda baca.
4	Jalankan kode tersebut. Verifikasi bahwa objek dibuat menggunakan Konsol Manajemen AWS. Untuk informasi lebih lanjut tentang Konsol Manajemen AWS, kunjungi http://aws.amazon.com/console/.

Mengatur Titik Akhir

Secara default, AWS SDK for .NET menetapkan titik akhir ke Wilayah Barat AS (Oregon) (`https://glacier.us-west-2.amazonaws.com`). Anda dapat mengatur titik akhir ke AWS Wilayah lain seperti yang ditunjukkan pada cuplikan C# berikut.

Potongan berikut menunjukkan cara mengatur titik akhir ke Wilayah US West (Oregon) (`us-west-2`) di API tingkat rendah.

Example

```
AmazonGlacierClient client = new AmazonGlacierClient(Amazon.RegionEndpoint.USWest2);
```

Potongan berikut menunjukkan cara mengatur titik akhir ke Wilayah US West (Oregon) di API tingkat tinggi.

```
var manager = new ArchiveTransferManager(Amazon.RegionEndpoint.USWest2);
```

Untuk daftar AWS Wilayah dan titik akhir yang didukung saat ini, lihatMengakses Amazon Glacier.

Contoh kode untuk Amazon Glacier menggunakan AWS SDKs

Contoh kode berikut menunjukkan cara menggunakan Amazon Glacier dengan kit pengembangan perangkat lunak (AWS SDK).

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Skenario adalah contoh kode yang menunjukkan kepada Anda bagaimana menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan atau dikombinasikan dengan yang lain Layanan AWS.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan Amazon Glacier dengan SDK AWS. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Contoh kode

- Contoh dasar untuk Amazon Glacier menggunakan AWS SDKs
  - Halo Amazon Glacier
  - Tindakan untuk Amazon Glacier menggunakan AWS SDKs
    - Gunakan AddTagsToVault dengan AWS SDK atau CLI
    - Gunakan CreateVault dengan AWS SDK atau CLI
    - Gunakan DeleteArchive dengan AWS SDK atau CLI
    - Gunakan DeleteVault dengan AWS SDK atau CLI
    - Gunakan DeleteVaultNotifications dengan AWS SDK atau CLI
    - Gunakan DescribeJob dengan AWS SDK atau CLI
    - Gunakan DescribeVault dengan AWS SDK atau CLI
    - Gunakan GetJobOutput dengan AWS SDK atau CLI
    - Gunakan GetVaultNotifications dengan AWS SDK atau CLI
    - Gunakan InitiateJob dengan AWS SDK atau CLI
    - Gunakan ListJobs dengan AWS SDK atau CLI
    - Gunakan ListTagsForVault dengan AWS SDK atau CLI

    - Gunakan ListVaults dengan AWS SDK atau CLI
    - Gunakan SetVaultNotifications dengan AWS SDK atau CLI
    - Gunakan UploadArchive dengan AWS SDK atau CLI
    - Gunakan UploadMultipartPart dengan AWS SDK atau CLI
- Skenario untuk Amazon Glacier menggunakan AWS SDKs
    - Arsipkan file ke Amazon Glacier, dapatkan notifikasi, dan mulai pekerjaan menggunakan SDK AWS
    - Dapatkan konten arsip Amazon Glacier dan hapus arsip menggunakan SDK AWS

Contoh dasar untuk Amazon Glacier menggunakan AWS SDKs

Contoh kode berikut menunjukkan cara menggunakan dasar-dasar Amazon AWS SDKs Glacier dengan.

Contoh

- Halo Amazon Glacier
- Tindakan untuk Amazon Glacier menggunakan AWS SDKs
    - Gunakan AddTagsToVault dengan AWS SDK atau CLI
    - Gunakan CreateVault dengan AWS SDK atau CLI
    - Gunakan DeleteArchive dengan AWS SDK atau CLI
    - Gunakan DeleteVault dengan AWS SDK atau CLI
    - Gunakan DeleteVaultNotifications dengan AWS SDK atau CLI
    - Gunakan DescribeJob dengan AWS SDK atau CLI
    - Gunakan DescribeVault dengan AWS SDK atau CLI
    - Gunakan GetJobOutput dengan AWS SDK atau CLI
    - Gunakan GetVaultNotifications dengan AWS SDK atau CLI
    - Gunakan InitiateJob dengan AWS SDK atau CLI
    - Gunakan ListJobs dengan AWS SDK atau CLI
    - Gunakan ListTagsForVault dengan AWS SDK atau CLI
    - Gunakan ListVaults dengan AWS SDK atau CLI
    - Gunakan SetVaultNotifications dengan AWS SDK atau CLI
    - Gunakan UploadArchive dengan AWS SDK atau CLI

- Gunakan UploadMultipartPart dengan AWS SDK atau CLI

Halo Amazon Glacier

Contoh kode berikut menunjukkan cara memulai menggunakan Amazon Glacier.

.NET

SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
using Amazon.Glacier;
using Amazon.Glacier.Model;

namespace GlacierActions;

public static class HelloGlacier
{
    static async Task Main()
    {
        var glacierService = new AmazonGlacierClient();

        Console.WriteLine("Hello Amazon Glacier!");
        Console.WriteLine("Let's list your Glacier vaults:");

        // You can use await and any of the async methods to get a response.
        // Let's get the vaults using a paginator.
        var glacierVaultPaginator = glacierService.Paginators.ListVaults(
            new ListVaultsRequest { AccountId = "-" });

        await foreach (var vault in glacierVaultPaginator.VaultList)
        {
            Console.WriteLine($"{vault.CreationDate}:{vault.VaultName}, ARN:
{vault.VaultARN}");
        }
```

```
    }
}
```

- Untuk detail API, lihat ListVaultsdi Referensi AWS SDK for .NET API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan Amazon Glacier dengan SDK AWS. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Tindakan untuk Amazon Glacier menggunakan AWS SDKs

Contoh kode berikut menunjukkan cara melakukan tindakan Amazon Glacier individual dengan. AWS SDKs Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan instruksi untuk mengatur dan menjalankan kode.

Kutipan ini menyebut Amazon Glacier API dan merupakan kutipan kode dari program yang lebih besar yang harus dijalankan dalam konteks. Anda dapat melihat tindakan dalam konteks diSkenario untuk Amazon Glacier menggunakan AWS SDKs.

Contoh berikut hanya mencakup tindakan yang paling umum digunakan. Untuk daftar lengkapnya, lihat Referensi API Amazon Glacier.

Contoh

- Gunakan AddTagsToVault dengan AWS SDK atau CLI
- Gunakan CreateVault dengan AWS SDK atau CLI
- Gunakan DeleteArchive dengan AWS SDK atau CLI
- Gunakan DeleteVault dengan AWS SDK atau CLI
- Gunakan DeleteVaultNotifications dengan AWS SDK atau CLI
- Gunakan DescribeJob dengan AWS SDK atau CLI
- Gunakan DescribeVault dengan AWS SDK atau CLI
- Gunakan GetJobOutput dengan AWS SDK atau CLI
- Gunakan GetVaultNotifications dengan AWS SDK atau CLI
- Gunakan InitiateJob dengan AWS SDK atau CLI
- Gunakan ListJobs dengan AWS SDK atau CLI
- Gunakan ListTagsForVault dengan AWS SDK atau CLI

- Gunakan ListVaults dengan AWS SDK atau CLI
- Gunakan SetVaultNotifications dengan AWS SDK atau CLI
- Gunakan UploadArchive dengan AWS SDK atau CLI
- Gunakan UploadMultipartPart dengan AWS SDK atau CLI

Gunakan **`AddTagsToVault`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`AddTagsToVault`.

.NET

SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   /// <summary>
   /// Add tags to the items in an Amazon S3 Glacier vault.
   /// </summary>
   /// <param name="vaultName">The name of the vault to add tags to.</param>
   /// <param name="key">The name of the object to tag.</param>
   /// <param name="value">The tag value to add.</param>
   /// <returns>A Boolean value indicating the success of the action.</returns>
   public async Task<bool> AddTagsToVaultAsync(string vaultName, string key,
string value)
   {
       var request = new AddTagsToVaultRequest
       {
           Tags = new Dictionary<string, string>
               {
                   { key, value },
               },
           AccountId = "-",
           VaultName = vaultName,
       };

       var response = await _glacierService.AddTagsToVaultAsync(request);
```

```
        return response.HttpStatusCode == HttpStatusCode.NoContent;
    }
```

- Untuk detail API, lihat AddTagsToVaultdi Referensi AWS SDK for .NET API.

CLI

AWS CLI

Perintah berikut menambahkan dua tag ke vault bernamamy-vault:

```
aws glacier add-tags-to-vault --account-id - --vault-name my-vault --
tags id=1234,date=july2015
```

Amazon Glacier memerlukan argumen ID akun saat melakukan operasi, tetapi Anda dapat menggunakan tanda hubung untuk menentukan akun yang sedang digunakan.

- Untuk detail API, lihat AddTagsToVaultdi Referensi AWS CLI Perintah.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan Amazon Glacier dengan SDK AWS. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`CreateVault`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`CreateVault`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Mengarsipkan file, mendapatkan notifikasi, dan memulai pekerjaan

.NET

SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
/// <summary>
/// Create an Amazon S3 Glacier vault.
/// </summary>
/// <param name="vaultName">The name of the vault to create.</param>
/// <returns>A Boolean value indicating the success of the action.</returns>
public async Task<bool> CreateVaultAsync(string vaultName)
{
    var request = new CreateVaultRequest
    {
        // Setting the AccountId to "-" means that
        // the account associated with the current
        // account will be used.
        AccountId = "-",
        VaultName = vaultName,
    };

    var response = await _glacierService.CreateVaultAsync(request);

    Console.WriteLine($"Created {vaultName} at: {response.Location}");

    return response.HttpStatusCode == HttpStatusCode.Created;
}
```

- Untuk detail API, lihat CreateVaultdi Referensi AWS SDK for .NET API.

CLI

AWS CLI

Perintah berikut membuat vault baru bernamamy-vault:

```
aws glacier create-vault --vault-name my-vault --account-id -
```

Amazon Glacier memerlukan argumen ID akun saat melakukan operasi, tetapi Anda dapat menggunakan tanda hubung untuk menentukan akun yang sedang digunakan.

- Untuk detail API, lihat CreateVaultdi Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.glacier.GlacierClient;
import software.amazon.awssdk.services.glacier.model.CreateVaultRequest;
import software.amazon.awssdk.services.glacier.model.CreateVaultResponse;
import software.amazon.awssdk.services.glacier.model.GlacierException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CreateVault {
    public static void main(String[] args) {
        final String usage = """

                Usage:    <vaultName>

                Where:
                   vaultName - The name of the vault to create.
```

```
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String vaultName = args[0];
        GlacierClient glacier = GlacierClient.builder()
                .region(Region.US_EAST_1)
                .build();

        createGlacierVault(glacier, vaultName);
        glacier.close();
    }

    public static void createGlacierVault(GlacierClient glacier, String
 vaultName) {
        try {
            CreateVaultRequest vaultRequest = CreateVaultRequest.builder()
                    .vaultName(vaultName)
                    .build();

            CreateVaultResponse createVaultResult =
 glacier.createVault(vaultRequest);
            System.out.println("The URI of the new vault is " +
 createVaultResult.location());

        } catch (GlacierException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Untuk detail API, lihat CreateVaultdi Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

Buat klien.

```
const { GlacierClient } = require("@aws-sdk/client-glacier");
// Set the AWS Region.
const REGION = "REGION";
//Set the Redshift Service Object
const glacierClient = new GlacierClient({ region: REGION });
export { glacierClient };
```

Buat lemari besi.

```
// Load the SDK for JavaScript
import { CreateVaultCommand } from "@aws-sdk/client-glacier";
import { glacierClient } from "./libs/glacierClient.js";

// Set the parameters
const vaultname = "VAULT_NAME"; // VAULT_NAME
const params = { vaultName: vaultname };

const run = async () => {
  try {
    const data = await glacierClient.send(new CreateVaultCommand(params));
    console.log("Success, vault created!");
    return data; // For unit tests.
  } catch (err) {
    console.log("Error");
  }
};
run();
```

- Untuk informasi selengkapnya, silakan lihat Panduan Developer AWS SDK for JavaScript.
- Untuk detail API, lihat CreateVaultdi Referensi AWS SDK for JavaScript API.

SDK untuk JavaScript (v2)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
// Load the SDK for JavaScript
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create a new service object
var glacier = new AWS.Glacier({ apiVersion: "2012-06-01" });
// Call Glacier to create the vault
glacier.createVault({ vaultName: "YOUR_VAULT_NAME" }, function (err) {
  if (!err) {
    console.log("Created vault!");
  }
});
```

- Untuk informasi selengkapnya, silakan lihat Panduan Developer AWS SDK for JavaScript.
- Untuk detail API, lihat CreateVaultdi Referensi AWS SDK for JavaScript API.

PowerShell

Alat untuk PowerShell V4

Contoh 1: Membuat vault baru untuk akun pengguna. Karena tidak ada nilai yang diberikan ke AccountId parameter -, cmdlet menggunakan default “-” yang menunjukkan akun saat ini.

```
New-GLCVault -VaultName myvault
```

Output:

```
/01234567812/vaults/myvault
```

- Untuk detail API, lihat CreateVaultdi Referensi AWS Tools for PowerShell Cmdlet (V4).

Alat untuk PowerShell V5

Contoh 1: Membuat vault baru untuk akun pengguna. Karena tidak ada nilai yang diberikan ke AccountId parameter -, cmdlet menggunakan default "-" yang menunjukkan akun saat ini.

```
New-GLCVault -VaultName myvault
```

Output:

```
/01234567812/vaults/myvault
```

- Untuk detail API, lihat CreateVaultdi Referensi AWS Tools for PowerShell Cmdlet (V5).

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class GlacierWrapper:
    """Encapsulates Amazon S3 Glacier API operations."""

    def __init__(self, glacier_resource):
        """
        :param glacier_resource: A Boto3 Amazon S3 Glacier resource.
        """
        self.glacier_resource = glacier_resource


    def create_vault(self, vault_name):
        """
        Creates a vault.
```

```
        :param vault_name: The name to give the vault.
        :return: The newly created vault.
        """
        try:
            vault = self.glacier_resource.create_vault(vaultName=vault_name)
            logger.info("Created vault %s.", vault_name)
        except ClientError:
            logger.exception("Couldn't create vault %s.", vault_name)
            raise
        else:
            return vault
```

- Untuk detail API, lihat CreateVaultdi AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan Amazon Glacier dengan SDK AWS. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`DeleteArchive`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`DeleteArchive`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Dapatkan konten arsip dan hapus arsip

CLI

AWS CLI

Untuk menghapus arsip dari vault

`delete-archive`Contoh berikut menghapus arsip yang ditentukan dari`example_vault`.

```
aws glacier delete-archive \
    --account-id 111122223333  \
    --vault-name example_vault \
```

```
    --archive-id Sc0u9ZP8yaWkmh-XGlIvAVprtLhaLCGnNwNl5I5x9HqPIkX5mjc0DrId3Ln-
Gi_k2HzmlIDZUz117KSdVMdMXLuFWi9PJUitxWO73edQ43eTlMWkH0pd9zVSAuV_XXZBVhKhyGhJ7w
```

Perintah ini tidak menghasilkan output.

- Untuk detail API, lihat DeleteArchivedi Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.glacier.GlacierClient;
import software.amazon.awssdk.services.glacier.model.DeleteArchiveRequest;
import software.amazon.awssdk.services.glacier.model.GlacierException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DeleteArchive {
    public static void main(String[] args) {
        final String usage = """

                Usage:    <vaultName> <accountId> <archiveId>

                Where:
                   vaultName - The name of the vault that contains the archive to
 delete.
                   accountId - The account ID value.
                   archiveId - The archive ID value.
```

```
                """;

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        String vaultName = args[0];
        String accountId = args[1];
        String archiveId = args[2];
        GlacierClient glacier = GlacierClient.builder()
                .region(Region.US_EAST_1)
                .build();

        deleteGlacierArchive(glacier, vaultName, accountId, archiveId);
        glacier.close();
    }

    public static void deleteGlacierArchive(GlacierClient glacier, String
 vaultName, String accountId,
            String archiveId) {
        try {
            DeleteArchiveRequest delArcRequest = DeleteArchiveRequest.builder()
                    .vaultName(vaultName)
                    .accountId(accountId)
                    .archiveId(archiveId)
                    .build();

            glacier.deleteArchive(delArcRequest);
            System.out.println("The archive was deleted.");

        } catch (GlacierException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Untuk detail API, lihat DeleteArchivedi Referensi AWS SDK for Java 2.x API.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class GlacierWrapper:
    """Encapsulates Amazon S3 Glacier API operations."""

    def __init__(self, glacier_resource):
        """
        :param glacier_resource: A Boto3 Amazon S3 Glacier resource.
        """
        self.glacier_resource = glacier_resource


    @staticmethod
    def delete_archive(archive):
        """
        Deletes an archive from a vault.

        :param archive: The archive to delete.
        """
        try:
            archive.delete()
            logger.info(
                "Deleted archive %s from vault %s.", archive.id,
 archive.vault_name
            )
        except ClientError:
            logger.exception("Couldn't delete archive %s.", archive.id)
            raise
```

- Untuk detail API, lihat DeleteArchivedi AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan Amazon Glacier dengan SDK AWS. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`DeleteVault`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`DeleteVault`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Dapatkan konten arsip dan hapus arsip

CLI

AWS CLI

Perintah berikut menghapus vault bernama: `my-vault`

```
aws glacier delete-vault --vault-name my-vault --account-id -
```

Perintah ini tidak menghasilkan output apa pun. Amazon Glacier memerlukan argumen ID akun saat melakukan operasi, tetapi Anda dapat menggunakan tanda hubung untuk menentukan akun yang sedang digunakan.

- Untuk detail API, lihat DeleteVaultdi Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.glacier.GlacierClient;
import software.amazon.awssdk.services.glacier.model.DeleteVaultRequest;
```

```
import software.amazon.awssdk.services.glacier.model.GlacierException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DeleteVault {
    public static void main(String[] args) {

        final String usage = """

                Usage:    <vaultName>

                Where:
                   vaultName - The name of the vault to delete.\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String vaultName = args[0];
        GlacierClient glacier = GlacierClient.builder()
                .region(Region.US_EAST_1)
                .build();

        deleteGlacierVault(glacier, vaultName);
        glacier.close();
    }

    public static void deleteGlacierVault(GlacierClient glacier, String
 vaultName) {
        try {
            DeleteVaultRequest delVaultRequest = DeleteVaultRequest.builder()
                    .vaultName(vaultName)
                    .build();

            glacier.deleteVault(delVaultRequest);
```

```
            System.out.println("The vault was deleted!");

        } catch (GlacierException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Untuk detail API, lihat DeleteVaultdi Referensi AWS SDK for Java 2.x API.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class GlacierWrapper:
    """Encapsulates Amazon S3 Glacier API operations."""

    def __init__(self, glacier_resource):
        """
        :param glacier_resource: A Boto3 Amazon S3 Glacier resource.
        """
        self.glacier_resource = glacier_resource


    @staticmethod
    def delete_vault(vault):
        """
        Deletes a vault.

        :param vault: The vault to delete.
        """
        try:
            vault.delete()
            logger.info("Deleted vault %s.", vault.name)
```

```
        except ClientError:
            logger.exception("Couldn't delete vault %s.", vault.name)
            raise
```

- Untuk detail API, lihat DeleteVaultdi AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan Amazon Glacier dengan SDK AWS. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`DeleteVaultNotifications`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`DeleteVaultNotifications`.

CLI

AWS CLI

Untuk menghapus notifikasi SNS untuk vault

`delete-vault-notifications`Contoh berikut menghapus notifikasi yang dikirim oleh Amazon Simple Notification Service (Amazon SNS) untuk vault yang ditentukan.

```
aws glacier delete-vault-notifications \
    --account-id 111122223333 \
    --vault-name example_vault
```

Perintah ini tidak menghasilkan output.

- Untuk detail API, lihat DeleteVaultNotificationsdi Referensi AWS CLI Perintah.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class GlacierWrapper:
    """Encapsulates Amazon S3 Glacier API operations."""

    def __init__(self, glacier_resource):
        """
        :param glacier_resource: A Boto3 Amazon S3 Glacier resource.
        """
        self.glacier_resource = glacier_resource


    @staticmethod
    def stop_notifications(notification):
        """
        Stops notifications to the configured Amazon SNS topic.

        :param notification: The notification configuration to remove.
        """
        try:
            notification.delete()
            logger.info("Notifications stopped.")
        except ClientError:
            logger.exception("Couldn't stop notifications.")
            raise
```

- Untuk detail API, lihat DeleteVaultNotificationsdi AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan Amazon Glacier dengan SDK AWS. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`DescribeJob`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`DescribeJob`.

CLI

AWS CLI

Perintah berikut mengambil informasi tentang pekerjaan pengambilan inventaris pada vault bernama: `my-vault`

```
aws glacier describe-job --account-id - --vault-name my-
vault --job-id zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
RO47Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61dsO4CnMW
```

Output:

```
{
    "InventoryRetrievalParameters": {
        "Format": "JSON"
    },
    "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-vault",
    "Completed": false,
    "JobId": "zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
RO47Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61dsO4CnMW",
    "Action": "InventoryRetrieval",
    "CreationDate": "2015-07-17T20:23:41.616Z",
    "StatusCode": "InProgress"
}
```

ID pekerjaan dapat ditemukan di output `aws glacier initiate-job` dan`aws glacier list-jobs`. Amazon Glacier memerlukan argumen ID akun saat melakukan operasi, tetapi Anda dapat menggunakan tanda hubung untuk menentukan akun yang sedang digunakan.

- Untuk detail API, lihat DescribeJobdi Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell V4

Contoh 1: Mengembalikan rincian pekerjaan yang ditentukan. Ketika pekerjaan berhasil diselesaikan, cmdlet Read- GCJob Output dapat digunakan untuk mengambil konten pekerjaan (arsip atau daftar inventaris) ke sistem file lokal.

```
Get-GLCJob -VaultName myvault -JobId "op1x...JSbthM"
```

Output:

```
Action                        : ArchiveRetrieval
ArchiveId                     : o9O9j...X-TpIhQJw
ArchiveSHA256TreeHash         : 79f3ea754c02f58...dc57bf4395b
ArchiveSizeInBytes            : 38034480
Completed                     : False
CompletionDate                : 1/1/0001 12:00:00 AM
CreationDate                  : 12/13/2018 11:00:14 AM
InventoryRetrievalParameters :
InventorySizeInBytes          : 0
JobDescription                :
JobId                         : op1x...JSbthM
JobOutputPath                 :
OutputLocation                :
RetrievalByteRange            : 0-38034479
SelectParameters              :
SHA256TreeHash                : 79f3ea754c02f58...dc57bf4395b
SNSTopic                      :
StatusCode                    : InProgress
StatusMessage                 :
Tier                          : Standard
VaultARN                      : arn:aws:glacier:us-west-2:012345678912:vaults/test
```

- Untuk detail API, lihat DescribeJobdi Referensi AWS Tools for PowerShell Cmdlet (V4).

Alat untuk PowerShell V5

Contoh 1: Mengembalikan rincian pekerjaan yang ditentukan. Ketika pekerjaan berhasil diselesaikan, cmdlet Read- GCJob Output dapat digunakan untuk mengambil konten pekerjaan (arsip atau daftar inventaris) ke sistem file lokal.

```
Get-GLCJob -VaultName myvault -JobId "op1x...JSbthM"
```

Output:

```
Action                        : ArchiveRetrieval
ArchiveId                     : o9O9j...X-TpIhQJw
ArchiveSHA256TreeHash         : 79f3ea754c02f58...dc57bf4395b
ArchiveSizeInBytes            : 38034480
Completed                     : False
CompletionDate                : 1/1/0001 12:00:00 AM
```

```
CreationDate                 : 12/13/2018 11:00:14 AM
InventoryRetrievalParameters :
InventorySizeInBytes         : 0
JobDescription               :
JobId                        : op1x...JSbthM
JobOutputPath                :
OutputLocation               :
RetrievalByteRange           : 0-38034479
SelectParameters             :
SHA256TreeHash               : 79f3ea754c02f58...dc57bf4395b
SNSTopic                     :
StatusCode                   : InProgress
StatusMessage                :
Tier                         : Standard
VaultARN                     : arn:aws:glacier:us-west-2:012345678912:vaults/test
```

- Untuk detail API, lihat DescribeJobdi Referensi AWS Tools for PowerShell Cmdlet (V5).

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class GlacierWrapper:
    """Encapsulates Amazon S3 Glacier API operations."""

    def __init__(self, glacier_resource):
        """
        :param glacier_resource: A Boto3 Amazon S3 Glacier resource.
        """
        self.glacier_resource = glacier_resource


    @staticmethod
    def get_job_status(job):
        """
        Gets the status of a job.
```

```
        :param job: The job to query.
        :return: The current status of the job.
        """
        try:
            job.load()
            logger.info(
                "Job %s is performing action %s and has status %s.",
                job.id,
                job.action,
                job.status_code,
            )
        except ClientError:
            logger.exception("Couldn't get status for job %s.", job.id)
            raise
        else:
            return job.status_code
```

- Untuk detail API, lihat DescribeJobdi AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan Amazon Glacier dengan SDK AWS. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`DescribeVault`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`DescribeVault`.

.NET

SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /// <summary>
```

```
    /// Describe an Amazon S3 Glacier vault.
    /// </summary>
    /// <param name="vaultName">The name of the vault to describe.</param>
    /// <returns>The Amazon Resource Name (ARN) of the vault.</returns>
    public async Task<string> DescribeVaultAsync(string vaultName)
    {
        var request = new DescribeVaultRequest
        {
            AccountId = "-",
            VaultName = vaultName,
        };

        var response = await _glacierService.DescribeVaultAsync(request);

        // Display the information about the vault.
        Console.WriteLine($"{response.VaultName}\tARN: {response.VaultARN}");
        Console.WriteLine($"Created on: {response.CreationDate}\tNumber
of Archives: {response.NumberOfArchives}\tSize (in bytes):
{response.SizeInBytes}");
        if (response.LastInventoryDate != DateTime.MinValue)
        {
            Console.WriteLine($"Last inventory: {response.LastInventoryDate}");
        }

        return response.VaultARN;
    }
```

- Untuk detail API, lihat DescribeVaultdi Referensi AWS SDK for .NET API.

CLI

AWS CLI

Perintah berikut mengambil data tentang vault bernama: `my-vault`

```
aws glacier describe-vault --vault-name my-vault --account-id -
```

Amazon Glacier memerlukan argumen ID akun saat melakukan operasi, tetapi Anda dapat menggunakan tanda hubung untuk menentukan akun yang sedang digunakan.

- Untuk detail API, lihat DescribeVaultdi Referensi AWS CLI Perintah.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan Amazon Glacier dengan SDK AWS. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan `GetJobOutput` dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`GetJobOutput`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Dapatkan konten arsip dan hapus arsip

CLI

AWS CLI

Perintah berikut menyimpan output dari pekerjaan inventaris vault ke file di direktori saat ini bernama`output.json`:

```
aws glacier get-job-output --account-id - --vault-name my-
vault --job-id zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
RO47Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61ds04CnMW output.json
```

`job-id`Tersedia dalam output dari`aws glacier list-jobs`. Perhatikan bahwa nama file output adalah argumen posisi yang tidak diawali dengan nama opsi. Amazon Glacier memerlukan argumen ID akun saat melakukan operasi, tetapi Anda dapat menggunakan tanda hubung untuk menentukan akun yang sedang digunakan.

Output:

```
{
    "status": 200,
    "acceptRanges": "bytes",
    "contentType": "application/json"
}
```

`output.json`:

```
{"VaultARN":"arn:aws:glacier:us-west-2:0123456789012:vaults/
my-vault","InventoryDate":"2015-04-07T00:26:18Z","ArchiveList":
```

```
[{"ArchiveId":"kKB7ymWJVpPSwhGP6ycSOAekp9ZYe_--zM_mw6k76ZFGEIWQX-
ybtRDvc2VkPSDtfKmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hHO0bjbGehXTcApVud_wyDw","ArchiveDescription":"multipart
 upload
 test","CreationDate":"2015-04-06T22:24:34Z","Size":3145728,"SHA256TreeHash":"9628195fcdb
```

- Untuk detail API, lihat GetJobOutputdi Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell V4

Contoh 1: Mengunduh konten arsip yang dijadwalkan untuk pengambilan dalam pekerjaan yang ditentukan dan menyimpan konten ke dalam file pada disk. Unduhan memvalidasi checksum untuk Anda, jika tersedia. Jika diinginkan seluruh respons termasuk checksum dapat dikembalikan dengan menentukan**`-Select '*'`**.

```
Read-GLCJobOutput -VaultName myvault -JobId "HSWjArc...Zq2XLiW" -FilePath "c:
\temp\blue.bin"
```

- Untuk detail API, lihat GetJobOutputdi Referensi AWS Tools for PowerShell Cmdlet (V4).

Alat untuk PowerShell V5

Contoh 1: Mengunduh konten arsip yang dijadwalkan untuk pengambilan dalam pekerjaan yang ditentukan dan menyimpan konten ke dalam file pada disk. Unduhan memvalidasi checksum untuk Anda, jika tersedia. Jika diinginkan seluruh respons termasuk checksum dapat dikembalikan dengan menentukan**`-Select '*'`**.

```
Read-GLCJobOutput -VaultName myvault -JobId "HSWjArc...Zq2XLiW" -FilePath "c:
\temp\blue.bin"
```

- Untuk detail API, lihat GetJobOutputdi Referensi AWS Tools for PowerShell Cmdlet (V5).

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class GlacierWrapper:
    """Encapsulates Amazon S3 Glacier API operations."""

    def __init__(self, glacier_resource):
        """
        :param glacier_resource: A Boto3 Amazon S3 Glacier resource.
        """
        self.glacier_resource = glacier_resource


    @staticmethod
    def get_job_output(job):
        """
        Gets the output of a job, such as a vault inventory or the contents of an
        archive.

        :param job: The job to get output from.
        :return: The job output, in bytes.
        """
        try:
            response = job.get_output()
            out_bytes = response["body"].read()
            logger.info("Read %s bytes from job %s.", len(out_bytes), job.id)
            if "archiveDescription" in response:
                logger.info(
                    "These bytes are described as '%s'",
 response["archiveDescription"]
                )
        except ClientError:
            logger.exception("Couldn't get output for job %s.", job.id)
            raise
        else:
```

```
            return out_bytes
```

- Untuk detail API, lihat GetJobOutputdi AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan Amazon Glacier dengan SDK AWS. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`GetVaultNotifications`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`GetVaultNotifications`.

CLI

AWS CLI

Perintah berikut mendapatkan deskripsi konfigurasi notifikasi untuk vault bernama`my-vault`:

```
aws glacier get-vault-notifications --account-id - --vault-name my-vault
```

Output:

```
{
    "vaultNotificationConfig": {
        "Events": [
            "InventoryRetrievalCompleted",
            "ArchiveRetrievalCompleted"
        ],
        "SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-vault"
    }
}
```

Jika tidak ada pemberitahuan yang dikonfigurasi untuk brankas, kesalahan akan dikembalikan. Amazon Glacier memerlukan argumen ID akun saat melakukan operasi, tetapi Anda dapat menggunakan tanda hubung untuk menentukan akun yang sedang digunakan.

- Untuk detail API, lihat GetVaultNotificationsdi Referensi AWS CLI Perintah.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class GlacierWrapper:
    """Encapsulates Amazon S3 Glacier API operations."""

    def __init__(self, glacier_resource):
        """
        :param glacier_resource: A Boto3 Amazon S3 Glacier resource.
        """
        self.glacier_resource = glacier_resource


    @staticmethod
    def get_notification(vault):
        """
        Gets the currently notification configuration for a vault.

        :param vault: The vault to query.
        :return: The notification configuration for the specified vault.
        """
        try:
            notification = vault.Notification()
            logger.info(
                "Vault %s notifies %s on %s events.",
                vault.name,
                notification.sns_topic,
                notification.events,
            )
        except ClientError:
            logger.exception("Couldn't get notification data for %s.",
 vault.name)
            raise
        else:
            return notification
```

- Untuk detail API, lihat GetVaultNotificationsdi AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan Amazon Glacier dengan SDK AWS. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`InitiateJob`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`InitiateJob`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Mengarsipkan file, mendapatkan notifikasi, dan memulai pekerjaan

.NET

SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

Ambil arsip dari lemari besi. Contoh ini menggunakan ArchiveTransferManager kelas. Untuk detail API, lihat ArchiveTransferManager.

```
    /// <summary>
    /// Download an archive from an Amazon S3 Glacier vault using the Archive
    /// Transfer Manager.
    /// </summary>
    /// <param name="vaultName">The name of the vault containing the object.</
param>
    /// <param name="archiveId">The Id of the archive to download.</param>
    /// <param name="localFilePath">The local directory where the file will
    /// be stored after download.</param>
```

```
   /// <returns>Async Task.</returns>
   public async Task<bool> DownloadArchiveWithArchiveManagerAsync(string
vaultName, string archiveId, string localFilePath)
   {
       try
       {
           var manager = new ArchiveTransferManager(_glacierService);

           var options = new DownloadOptions
           {
               StreamTransferProgress = Progress!,
           };

           // Download an archive.
           Console.WriteLine("Initiating the archive retrieval job and then
polling SQS queue for the archive to be available.");
           Console.WriteLine("When the archive is available, downloading will
begin.");
           await manager.DownloadAsync(vaultName, archiveId, localFilePath,
options);

           return true;
       }
       catch (AmazonGlacierException ex)
       {
           Console.WriteLine(ex.Message);
           return false;
       }
   }

   /// <summary>
   /// Event handler to track the progress of the Archive Transfer Manager.
   /// </summary>
   /// <param name="sender">The object that raised the event.</param>
   /// <param name="args">The argument values from the object that raised the
   /// event.</param>
   static void Progress(object sender, StreamTransferProgressArgs args)
   {
       if (args.PercentDone != _currentPercentage)
       {
           _currentPercentage = args.PercentDone;
           Console.WriteLine($"Downloaded {_currentPercentage}%");
       }
   }
```

- Untuk detail API, lihat InitiateJobdi Referensi AWS SDK for .NET API.

CLI

AWS CLI

Perintah berikut memulai pekerjaan untuk mendapatkan inventaris `my-vault` vault:

```
aws glacier initiate-job --account-id - --vault-name my-vault --job-parameters
 '{"Type": "inventory-retrieval"}'
```

Output:

```
{
    "location": "/0123456789012/vaults/my-vault/jobs/
zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
RO47Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61dsO4CnMW",
    "jobId": "zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
RO47Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61dsO4CnMW"
}
```

Amazon Glacier memerlukan argumen ID akun saat melakukan operasi, tetapi Anda dapat menggunakan tanda hubung untuk menentukan akun yang sedang digunakan.

Perintah berikut memulai pekerjaan untuk mengambil arsip dari vault: `my-vault`

```
aws glacier initiate-job --account-id - --vault-name my-vault --job-
parameters file://job-archive-retrieval.json
```

`job-archive-retrieval.json`adalah file JSON di folder lokal yang menentukan jenis pekerjaan, ID arsip, dan beberapa parameter opsional:

```
{
  "Type": "archive-retrieval",
  "ArchiveId": "kKB7ymWJVpPSwhGP6ycSOAekp9ZYe_--zM_mw6k76ZFGEIWQX-
ybtRDvc2VkPSDtfKmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hHO0bjbGehXTcApVud_wyDw",
  "Description": "Retrieve archive on 2015-07-17",
```

```
  "SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-topic"
}
```

Arsip IDs tersedia dalam output `aws glacier upload-archive` dan`aws glacier get-job-output`.

Output:

```
{
    "location": "/011685312445/vaults/mwunderl/jobs/l7IL5-
EkXyEY9Ws95fClzIbk2O5uLYaFdAYOi-
azsX_Z8V6NH4yERHzars8wTKYQMX6nBDI9cMNHzyZJO59-8N9aHWav",
    "jobId": "l7IL5-EkXy2O5uLYaFdAYOiEY9Ws95fClzIbk-
azsX_Z8V6NH4yERHzars8wTKYQMX6nBDI9cMNHzyZJO59-8N9aHWav"
}
```

Lihat Memulai Pekerjaan di Referensi Amazon Glacier API untuk detail tentang format parameter pekerjaan.

- Untuk detail API, lihat InitiateJobdi Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

Ambil inventaris lemari besi.

```
import software.amazon.awssdk.core.ResponseBytes;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.glacier.GlacierClient;
import software.amazon.awssdk.services.glacier.model.JobParameters;
import software.amazon.awssdk.services.glacier.model.InitiateJobResponse;
import software.amazon.awssdk.services.glacier.model.GlacierException;
import software.amazon.awssdk.services.glacier.model.InitiateJobRequest;
import software.amazon.awssdk.services.glacier.model.DescribeJobRequest;
import software.amazon.awssdk.services.glacier.model.DescribeJobResponse;
```

```
import software.amazon.awssdk.services.glacier.model.GetJobOutputRequest;
import software.amazon.awssdk.services.glacier.model.GetJobOutputResponse;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ArchiveDownload {
    public static void main(String[] args) {

        final String usage = """

                Usage:    <vaultName> <accountId> <path>

                Where:
                   vaultName - The name of the vault.
                   accountId - The account ID value.
                   path - The path where the file is written to.
                """;

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        String vaultName = args[0];
        String accountId = args[1];
        String path = args[2];
        GlacierClient glacier = GlacierClient.builder()
                .region(Region.US_EAST_1)
                .build();

        String jobNum = createJob(glacier, vaultName, accountId);
        checkJob(glacier, jobNum, vaultName, accountId, path);
        glacier.close();
```

```
   }

   public static String createJob(GlacierClient glacier, String vaultName,
String accountId) {
       try {
           JobParameters job = JobParameters.builder()
                   .type("inventory-retrieval")
                   .build();

           InitiateJobRequest initJob = InitiateJobRequest.builder()
                   .jobParameters(job)
                   .accountId(accountId)
                   .vaultName(vaultName)
                   .build();

           InitiateJobResponse response = glacier.initiateJob(initJob);
           System.out.println("The job ID is: " + response.jobId());
           System.out.println("The relative URI path of the job is: " +
response.location());
           return response.jobId();

       } catch (GlacierException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);

       }
       return "";
   }

   // Poll S3 Glacier = Polling a Job may take 4-6 hours according to the
   // Documentation.
   public static void checkJob(GlacierClient glacier, String jobId, String name,
String account, String path) {
       try {
           boolean finished = false;
           String jobStatus;
           int yy = 0;

           while (!finished) {
               DescribeJobRequest jobRequest = DescribeJobRequest.builder()
                       .jobId(jobId)
                       .accountId(account)
                       .vaultName(name)
                       .build();
```

```
                DescribeJobResponse response = glacier.describeJob(jobRequest);
                jobStatus = response.statusCodeAsString();

                if (jobStatus.compareTo("Succeeded") == 0)
                    finished = true;
                else {
                    System.out.println(yy + " status is: " + jobStatus);
                    Thread.sleep(1000);
                }
                yy++;
            }

            System.out.println("Job has Succeeded");
            GetJobOutputRequest jobOutputRequest = GetJobOutputRequest.builder()
                    .jobId(jobId)
                    .vaultName(name)
                    .accountId(account)
                    .build();

            ResponseBytes<GetJobOutputResponse> objectBytes =
 glacier.getJobOutputAsBytes(jobOutputRequest);
            // Write the data to a local file.
            byte[] data = objectBytes.asByteArray();
            File myFile = new File(path);
            OutputStream os = new FileOutputStream(myFile);
            os.write(data);
            System.out.println("Successfully obtained bytes from a Glacier
 vault");
            os.close();

        } catch (GlacierException | InterruptedException | IOException e) {
            System.out.println(e.getMessage());
            System.exit(1);

        }
    }
}
```

- Untuk detail API, lihat InitiateJobdi Referensi AWS SDK for Java 2.x API.

PowerShell

Alat untuk PowerShell V4

Contoh 1: Memulai pekerjaan untuk mengambil arsip dari brankas tertentu yang dimiliki oleh pengguna. Status pekerjaan dapat diperiksa menggunakan Get- GLCJob cmdlet. Ketika pekerjaan berhasil diselesaikan, cmdlet Read- GCJob Output dapat digunakan untuk mengambil isi arsip ke sistem file lokal.

```
Start-GLCJob -VaultName myvault -JobType "archive-retrieval" -JobDescription
 "archive retrieval" -ArchiveId "o9O9j...TX-TpIhQJw"
```

Output:

```
JobId            JobOutputPath Location
-----            ------------- --------
op1x...JSbthM                  /012345678912/vaults/test/jobs/
op1xe...I4HqCHkSJSbthM
```

- Untuk detail API, lihat InitiateJobdi Referensi AWS Tools for PowerShell Cmdlet (V4).

Alat untuk PowerShell V5

Contoh 1: Memulai pekerjaan untuk mengambil arsip dari brankas tertentu yang dimiliki oleh pengguna. Status pekerjaan dapat diperiksa menggunakan Get- GLCJob cmdlet. Ketika pekerjaan berhasil diselesaikan, cmdlet Read- GCJob Output dapat digunakan untuk mengambil isi arsip ke sistem file lokal.

```
Start-GLCJob -VaultName myvault -JobType "archive-retrieval" -JobDescription
 "archive retrieval" -ArchiveId "o9O9j...TX-TpIhQJw"
```

Output:

```
JobId            JobOutputPath Location
-----            ------------- --------
op1x...JSbthM                  /012345678912/vaults/test/jobs/
op1xe...I4HqCHkSJSbthM
```

- Untuk detail API, lihat InitiateJobdi Referensi AWS Tools for PowerShell Cmdlet (V5).

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

Ambil inventaris lemari besi.

```
class GlacierWrapper:
    """Encapsulates Amazon S3 Glacier API operations."""

    def __init__(self, glacier_resource):
        """
        :param glacier_resource: A Boto3 Amazon S3 Glacier resource.
        """
        self.glacier_resource = glacier_resource


    @staticmethod
    def initiate_inventory_retrieval(vault):
        """
        Initiates an inventory retrieval job. The inventory describes the
 contents
        of the vault. Standard retrievals typically complete within 3–5 hours.
        When the job completes, you can get the inventory by calling
 get_output().

        :param vault: The vault to inventory.
        :return: The inventory retrieval job.
        """
        try:
            job = vault.initiate_inventory_retrieval()
            logger.info("Started %s job with ID %s.", job.action, job.id)
        except ClientError:
            logger.exception("Couldn't start job on vault %s.", vault.name)
            raise
        else:
            return job
```

Ambil arsip dari lemari besi.

```
class GlacierWrapper:
    """Encapsulates Amazon S3 Glacier API operations."""

    def __init__(self, glacier_resource):
        """
        :param glacier_resource: A Boto3 Amazon S3 Glacier resource.
        """
        self.glacier_resource = glacier_resource


    @staticmethod
    def initiate_archive_retrieval(archive):
        """
        Initiates an archive retrieval job. Standard retrievals typically
 complete
        within 3–5 hours. When the job completes, you can get the archive
 contents
        by calling get_output().

        :param archive: The archive to retrieve.
        :return: The archive retrieval job.
        """
        try:
            job = archive.initiate_archive_retrieval()
            logger.info("Started %s job with ID %s.", job.action, job.id)
        except ClientError:
            logger.exception("Couldn't start job on archive %s.", archive.id)
            raise
        else:
            return job
```

- Untuk detail API, lihat InitiateJobdi AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan Amazon Glacier dengan SDK AWS. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan `ListJobs` dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`ListJobs`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Mengarsipkan file, mendapatkan notifikasi, dan memulai pekerjaan
- Dapatkan konten arsip dan hapus arsip

.NET

SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   /// <summary>
   /// List Amazon S3 Glacier jobs.
   /// </summary>
   /// <param name="vaultName">The name of the vault to list jobs for.</param>
   /// <returns>A list of Amazon S3 Glacier jobs.</returns>
   public async Task<List<GlacierJobDescription>> ListJobsAsync(string
vaultName)
   {
       var request = new ListJobsRequest
       {
           // Using a hyphen "-" for the Account Id will
           // cause the SDK to use the Account Id associated
           // with the current account.
           AccountId = "-",
           VaultName = vaultName,
       };

       var response = await _glacierService.ListJobsAsync(request);

       return response.JobList;
   }
```

- Untuk detail API, lihat ListJobsdi Referensi AWS SDK for .NET API.

CLI

AWS CLI

Perintah berikut mencantumkan pekerjaan yang sedang berlangsung dan baru saja diselesaikan untuk vault bernama: `my-vault`

```
aws glacier list-jobs --account-id - --vault-name my-vault
```

Output:

```
{
    "JobList": [
        {
            "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-
vault",
            "RetrievalByteRange": "0-3145727",
            "SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-vault",
            "Completed": false,
            "SHA256TreeHash":
 "9628195fcdbcbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67",
            "JobId": "l7IL5-EkXyEY9Ws95fClzIbk2O5uLYaFdAYOi-
azsX_Z8V6NH4yERHzars8wTKYQMX6nBDI9cMNHzyZJO59-8N9aHWav",
            "ArchiveId": "kKB7ymWJVpPSwhGP6ycSOAekp9ZYe_--
zM_mw6k76ZFGEIWQX-ybtRDvc2VkPSDtfKmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hHO0bjbGehXTcApVud_wyDw",
            "JobDescription": "Retrieve archive on 2015-07-17",
            "ArchiveSizeInBytes": 3145728,
            "Action": "ArchiveRetrieval",
            "ArchiveSHA256TreeHash":
 "9628195fcdbcbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67",
            "CreationDate": "2015-07-17T21:16:13.840Z",
            "StatusCode": "InProgress"
        },
        {
            "InventoryRetrievalParameters": {
                "Format": "JSON"
```

```
            },
            "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-
vault",
            "Completed": false,
            "JobId": "zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
RO47Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61dsO4CnMW",
            "Action": "InventoryRetrieval",
            "CreationDate": "2015-07-17T20:23:41.616Z",
            "StatusCode": ""InProgress""
        }
    ]
}
```

Amazon Glacier memerlukan argumen ID akun saat melakukan operasi, tetapi Anda dapat menggunakan tanda hubung untuk menentukan akun yang sedang digunakan.

- Untuk detail API, lihat ListJobsdi Referensi AWS CLI Perintah.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class GlacierWrapper:
    """Encapsulates Amazon S3 Glacier API operations."""

    def __init__(self, glacier_resource):
        """
        :param glacier_resource: A Boto3 Amazon S3 Glacier resource.
        """
        self.glacier_resource = glacier_resource


    @staticmethod
    def list_jobs(vault, job_type):
        """
        Lists jobs by type for the specified vault.
```

```
        :param vault: The vault to query.
        :param job_type: The type of job to list.
        :return: The list of jobs of the requested type.
        """
        job_list = []
        try:
            if job_type == "all":
                jobs = vault.jobs.all()
            elif job_type == "in_progress":
                jobs = vault.jobs_in_progress.all()
            elif job_type == "completed":
                jobs = vault.completed_jobs.all()
            elif job_type == "succeeded":
                jobs = vault.succeeded_jobs.all()
            elif job_type == "failed":
                jobs = vault.failed_jobs.all()
            else:
                jobs = []
                logger.warning("%s isn't a type of job I can get.", job_type)
            for job in jobs:
                job_list.append(job)
                logger.info("Got %s %s job %s.", job_type, job.action, job.id)
        except ClientError:
            logger.exception("Couldn't get %s jobs from %s.", job_type,
vault.name)
            raise
        else:
            return job_list
```

- Untuk detail API, lihat ListJobsdi AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan Amazon Glacier dengan SDK AWS. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`ListTagsForVault`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`ListTagsForVault`.

.NET

SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /// <summary>
    /// List tags for an Amazon S3 Glacier vault.
    /// </summary>
    /// <param name="vaultName">The name of the vault to list tags for.</param>
    /// <returns>A dictionary listing the tags attached to each object in the
    /// vault and its tags.</returns>
    public async Task<Dictionary<string, string>> ListTagsForVaultAsync(string
vaultName)
    {
        var request = new ListTagsForVaultRequest
        {
            // Using a hyphen "-" for the Account Id will
            // cause the SDK to use the Account Id associated
            // with the default user.
            AccountId = "-",
            VaultName = vaultName,
        };

        var response = await _glacierService.ListTagsForVaultAsync(request);

        return response.Tags;
    }
```

- Untuk detail API, lihat ListTagsForVaultdi Referensi AWS SDK for .NET API.

CLI

AWS CLI

Perintah berikut mencantumkan tag yang diterapkan ke vault bernamamy-vault:

```
aws glacier list-tags-for-vault --account-id - --vault-name my-vault
```

Output:

```
{
    "Tags": {
        "date": "july2015",
        "id": "1234"
    }
}
```

Amazon Glacier memerlukan argumen ID akun saat melakukan operasi, tetapi Anda dapat menggunakan tanda hubung untuk menentukan akun yang sedang digunakan.

- Untuk detail API, lihat ListTagsForVaultdi Referensi AWS CLI Perintah.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan Amazon Glacier dengan SDK AWS. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`ListVaults`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`ListVaults`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Mengarsipkan file, mendapatkan notifikasi, dan memulai pekerjaan

.NET

SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
/// <summary>
/// List the Amazon S3 Glacier vaults associated with the current account.
/// </summary>
/// <returns>A list containing information about each vault.</returns>
public async Task<List<DescribeVaultOutput>> ListVaultsAsync()
{
    var glacierVaultPaginator = _glacierService.Paginators.ListVaults(
        new ListVaultsRequest { AccountId = "-" });
    var vaultList = new List<DescribeVaultOutput>();

    await foreach (var vault in glacierVaultPaginator.VaultList)
    {
        vaultList.Add(vault);
    }

    return vaultList;
}
```

- Untuk detail API, lihat ListVaultsdi Referensi AWS SDK for .NET API.

CLI

AWS CLI

Perintah berikut mencantumkan vault di akun dan wilayah default:

```
aws glacier list-vaults --account-id -
```

Output:

```
{
    "VaultList": [
        {
            "SizeInBytes": 3178496,
            "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-
vault",
            "LastInventoryDate": "2015-04-07T00:26:19.028Z",
            "VaultName": "my-vault",
            "NumberOfArchives": 1,
            "CreationDate": "2015-04-06T21:23:45.708Z"
        }
    ]
}
```

Amazon Glacier memerlukan argumen ID akun saat melakukan operasi, tetapi Anda dapat menggunakan tanda hubung untuk menentukan akun yang sedang digunakan.

- Untuk detail API, lihat ListVaultsdi Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.glacier.model.ListVaultsRequest;
import software.amazon.awssdk.services.glacier.model.ListVaultsResponse;
import software.amazon.awssdk.services.glacier.GlacierClient;
import software.amazon.awssdk.services.glacier.model.DescribeVaultOutput;
import software.amazon.awssdk.services.glacier.model.GlacierException;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
```

```
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ListVaults {
    public static void main(String[] args) {
        GlacierClient glacier = GlacierClient.builder()
                .region(Region.US_EAST_1)
                .build();

        listAllVault(glacier);
        glacier.close();
    }

    public static void listAllVault(GlacierClient glacier) {
        boolean listComplete = false;
        String newMarker = null;
        int totalVaults = 0;
        System.out.println("Your Amazon Glacier vaults:");
        try {
            while (!listComplete) {
                ListVaultsResponse response = null;
                if (newMarker != null) {
                    ListVaultsRequest request = ListVaultsRequest.builder()
                            .marker(newMarker)
                            .build();

                    response = glacier.listVaults(request);
                } else {
                    ListVaultsRequest request = ListVaultsRequest.builder()
                            .build();
                    response = glacier.listVaults(request);
                }

                List<DescribeVaultOutput> vaultList = response.vaultList();
                for (DescribeVaultOutput v : vaultList) {
                    totalVaults += 1;
                    System.out.println("* " + v.vaultName());
                }

                // Check for further results.
                newMarker = response.marker();
                if (newMarker == null) {
```

```
                    listComplete = true;
                }
            }

            if (totalVaults == 0) {
                System.out.println("No vaults found.");
            }

        } catch (GlacierException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Untuk detail API, lihat ListVaultsdi Referensi AWS SDK for Java 2.x API.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class GlacierWrapper:
    """Encapsulates Amazon S3 Glacier API operations."""

    def __init__(self, glacier_resource):
        """
        :param glacier_resource: A Boto3 Amazon S3 Glacier resource.
        """
        self.glacier_resource = glacier_resource


    def list_vaults(self):
        """
        Lists vaults for the current account.
        """
```

```
        try:
            for vault in self.glacier_resource.vaults.all():
                logger.info("Got vault %s.", vault.name)
        except ClientError:
            logger.exception("Couldn't list vaults.")
            raise
```

- Untuk detail API, lihat ListVaultsdi AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan Amazon Glacier dengan SDK AWS. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`SetVaultNotifications`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`SetVaultNotifications`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Mengarsipkan file, mendapatkan notifikasi, dan memulai pekerjaan

CLI

AWS CLI

Perintah berikut mengonfigurasi notifikasi SNS untuk vault bernama: `my-vault`

```
aws glacier set-vault-notifications --account-id - --vault-name my-vault --vault-
notification-config file://notificationconfig.json
```

`notificationconfig.json`adalah file JSON di folder saat ini yang menentukan topik SNS dan peristiwa yang akan dipublikasikan:

```
{
  "SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-vault",
  "Events": ["ArchiveRetrievalCompleted", "InventoryRetrievalCompleted"]
}
```

Amazon Glacier memerlukan argumen ID akun saat melakukan operasi, tetapi Anda dapat menggunakan tanda hubung untuk menentukan akun yang sedang digunakan.

- Untuk detail API, lihat SetVaultNotificationsdi Referensi AWS CLI Perintah.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class GlacierWrapper:
    """Encapsulates Amazon S3 Glacier API operations."""

    def __init__(self, glacier_resource):
        """
        :param glacier_resource: A Boto3 Amazon S3 Glacier resource.
        """
        self.glacier_resource = glacier_resource


    def set_notifications(self, vault, sns_topic_arn):
        """
        Sets an Amazon Simple Notification Service (Amazon SNS) topic as a target
        for notifications. Amazon S3 Glacier publishes messages to this topic for
        the configured list of events.

        :param vault: The vault to set up to publish notifications.
        :param sns_topic_arn: The Amazon Resource Name (ARN) of the topic that
                              receives notifications.
        :return: Data about the new notification configuration.
        """
        try:
            notification = self.glacier_resource.Notification("-", vault.name)
            notification.set(
                vaultNotificationConfig={
                    "SNSTopic": sns_topic_arn,
                    "Events": [
```

```
                        "ArchiveRetrievalCompleted",
                        "InventoryRetrievalCompleted",
                    ],
                }
            )
            logger.info(
                "Notifications will be sent to %s for events %s from %s.",
                notification.sns_topic,
                notification.events,
                notification.vault_name,
            )
        except ClientError:
            logger.exception(
                "Couldn't set notifications to %s on %s.", sns_topic_arn,
vault.name
            )
            raise
        else:
            return notification
```

- Untuk detail API, lihat SetVaultNotificationsdi AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan Amazon Glacier dengan SDK AWS. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`UploadArchive`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`UploadArchive`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Mengarsipkan file, mendapatkan notifikasi, dan memulai pekerjaan

.NET

SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   /// <summary>
   /// Upload an object to an Amazon S3 Glacier vault.
   /// </summary>
   /// <param name="vaultName">The name of the Amazon S3 Glacier vault to upload
   /// the archive to.</param>
   /// <param name="archiveFilePath">The file path of the archive to upload to
the vault.</param>
   /// <returns>A Boolean value indicating the success of the action.</returns>
   public async Task<string> UploadArchiveWithArchiveManager(string vaultName,
string archiveFilePath)
   {
       try
       {
           var manager = new ArchiveTransferManager(_glacierService);

           // Upload an archive.
           var response = await manager.UploadAsync(vaultName, "upload archive
test", archiveFilePath);
           return response.ArchiveId;
       }
       catch (AmazonGlacierException ex)
       {
           Console.WriteLine(ex.Message);
           return string.Empty;
       }
   }
```

- Untuk detail API, lihat UploadArchivedi Referensi AWS SDK for .NET API.

CLI

AWS CLI

Perintah berikut mengunggah arsip di folder saat ini bernama `archive.zip` ke vault bernama: `my-vault`

```
aws glacier upload-archive --account-id - --vault-name my-vault --
body archive.zip
```

Output:

```
{
    "archiveId": "kKB7ymWJVpPSwhGP6ycSOAekp9ZYe_--
zM_mw6k76ZFGEIWQX-ybtRDvc2VkPSDtfKmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hHO0bjbGehXTcApVud_wyDw",
    "checksum":
 "969fb39823836d81f0cc028195fcdbcbbe76cdde932d4646fa7de5f21e18aa67",
    "location": "/0123456789012/vaults/my-vault/archives/
kKB7ymWJVpPSwhGP6ycSOAekp9ZYe_--zM_mw6k76ZFGEIWQX-
ybtRDvc2VkPSDtfKmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hHO0bjbGehXTcApVud_wyDw"
}
```

Amazon Glacier memerlukan argumen ID akun saat melakukan operasi, tetapi Anda dapat menggunakan tanda hubung untuk menentukan akun yang sedang digunakan.

Untuk mengambil arsip yang diunggah, mulailah pekerjaan pengambilan dengan perintah aws glacier initiate-job.

- Untuk detail API, lihat UploadArchivedi Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.glacier.GlacierClient;
import software.amazon.awssdk.services.glacier.model.UploadArchiveRequest;
import software.amazon.awssdk.services.glacier.model.UploadArchiveResponse;
import software.amazon.awssdk.services.glacier.model.GlacierException;
import java.io.File;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.io.FileInputStream;
import java.io.IOException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class UploadArchive {

    static final int ONE_MB = 1024 * 1024;

    public static void main(String[] args) {
        final String usage = """

                Usage:   <strPath> <vaultName>\s

                Where:
                   strPath - The path to the archive to upload (for example, C:\
\AWS\\test.pdf).
                   vaultName - The name of the vault.
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String strPath = args[0];
```

```
        String vaultName = args[1];
        File myFile = new File(strPath);
        Path path = Paths.get(strPath);
        GlacierClient glacier = GlacierClient.builder()
                .region(Region.US_EAST_1)
                .build();

        String archiveId = uploadContent(glacier, path, vaultName, myFile);
        System.out.println("The ID of the archived item is " + archiveId);
        glacier.close();
    }

    public static String uploadContent(GlacierClient glacier, Path path, String
vaultName, File myFile) {
        // Get an SHA-256 tree hash value.
        String checkVal = computeSHA256(myFile);
        try {
            UploadArchiveRequest uploadRequest = UploadArchiveRequest.builder()
                    .vaultName(vaultName)
                    .checksum(checkVal)
                    .build();

            UploadArchiveResponse res = glacier.uploadArchive(uploadRequest,
path);
            return res.archiveId();

        } catch (GlacierException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }

    private static String computeSHA256(File inputFile) {
        try {
            byte[] treeHash = computeSHA256TreeHash(inputFile);
            System.out.printf("SHA-256 tree hash = %s\n", toHex(treeHash));
            return toHex(treeHash);

        } catch (IOException ioe) {
            System.err.format("Exception when reading from file %s: %s",
inputFile, ioe.getMessage());
            System.exit(-1);
```

```
        } catch (NoSuchAlgorithmException nsae) {
            System.err.format("Cannot locate MessageDigest algorithm for SHA-256:
%s", nsae.getMessage());
            System.exit(-1);
        }
        return "";
    }

    public static byte[] computeSHA256TreeHash(File inputFile) throws
IOException,
            NoSuchAlgorithmException {

        byte[][] chunkSHA256Hashes = getChunkSHA256Hashes(inputFile);
        return computeSHA256TreeHash(chunkSHA256Hashes);
    }

    /**
     * Computes an SHA256 checksum for each 1 MB chunk of the input file. This
     * includes the checksum for the last chunk, even if it's smaller than 1 MB.
     */
    public static byte[][] getChunkSHA256Hashes(File file) throws IOException,
            NoSuchAlgorithmException {

        MessageDigest md = MessageDigest.getInstance("SHA-256");
        long numChunks = file.length() / ONE_MB;
        if (file.length() % ONE_MB > 0) {
            numChunks++;
        }

        if (numChunks == 0) {
            return new byte[][] { md.digest() };
        }

        byte[][] chunkSHA256Hashes = new byte[(int) numChunks][];
        FileInputStream fileStream = null;

        try {
            fileStream = new FileInputStream(file);
            byte[] buff = new byte[ONE_MB];

            int bytesRead;
            int idx = 0;

            while ((bytesRead = fileStream.read(buff, 0, ONE_MB)) > 0) {
```

```
                md.reset();
                md.update(buff, 0, bytesRead);
                chunkSHA256Hashes[idx++] = md.digest();
            }

            return chunkSHA256Hashes;

        } finally {
            if (fileStream != null) {
                try {
                    fileStream.close();
                } catch (IOException ioe) {
                    System.err.printf("Exception while closing %s.\n %s",
file.getName(),
                            ioe.getMessage());
                }
            }
        }
    }

    /**
     * Computes the SHA-256 tree hash for the passed array of 1 MB chunk
     * checksums.
     */
    public static byte[] computeSHA256TreeHash(byte[][] chunkSHA256Hashes)
            throws NoSuchAlgorithmException {

        MessageDigest md = MessageDigest.getInstance("SHA-256");
        byte[][] prevLvlHashes = chunkSHA256Hashes;
        while (prevLvlHashes.length > 1) {
            int len = prevLvlHashes.length / 2;
            if (prevLvlHashes.length % 2 != 0) {
                len++;
            }

            byte[][] currLvlHashes = new byte[len][];
            int j = 0;
            for (int i = 0; i < prevLvlHashes.length; i = i + 2, j++) {

                // If there are at least two elements remaining.
                if (prevLvlHashes.length - i > 1) {

                    // Calculate a digest of the concatenated nodes.
                    md.reset();
```

```
                    md.update(prevLvlHashes[i]);
                    md.update(prevLvlHashes[i + 1]);
                    currLvlHashes[j] = md.digest();

                } else { // Take care of the remaining odd chunk
                    currLvlHashes[j] = prevLvlHashes[i];
                }
            }

            prevLvlHashes = currLvlHashes;
        }

        return prevLvlHashes[0];
    }

    /**
     * Returns the hexadecimal representation of the input byte array
     */
    public static String toHex(byte[] data) {
        StringBuilder sb = new StringBuilder(data.length * 2);
        for (byte datum : data) {
            String hex = Integer.toHexString(datum & 0xFF);

            if (hex.length() == 1) {
                // Append leading zero.
                sb.append("0");
            }
            sb.append(hex);
        }
        return sb.toString().toLowerCase();
    }
}
```

- Untuk detail API, lihat UploadArchivedi Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

Buat klien.

```
const { GlacierClient } = require("@aws-sdk/client-glacier");
// Set the AWS Region.
const REGION = "REGION";
//Set the Redshift Service Object
const glacierClient = new GlacierClient({ region: REGION });
export { glacierClient };
```

Unggah arsip.

```
// Load the SDK for JavaScript
import { UploadArchiveCommand } from "@aws-sdk/client-glacier";
import { glacierClient } from "./libs/glacierClient.js";

// Set the parameters
const vaultname = "VAULT_NAME"; // VAULT_NAME

// Create a new service object and buffer
const buffer = new Buffer.alloc(2.5 * 1024 * 1024); // 2.5MB buffer
const params = { vaultName: vaultname, body: buffer };

const run = async () => {
  try {
    const data = await glacierClient.send(new UploadArchiveCommand(params));
    console.log("Archive ID", data.archiveId);
    return data; // For unit tests.
  } catch (err) {
    console.log("Error uploading archive!", err);
  }
};
```

```
run();
```

- Untuk informasi selengkapnya, silakan lihat Panduan Developer AWS SDK for JavaScript.
- Untuk detail API, lihat UploadArchivedi Referensi AWS SDK for JavaScript API.

SDK untuk JavaScript (v2)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
// Load the SDK for JavaScript
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create a new service object and buffer
var glacier = new AWS.Glacier({ apiVersion: "2012-06-01" });
buffer = Buffer.alloc(2.5 * 1024 * 1024); // 2.5MB buffer

var params = { vaultName: "YOUR_VAULT_NAME", body: buffer };
// Call Glacier to upload the archive.
glacier.uploadArchive(params, function (err, data) {
  if (err) {
    console.log("Error uploading archive!", err);
  } else {
    console.log("Archive ID", data.archiveId);
  }
});
```

- Untuk informasi selengkapnya, silakan lihat Panduan Developer AWS SDK for JavaScript.
- Untuk detail API, lihat UploadArchivedi Referensi AWS SDK for JavaScript API.

PowerShell

Alat untuk PowerShell V4

Contoh 1: Mengunggah satu file ke brankas yang ditentukan, mengembalikan ID arsip dan checksum yang dihitung.

```
Write-GLCArchive -VaultName myvault -FilePath c:\temp\blue.bin
```

Output:

```
FilePath                    ArchiveId             Checksum
--------                    ---------             --------
C:\temp\blue.bin            o9O9jUUs...TTX-TpIhQJw 79f3e...f4395b
```

Contoh 2: Mengunggah konten hierarki folder ke brankas yang ditentukan di akun pengguna. Untuk setiap file yang diunggah, cmdlet memancarkan nama file, ID arsip yang sesuai, dan checksum arsip yang dihitung.

```
Write-GLCArchive -VaultName myvault -FolderPath . -Recurse
```

Output:

```
FilePath                    ArchiveId             Checksum
--------                    ---------             --------
C:\temp\blue.bin            o9O9jUUs...TTX-TpIhQJw 79f3e...f4395b
C:\temp\green.bin           qXAfOdSG...czo729UHXrw d50a1...9184b9
C:\temp\lum.bin             39aNifP3...q9nb8nZkFIg 28886...5c3e27
C:\temp\red.bin             vp7E6rU_...Ejk_HhjAxKA e05f7...4e34f5
C:\temp\Folder1\file1.txt   _eRINlip...5Sxy7dD2BaA d0d2a...c8a3ba
C:\temp\Folder2\file2.iso   -Ix3jlmu...iXiDh-XfOPA 7469e...3e86f1
```

- Untuk detail API, lihat UploadArchivedi Referensi AWS Tools for PowerShell Cmdlet (V4).

Alat untuk PowerShell V5

Contoh 1: Mengunggah satu file ke brankas yang ditentukan, mengembalikan ID arsip dan checksum yang dihitung.

```
Write-GLCArchive -VaultName myvault -FilePath c:\temp\blue.bin
```

Output:

```
FilePath                    ArchiveId             Checksum
--------                    ---------             --------
C:\temp\blue.bin            o9O9jUUs...TTX-TpIhQJw 79f3e...f4395b
```

Contoh 2: Mengunggah konten hierarki folder ke brankas yang ditentukan di akun pengguna. Untuk setiap file yang diunggah, cmdlet memancarkan nama file, ID arsip yang sesuai, dan checksum arsip yang dihitung.

```
Write-GLCArchive -VaultName myvault -FolderPath . -Recurse
```

Output:

```
FilePath                    ArchiveId             Checksum
--------                    ---------             --------
C:\temp\blue.bin            o9O9jUUs...TTX-TpIhQJw 79f3e...f4395b
C:\temp\green.bin           qXAfOdSG...czo729UHXrw d50a1...9184b9
C:\temp\lum.bin             39aNifP3...q9nb8nZkFIg 28886...5c3e27
C:\temp\red.bin             vp7E6rU_...Ejk_HhjAxKA e05f7...4e34f5
C:\temp\Folder1\file1.txt   _eRINlip...5Sxy7dD2BaA d0d2a...c8a3ba
C:\temp\Folder2\file2.iso   -Ix3jlmu...iXiDh-XfOPA 7469e...3e86f1
```

- Untuk detail API, lihat UploadArchivedi Referensi AWS Tools for PowerShell Cmdlet (V5).

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class GlacierWrapper:
    """Encapsulates Amazon S3 Glacier API operations."""

    def __init__(self, glacier_resource):
```

```
        """
        :param glacier_resource: A Boto3 Amazon S3 Glacier resource.
        """
        self.glacier_resource = glacier_resource


    @staticmethod
    def upload_archive(vault, archive_description, archive_file):
        """
        Uploads an archive to a vault.

        :param vault: The vault where the archive is put.
        :param archive_description: A description of the archive.
        :param archive_file: The archive file to put in the vault.
        :return: The uploaded archive.
        """
        try:
            archive = vault.upload_archive(
                archiveDescription=archive_description, body=archive_file
            )
            logger.info(
                "Uploaded %s with ID %s to vault %s.",
                archive_description,
                archive.id,
                vault.name,
            )
        except ClientError:
            logger.exception(
                "Couldn't upload %s to %s.", archive_description, vault.name
            )
            raise
        else:
            return archive
```

- Untuk detail API, lihat UploadArchivedi AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan Amazon Glacier dengan SDK AWS. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`UploadMultipartPart`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`UploadMultipartPart`.

CLI

AWS CLI

Perintah berikut mengunggah bagian 1 MiB (1024 x 1024 byte) pertama dari arsip:

```
aws glacier upload-multipart-part --body part1 --range 'bytes
 0-1048575/*' --account-id - --vault-name my-vault --upload-
id 19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_lR7vgFuJV6NtcV5zpsJ
```

Amazon Glacier memerlukan argumen ID akun saat melakukan operasi, tetapi Anda dapat menggunakan tanda hubung untuk menentukan akun yang sedang digunakan.

Parameter body mengambil jalur ke file bagian pada sistem file lokal. Parameter rentang mengambil rentang konten HTTP yang menunjukkan byte yang ditempati bagian dalam arsip yang lengkap. ID unggahan dikembalikan oleh `aws glacier initiate-multipart-upload` perintah dan juga dapat diperoleh dengan menggunakan`aws glacier list-multipart-uploads`.

Untuk informasi selengkapnya tentang unggahan multipart ke Amazon Glacier menggunakan CLI, lihat Menggunakan Amazon Glacier AWS di Panduan Pengguna CLI.AWS

- Untuk detail API, lihat UploadMultipartPartdi Referensi AWS CLI Perintah.

JavaScript

SDK untuk JavaScript (v2)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

Buat unggahan multipart dari potongan 1 megabyte dari objek Buffer.

```
// Create a new service object and some supporting variables
var glacier = new AWS.Glacier({ apiVersion: "2012-06-01" }),
  vaultName = "YOUR_VAULT_NAME",
  buffer = new Buffer(2.5 * 1024 * 1024), // 2.5MB buffer
  partSize = 1024 * 1024, // 1MB chunks,
  numPartsLeft = Math.ceil(buffer.length / partSize),
  startTime = new Date(),
  params = { vaultName: vaultName, partSize: partSize.toString() };

// Compute the complete SHA-256 tree hash so we can pass it
// to completeMultipartUpload request at the end
var treeHash = glacier.computeChecksums(buffer).treeHash;

// Initiate the multipart upload
console.log("Initiating upload to", vaultName);
// Call Glacier to initiate the upload.
glacier.initiateMultipartUpload(params, function (mpErr, multipart) {
  if (mpErr) {
    console.log("Error!", mpErr.stack);
    return;
  }
  console.log("Got upload ID", multipart.uploadId);

  // Grab each partSize chunk and upload it as a part
  for (var i = 0; i < buffer.length; i += partSize) {
    var end = Math.min(i + partSize, buffer.length),
      partParams = {
        vaultName: vaultName,
        uploadId: multipart.uploadId,
        range: "bytes " + i + "-" + (end - 1) + "/*",
        body: buffer.slice(i, end),
      };

    // Send a single part
    console.log("Uploading part", i, "=", partParams.range);
    glacier.uploadMultipartPart(partParams, function (multiErr, mData) {
      if (multiErr) return;
      console.log("Completed part", this.request.params.range);
      if (--numPartsLeft > 0) return; // complete only when all parts uploaded

      var doneParams = {
        vaultName: vaultName,
        uploadId: multipart.uploadId,
```

```
        archiveSize: buffer.length.toString(),
        checksum: treeHash, // the computed tree hash
      };

      console.log("Completing upload...");
      glacier.completeMultipartUpload(doneParams, function (err, data) {
        if (err) {
          console.log("An error occurred while uploading the archive");
          console.log(err);
        } else {
          var delta = (new Date() - startTime) / 1000;
          console.log("Completed upload in", delta, "seconds");
          console.log("Archive ID:", data.archiveId);
          console.log("Checksum:  ", data.checksum);
        }
      });
    });
  }
});
```

- Untuk informasi selengkapnya, silakan lihat Panduan Developer AWS SDK for JavaScript.
- Untuk detail API, lihat UploadMultipartPartdi Referensi AWS SDK for JavaScript API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan Amazon Glacier dengan SDK AWS. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Skenario untuk Amazon Glacier menggunakan AWS SDKs

Contoh kode berikut menunjukkan cara menerapkan skenario umum di Amazon AWS SDKs Glacier dengan. Skenario ini menunjukkan kepada Anda cara menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam Amazon Glacier atau digabungkan dengan yang lain. Layanan AWS Setiap skenario menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode.

Skenario menargetkan tingkat pengalaman menengah untuk membantu Anda memahami tindakan layanan dalam konteks.

Contoh

- Arsipkan file ke Amazon Glacier, dapatkan notifikasi, dan mulai pekerjaan menggunakan SDK AWS
- Dapatkan konten arsip Amazon Glacier dan hapus arsip menggunakan SDK AWS

Arsipkan file ke Amazon Glacier, dapatkan notifikasi, dan mulai pekerjaan menggunakan SDK AWS

Contoh kode berikut ini menunjukkan cara untuk melakukan:

- Buat lemari besi Amazon Glacier.
- Konfigurasikan vault untuk memublikasikan notifikasi ke topik Amazon SNS.
- Unggah file arsip ke brankas.
- Mulai tugas pengambilan arsip.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

Buat kelas yang membungkus operasi Amazon Glacier.

```
import argparse
import logging
import os
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)


class GlacierWrapper:
    """Encapsulates Amazon S3 Glacier API operations."""
```

```
    def __init__(self, glacier_resource):
        """
        :param glacier_resource: A Boto3 Amazon S3 Glacier resource.
        """
        self.glacier_resource = glacier_resource


    def create_vault(self, vault_name):
        """
        Creates a vault.

        :param vault_name: The name to give the vault.
        :return: The newly created vault.
        """
        try:
            vault = self.glacier_resource.create_vault(vaultName=vault_name)
            logger.info("Created vault %s.", vault_name)
        except ClientError:
            logger.exception("Couldn't create vault %s.", vault_name)
            raise
        else:
            return vault


    def list_vaults(self):
        """
        Lists vaults for the current account.
        """
        try:
            for vault in self.glacier_resource.vaults.all():
                logger.info("Got vault %s.", vault.name)
        except ClientError:
            logger.exception("Couldn't list vaults.")
            raise


    @staticmethod
    def upload_archive(vault, archive_description, archive_file):
        """
        Uploads an archive to a vault.

        :param vault: The vault where the archive is put.
        :param archive_description: A description of the archive.
        :param archive_file: The archive file to put in the vault.
```

```
        :return: The uploaded archive.
        """
        try:
            archive = vault.upload_archive(
                archiveDescription=archive_description, body=archive_file
            )
            logger.info(
                "Uploaded %s with ID %s to vault %s.",
                archive_description,
                archive.id,
                vault.name,
            )
        except ClientError:
            logger.exception(
                "Couldn't upload %s to %s.", archive_description, vault.name
            )
            raise
        else:
            return archive


    @staticmethod
    def initiate_archive_retrieval(archive):
        """
        Initiates an archive retrieval job. Standard retrievals typically
complete
        within 3–5 hours. When the job completes, you can get the archive
contents
        by calling get_output().

        :param archive: The archive to retrieve.
        :return: The archive retrieval job.
        """
        try:
            job = archive.initiate_archive_retrieval()
            logger.info("Started %s job with ID %s.", job.action, job.id)
        except ClientError:
            logger.exception("Couldn't start job on archive %s.", archive.id)
            raise
        else:
            return job


    @staticmethod
```

```
    def list_jobs(vault, job_type):
        """
        Lists jobs by type for the specified vault.

        :param vault: The vault to query.
        :param job_type: The type of job to list.
        :return: The list of jobs of the requested type.
        """
        job_list = []
        try:
            if job_type == "all":
                jobs = vault.jobs.all()
            elif job_type == "in_progress":
                jobs = vault.jobs_in_progress.all()
            elif job_type == "completed":
                jobs = vault.completed_jobs.all()
            elif job_type == "succeeded":
                jobs = vault.succeeded_jobs.all()
            elif job_type == "failed":
                jobs = vault.failed_jobs.all()
            else:
                jobs = []
                logger.warning("%s isn't a type of job I can get.", job_type)
            for job in jobs:
                job_list.append(job)
                logger.info("Got %s %s job %s.", job_type, job.action, job.id)
        except ClientError:
            logger.exception("Couldn't get %s jobs from %s.", job_type,
vault.name)
            raise
        else:
            return job_list


    def set_notifications(self, vault, sns_topic_arn):
        """
        Sets an Amazon Simple Notification Service (Amazon SNS) topic as a target
        for notifications. Amazon S3 Glacier publishes messages to this topic for
        the configured list of events.

        :param vault: The vault to set up to publish notifications.
        :param sns_topic_arn: The Amazon Resource Name (ARN) of the topic that
                              receives notifications.
        :return: Data about the new notification configuration.
```

```
        """
        try:
            notification = self.glacier_resource.Notification("-", vault.name)
            notification.set(
                vaultNotificationConfig={
                    "SNSTopic": sns_topic_arn,
                    "Events": [
                        "ArchiveRetrievalCompleted",
                        "InventoryRetrievalCompleted",
                    ],
                }
            )
            logger.info(
                "Notifications will be sent to %s for events %s from %s.",
                notification.sns_topic,
                notification.events,
                notification.vault_name,
            )
        except ClientError:
            logger.exception(
                "Couldn't set notifications to %s on %s.", sns_topic_arn,
 vault.name
            )
            raise
        else:
            return notification
```

Panggil fungsi pada kelas pembungkus untuk membuat vault dan mengunggah file, lalu konfigurasikan vault untuk mempublikasikan notifikasi dan memulai pekerjaan untuk mengambil arsip.

```
def upload_demo(glacier, vault_name, topic_arn):
    """
    Shows how to:
    * Create a vault.
    * Configure the vault to publish notifications to an Amazon SNS topic.
    * Upload an archive.
    * Start a job to retrieve the archive.

    :param glacier: A Boto3 Amazon S3 Glacier resource.
    :param vault_name: The name of the vault to create.
```

```
    :param topic_arn: The ARN of an Amazon SNS topic that receives notification
of
                      Amazon S3 Glacier events.
    """
    print(f"\nCreating vault {vault_name}.")
    vault = glacier.create_vault(vault_name)
    print("\nList of vaults in your account:")
    glacier.list_vaults()
    print(f"\nUploading glacier_basics.py to {vault.name}.")
    with open("glacier_basics.py", "rb") as upload_file:
        archive = glacier.upload_archive(vault, "glacier_basics.py", upload_file)
    print(
        "\nStarting an archive retrieval request to get the file back from the "
        "vault."
    )
    glacier.initiate_archive_retrieval(archive)
    print("\nListing in progress jobs:")
    glacier.list_jobs(vault, "in_progress")
    print(
        "\nBecause Amazon S3 Glacier is intended for infrequent retrieval, an "
        "archive request with Standard retrieval typically completes within 3-5 "
        "hours."
    )
    if topic_arn:
        notification = glacier.set_notifications(vault, topic_arn)
        print(
            f"\nVault {vault.name} is configured to notify the "
            f"{notification.sns_topic} topic when {notification.events} "
            f"events occur. You can subscribe to this topic to receive "
            f"a message when the archive retrieval completes.\n"
        )
    else:
        print(
            f"\nVault {vault.name} is not configured to notify an Amazon SNS
topic "
            f"when the archive retrieval completes so wait a few hours."
        )
    print("\nRetrieve your job output by running this script with the --retrieve
flag.")
```

- Untuk detail API, lihat topik berikut di Referensi API AWS SDK untuk Python (Boto3).

- CreateVault
- InitiateJob
- ListJobs
- ListVaults
- SetVaultNotifications
- UploadArchive

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan Amazon Glacier dengan SDK AWS. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Dapatkan konten arsip Amazon Glacier dan hapus arsip menggunakan SDK AWS

Contoh kode berikut ini menunjukkan cara untuk melakukan:

- Buat daftar pekerjaan untuk lemari besi Amazon Glacier dan dapatkan status pekerjaan.
- Dapatkan output dari pekerjaan pengambilan arsip yang sudah selesai.
- Hapus arsip.
- Hapus lemari besi.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

Buat kelas yang membungkus operasi Amazon Glacier.

```
import argparse
import logging
import os
```

```
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)


class GlacierWrapper:
    """Encapsulates Amazon S3 Glacier API operations."""

    def __init__(self, glacier_resource):
        """
        :param glacier_resource: A Boto3 Amazon S3 Glacier resource.
        """
        self.glacier_resource = glacier_resource


    @staticmethod
    def list_jobs(vault, job_type):
        """
        Lists jobs by type for the specified vault.

        :param vault: The vault to query.
        :param job_type: The type of job to list.
        :return: The list of jobs of the requested type.
        """
        job_list = []
        try:
            if job_type == "all":
                jobs = vault.jobs.all()
            elif job_type == "in_progress":
                jobs = vault.jobs_in_progress.all()
            elif job_type == "completed":
                jobs = vault.completed_jobs.all()
            elif job_type == "succeeded":
                jobs = vault.succeeded_jobs.all()
            elif job_type == "failed":
                jobs = vault.failed_jobs.all()
            else:
                jobs = []
                logger.warning("%s isn't a type of job I can get.", job_type)
            for job in jobs:
                job_list.append(job)
                logger.info("Got %s %s job %s.", job_type, job.action, job.id)
        except ClientError:
```

```
            logger.exception("Couldn't get %s jobs from %s.", job_type,
vault.name)
            raise
        else:
            return job_list


    @staticmethod
    def get_job_output(job):
        """
        Gets the output of a job, such as a vault inventory or the contents of an
        archive.

        :param job: The job to get output from.
        :return: The job output, in bytes.
        """
        try:
            response = job.get_output()
            out_bytes = response["body"].read()
            logger.info("Read %s bytes from job %s.", len(out_bytes), job.id)
            if "archiveDescription" in response:
                logger.info(
                    "These bytes are described as '%s'",
response["archiveDescription"]
                )
        except ClientError:
            logger.exception("Couldn't get output for job %s.", job.id)
            raise
        else:
            return out_bytes


    @staticmethod
    def delete_archive(archive):
        """
        Deletes an archive from a vault.

        :param archive: The archive to delete.
        """
        try:
            archive.delete()
            logger.info(
                "Deleted archive %s from vault %s.", archive.id,
archive.vault_name
```

```
            )
        except ClientError:
            logger.exception("Couldn't delete archive %s.", archive.id)
            raise


    @staticmethod
    def delete_vault(vault):
        """
        Deletes a vault.

        :param vault: The vault to delete.
        """
        try:
            vault.delete()
            logger.info("Deleted vault %s.", vault.name)
        except ClientError:
            logger.exception("Couldn't delete vault %s.", vault.name)
            raise
```

Panggil fungsi pada kelas pembungkus untuk mendapatkan konten arsip dari pekerjaan yang sudah selesai, lalu hapus arsip.

```
def retrieve_demo(glacier, vault_name):
    """
    Shows how to:
    * List jobs for a vault and get job status.
    * Get the output of a completed archive retrieval job.
    * Delete an archive.
    * Delete a vault.

    :param glacier: A Boto3 Amazon S3 Glacier resource.
    :param vault_name: The name of the vault to query for jobs.
    """
    vault = glacier.glacier_resource.Vault("-", vault_name)
    try:
        vault.load()
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceNotFoundException":
            print(
```

```
                f"\nVault {vault_name} doesn't exist. You must first run this
script "
                f"with the --upload flag to create the vault."
            )
            return
        else:
            raise

    print(f"\nGetting completed jobs for {vault.name}.")
    jobs = glacier.list_jobs(vault, "completed")
    if not jobs:
        print("\nNo completed jobs found. Give it some time and try again
later.")
        return

    retrieval_job = None
    for job in jobs:
        if job.action == "ArchiveRetrieval" and job.status_code == "Succeeded":
            retrieval_job = job
            break
    if retrieval_job is None:
        print(
            "\nNo ArchiveRetrieval jobs found. Give it some time and try again "
            "later."
        )
        return

    print(f"\nGetting output from job {retrieval_job.id}.")
    archive_bytes = glacier.get_job_output(retrieval_job)
    archive_str = archive_bytes.decode("utf-8")
    print("\nGot archive data. Printing the first 10 lines.")
    print(os.linesep.join(archive_str.split(os.linesep)[:10]))

    print(f"\nDeleting the archive from {vault.name}.")
    archive = glacier.glacier_resource.Archive(
        "-", vault.name, retrieval_job.archive_id
    )
    glacier.delete_archive(archive)

    print(f"\nDeleting {vault.name}.")
    glacier.delete_vault(vault)
```

- Untuk detail API, lihat topik berikut di Referensi API AWS SDK untuk Python (Boto3).
  - DeleteArchive
  - DeleteVault
  - GetJobOutput
  - ListJobs

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan Amazon Glacier dengan SDK AWS. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Keamanan di Amazon Glacier

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 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 Glacier (Amazon Glacier), lihat Layanan dalam Lingkup berdasarkan Program Kepatuhan.AWS
- Keamanan di 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.

Dokumentasi ini akan membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan Amazon Glacier. Topik berikut menunjukkan cara mengonfigurasi Amazon Glacier 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 Glacier Anda.

Topik

- Perlindungan Data di Amazon Glacier
- Identity and Access Management untuk Amazon Glacier
- Penebangan dan Pemantauan di Amazon Glacier
- Validasi Kepatuhan untuk Amazon Glacier
- Ketahanan di Amazon Glacier
- Keamanan Infrastruktur di Amazon Glacier

Perlindungan Data di Amazon Glacier

Amazon Glacier (Amazon Glacier) menyediakan penyimpanan cloud yang sangat tahan lama untuk pengarsipan data dan pencadangan jangka panjang. Amazon Glacier dirancang untuk memberikan daya tahan 99,999999999 persen dan memberikan kemampuan keamanan dan kepatuhan komprehensif yang dapat membantu Anda memenuhi persyaratan peraturan yang ketat. Amazon Glacier menyimpan data secara berlebihan di beberapa Availability Zone (AZ) dan di AWS beberapa perangkat dalam setiap AZ. Untuk meningkatkan daya tahan, Amazon Glacier secara serempak menyimpan data Anda di AZs beberapa sebelum mengonfirmasi unggahan yang berhasil.

Untuk informasi selengkapnya tentang infrastruktur cloud AWS global, lihat Infrastruktur Global.

Untuk tujuan perlindungan data, kami menyarankan Anda untuk melindungi Akun AWS kredensil dan hanya memberikan izin kepada pengguna, grup, atau peran individu 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 lebih lanjut tentang titik akhir FIPS yang tersedia, lihat Standar Pemrosesan Informasi Federal (FIPS) 140-2.

Topik

- Enkripsi Data
- Manajemen Kunci
- Privasi Lalu Lintas Kerja Internet

Enkripsi Data

Perlindungan data mengacu pada perlindungan data saat dalam perjalanan (saat bepergian ke dan dari Amazon Glacier) dan saat istirahat (saat AWS disimpan di pusat data). Anda dapat melindungi data dalam perjalanan yang diunggah langsung ke Amazon Glacier menggunakan Secure Sockets Layer (SSL) atau enkripsi sisi klien.

Anda juga dapat mengakses Amazon Glacier melalui Amazon S3. Amazon S3 mendukung konfigurasi siklus hidup pada bucket Amazon S3, yang memungkinkan Anda mentransisikan objek ke kelas penyimpanan Amazon Glacier untuk arsip. Data dalam transit antara Amazon S3 dan Amazon Glacier melalui kebijakan siklus hidup dienkripsi menggunakan SSL.

Data saat istirahat yang disimpan di Amazon Glacier secara otomatis dienkripsi sisi server menggunakan Standar Enkripsi Lanjutan 256-bit (AES-256) dengan kunci yang dikelola oleh. AWS Jika Anda lebih suka mengelola kunci Anda sendiri, Anda juga dapat menggunakan enkripsi sisi klien sebelum menyimpan data di Amazon Glacier. Untuk informasi selengkapnya tentang cara mengatur enkripsi default untuk Amazon S3, lihat Enkripsi Default Amazon S3 di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Manajemen Kunci

Enkripsi sisi server menangani enkripsi data saat istirahat—yaitu, Amazon Glacier mengenkripsi data Anda saat menuliskannya ke pusat datanya dan mendekripsi untuk Anda saat Anda mengaksesnya. Selama Anda mengautentikasi permintaan Anda dan memiliki izin akses, tidak ada perbedaan dalam cara Anda mengakses data terenkripsi atau tidak terenkripsi.

Data saat istirahat yang disimpan di Amazon Glacier secara otomatis dienkripsi sisi server menggunakan AES-256, menggunakan kunci yang dikelola oleh. AWS Sebagai perlindungan tambahan, AWS mengenkripsi kunci itu sendiri dengan kunci root yang kami putar secara teratur.

Privasi Lalu Lintas Kerja Internet

Akses ke Amazon Glacier melalui jaringan melalui dipublikasikan. AWS APIs Klien harus mendukung Keamanan Lapisan Pengangkutan (TLS) 1.2. Kami merekomendasikan TLS 1.3 atau versi yang lebih baru. 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 access key ID dan secret access key yang terkait dengan IAM utama, atau Anda dapat menggunakan AWS Security Token Service (AWS STS) guna membuat kredensial keamanan sementara untuk menandatangani permintaan.

Titik Akhir VPC

Titik akhir virtual private cloud (VPC) memungkinkan Anda menghubungkan VPC Anda secara pribadi ke layanan yang didukung AWS dan layanan titik akhir VPC yang didukung AWS PrivateLink tanpa memerlukan gateway internet, perangkat NAT, koneksi VPN, atau koneksi. AWS Direct Connect Meskipun Amazon Glacier tidak mendukung titik akhir VPC secara langsung, Anda dapat memanfaatkan Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) VPC endpoint jika Anda mengakses Amazon Glacier sebagai tingkat penyimpanan yang terintegrasi dengan Amazon S3.

Untuk informasi selengkapnya tentang konfigurasi siklus hidup Amazon S3 dan objek transisi ke kelas penyimpanan Amazon Glacier, lihat Objek Manajemen dan Transisi Siklus Hidup Objek di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon. Untuk informasi selengkapnya tentang VPC endpoint, lihat VPC Endpoint dalam Panduan Pengguna Amazon VPC.

Identity and Access Management untuk Amazon Glacier

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 diberi wewenang (memiliki izin) untuk menggunakan sumber daya Amazon Glacier. IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

- Audiens
- Mengautentikasi dengan identitas
- Mengelola akses menggunakan kebijakan
- Bagaimana Amazon Glacier bekerja dengan IAM
- Contoh kebijakan berbasis identitas untuk Amazon Glacier
- Contoh kebijakan berbasis sumber daya untuk Amazon Glacier
- Memecahkan masalah identitas dan akses Amazon Glacier
- Referensi Izin API

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda berdasarkan peran Anda:

- Pengguna layanan - minta izin dari administrator Anda jika Anda tidak dapat mengakses fitur (lihat Memecahkan masalah identitas dan akses Amazon Glacier)
- Administrator layanan - tentukan akses pengguna dan mengirimkan permintaan izin (lihat Bagaimana Amazon Glacier bekerja dengan IAM)
- Administrator IAM - tulis kebijakan untuk mengelola akses (lihat Contoh kebijakan berbasis identitas untuk Amazon Glacier)

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensi identitas Anda. Anda harus diautentikasi sebagai Pengguna root akun AWS, pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk sebagai identitas federasi menggunakan kredensional dari sumber identitas seperti AWS IAM Identity Center (Pusat Identitas IAM), autentikasi masuk tunggal, atau kredensional. Google/Facebook Untuk informasi selengkapnya tentang cara masuk, lihat Cara masuk ke Akun AWS Anda dalam Panduan Pengguna AWS Sign-In .

Untuk akses terprogram, AWS sediakan SDK dan CLI untuk menandatangani permintaan secara kriptografis. Untuk informasi selengkapnya, lihat AWS Signature Version 4 untuk permintaan API dalam Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang disebut pengguna Akun AWS root yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Untuk tugas yang memerlukan kredensial pengguna root, lihat Tugas yang memerlukan kredensial pengguna root dalam Panduan Pengguna IAM.

Identitas terfederasi

Sebagai praktik terbaik, mewajibkan pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS menggunakan kredensi sementara.

Identitas federasi adalah pengguna dari direktori perusahaan Anda, penyedia identitas web, atau Directory Service yang mengakses Layanan AWS menggunakan kredensi dari sumber identitas. Identitas terfederasi mengambil peran yang memberikan kredensial sementara.

Untuk manajemen akses terpusat, kami menyarankan AWS IAM Identity Center. Untuk informasi selengkapnya, lihat Apa itu Pusat Identitas IAM? dalam Panduan Pengguna AWS IAM Identity Center .

Pengguna dan grup IAM

Pengguna IAM adalah identitas dengan izin khusus untuk satu orang atau aplikasi. Sebaiknya gunakan kredensial sementara alih-alih pengguna IAM dengan kredensial jangka panjang. Untuk

informasi selengkapnya, lihat Mewajibkan pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses AWS menggunakan kredensi sementara di Panduan Pengguna IAM.

Grup IAM menentukan kumpulan pengguna IAM dan mempermudah pengelolaan izin untuk pengguna dalam jumlah besar. Untuk mempelajari selengkapnya, lihat Kasus penggunaan untuk pengguna IAM dalam Panduan Pengguna IAM.

Peran IAM

Peran IAM adalah identitas dengan izin khusus yang menyediakan kredensial sementara. Anda dapat mengambil peran dengan beralih dari pengguna ke peran IAM (konsol) atau dengan memanggil operasi AWS CLI atau AWS API. Untuk informasi selengkapnya, lihat Metode untuk mengambil peran dalam Panduan Pengguna IAM.

Peran IAM berguna untuk akses pengguna terfederasi, izin pengguna IAM sementara, akses lintas akun, akses lintas layanan, dan aplikasi yang berjalan di Amazon EC2. Untuk informasi selengkapnya, lihat Akses sumber daya lintas akun di IAM dalam Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan menentukan izin saat dikaitkan dengan identitas atau sumber daya. AWS mengevaluasi kebijakan ini ketika kepala sekolah membuat permintaan. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang dokumen kebijakan JSON, lihat Gambaran umum kebijakan JSON dalam Panduan Pengguna IAM.

Menggunakan kebijakan, administrator menentukan siapa yang memiliki akses ke apa dengan mendefinisikan principal mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Administrator IAM membuat kebijakan IAM dan menambahkannya ke peran, yang kemudian dapat diambil oleh pengguna. Kebijakan IAM mendefinisikan izin terlepas dari metode yang Anda gunakan untuk melakukan operasinya.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang Anda lampirkan ke identitas (pengguna, grup, atau peran). Kebijakan ini mengontrol tindakan apa yang bisa dilakukan oleh

identitas tersebut, terhadap sumber daya yang mana, dan dalam kondisi apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan yang dikelola pelanggan dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat berupa kebijakan inline (disematkan langsung ke dalam satu identitas) atau kebijakan terkelola (kebijakan mandiri yang dilampirkan pada banyak identitas). Untuk mempelajari cara memilih antara kebijakan terkelola dan kebijakan inline, lihat Pilih antara kebijakan terkelola 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. Contohnya termasuk 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. Anda harus menentukan principal dalam kebijakan berbasis sumber daya.

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.

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang dapat menetapkan izin maksimum yang diberikan oleh jenis kebijakan yang lebih umum:

- Batasan izin – Menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM. Untuk informasi selengkapnya, lihat Batasan izin untuk entitas IAM dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCPs) — Tentukan izin maksimum untuk organisasi atau unit organisasi di AWS Organizations. Untuk informasi selengkapnya, lihat Kebijakan kontrol layanan dalam Panduan Pengguna AWS Organizations .
- Kebijakan kontrol sumber daya (RCPs) — Tetapkan izin maksimum yang tersedia untuk sumber daya di akun Anda. Untuk informasi selengkapnya, lihat Kebijakan kontrol sumber daya (RCPs) di Panduan AWS Organizations Pengguna.
- Kebijakan sesi – Kebijakan lanjutan yang diteruskan sebagai parameter saat membuat sesi sementara untuk peran atau pengguna terfederasi. 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 Glacier bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses ke Amazon Glacier, pelajari fitur IAM apa yang tersedia untuk digunakan dengan Amazon Glacier.

Fitur IAM yang dapat Anda gunakan dengan Amazon Glacier

Fitur IAM	Dukungan Amazon Glacier
Kebijakan berbasis identitas	Ya
Kebijakan berbasis sumber daya	Ya
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya
kunci-kunci persyaratan kebijakan (spesifik layanan)	Ya
ACLs	Tidak
ABAC (tanda dalam kebijakan)	Tidak
Kredensial sementara	Ya
Izin principal	Tidak
Peran layanan	Tidak
Peran terkait layanan	Tidak

Untuk mendapatkan tampilan tingkat tinggi tentang cara kerja Amazon Glacier AWS dan layanan lainnya dengan sebagian besar fitur IAM AWS , lihat layanan yang bekerja dengan IAM di Panduan Pengguna IAM.

Kebijakan berbasis identitas untuk Amazon Glacier

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. 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 Glacier

Untuk melihat contoh kebijakan berbasis identitas Amazon Glacier, lihat. Contoh kebijakan berbasis identitas untuk Amazon Glacier

Kebijakan berbasis sumber daya dalam Amazon Glacier

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 principal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan principal 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 principal dalam kebijakan berbasis sumber daya. Untuk informasi selengkapnya, lihat Akses sumber daya lintas akun di IAM dalam Panduan Pengguna IAM.

Layanan Amazon Glacier hanya mendukung satu jenis kebijakan berbasis sumber daya yang disebut kebijakan vault, yang dilampirkan ke brankas. Kebijakan ini menentukan prinsipal mana yang dapat melakukan tindakan pada brankas.

Kebijakan Amazon Glacier vault mengelola izin dengan cara berikut:

- Mengelola izin pengguna di akun menggunakan kebijakan vault tunggal, bukan lebih dari satu kebijakan pengguna individual.
- Mengelola izin lintas akun sebagai alternatif untuk menggunakan IAM role.

Contoh kebijakan berbasis sumber daya dalam Amazon Glacier

Untuk melihat contoh kebijakan berbasis sumber daya Amazon Glacier, lihat. Contoh kebijakan berbasis sumber daya untuk Amazon Glacier

Tindakan kebijakan untuk Amazon Glacier

Mendukung tindakan kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Sertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Untuk melihat daftar tindakan Amazon Glacier, lihat Tindakan yang ditentukan oleh Amazon Glacier di Referensi Otorisasi Layanan.

Tindakan kebijakan di Amazon Glacier menggunakan awalan berikut sebelum tindakan:

```
glacier
```

Untuk menetapkan secara spesifik beberapa tindakan dalam satu pernyataan, pisahkan tindakan tersebut dengan koma.

```
"Action": [
            "glacier:CreateVault",
            "glacier:DescribeVault",
            "glacier:ListVaults"
          ]
```

Anda juga dapat menentukan beberapa tindakan menggunakan wildcard (*). Sebagai contoh, untuk menentukan semua tindakan yang dimulai dengan kata `Describe`, sertakan tindakan berikut:

```
"Action": "glacier:GetVault*"
```

Untuk melihat contoh kebijakan berbasis identitas Amazon Glacier, lihat. Contoh kebijakan berbasis identitas untuk Amazon Glacier

Sumber daya kebijakan untuk Amazon Glacier

Mendukung sumber daya kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Praktik terbaiknya, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

```
"Resource": "*"
```

Untuk melihat daftar jenis sumber daya Amazon Glacier beserta ARNs jenisnya, lihat Sumber Daya yang ditentukan oleh Amazon Glacier dalam Referensi Otorisasi Layanan. Untuk mempelajari tindakan mana yang dapat Anda tentukan ARN dari setiap sumber daya, lihat Tindakan yang ditentukan oleh Amazon Glacier.

Di Amazon Glacier, sumber daya utamanya adalah lemari besi. Amazon Glacier hanya mendukung kebijakan di tingkat brankas. Artinya, dalam kebijakan IAM, `Resource` nilai yang Anda tentukan

dapat berupa brankas tertentu atau sekumpulan vault di Wilayah tertentu. AWS Amazon Glacier tidak mendukung izin tingkat arsip.

Untuk semua tindakan Amazon Glacier`Resource`, tentukan brankas tempat Anda ingin memberikan izin. Resource ini memiliki Nama Sumber Daya Amazon (ARNs) unik yang terkait dengannya seperti yang ditunjukkan pada tabel berikut, dan Anda dapat menggunakan karakter wildcard (*) di ARN untuk mencocokkan nama vault yang dimulai dengan awalan yang sama.

Amazon Glacier menyediakan serangkaian operasi untuk bekerja dengan sumber daya Amazon Glacier. Untuk informasi tentang operasi yang tersedia, lihat Referensi API untuk Amazon Glacier.

Beberapa tindakan Amazon Glacier API mendukung beberapa sumber daya. Misalnya, `glacier:AddTagsToVault` mengakses examplevault1 dan examplevault2, jadi prinsipal harus memiliki izin untuk mengakses kedua sumber daya. Untuk menentukan beberapa sumber daya dalam satu pernyataan, pisahkan ARNs dengan koma.

```
"Resource": [
""arn:aws:glacier:us-west-2:123456789012:vaults/examplevault1",",
""arn:aws:glacier:us-west-2:123456789012:vaults/examplevault2","
]
```

Kunci kondisi kebijakan untuk Amazon Glacier

Mendukung kunci kondisi kebijakan khusus layanan: Yes

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, principal dapat melakukan tindakan pada suatu sumber daya, dan dalam suatu syarat.

Elemen `Condition` menentukan ketika pernyataan dieksekusi berdasarkan kriteria yang ditetapkan. 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. Untuk melihat semua kunci kondisi AWS global, lihat kunci konteks kondisi AWS global di Panduan Pengguna IAM.

Untuk melihat daftar kunci kondisi Amazon Glacier, lihat Kunci kondisi untuk Amazon Glacier di Referensi Otorisasi Layanan. Untuk mempelajari tindakan dan sumber daya yang dapat Anda gunakan kunci kondisi, lihat Tindakan yang ditentukan oleh Amazon Glacier.

Untuk contoh menggunakan kunci kondisi khusus gletser, lihat. Kebijakan Vault Lock

ACLs di Amazon Glacier

Mendukung ACLs: Tidak

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.

ABAC dengan Amazon Glacier

Mendukung ABAC (tag dalam kebijakan): Tidak

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang menentukan izin berdasarkan atribut tanda. Anda dapat melampirkan tag ke entitas dan AWS sumber daya IAM, lalu merancang kebijakan ABAC untuk mengizinkan operasi saat tag prinsipal cocok dengan tag pada sumber daya.

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.

Menggunakan kredensi sementara dengan Amazon Glacier

Mendukung kredensial sementara: Ya

Kredensi sementara menyediakan akses jangka pendek ke AWS sumber daya dan secara otomatis dibuat saat Anda menggunakan federasi atau beralih peran. 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 dan Layanan AWS yang berfungsi dengan IAM dalam Panduan Pengguna IAM.

Izin utama lintas layanan untuk Amazon Glacier

Mendukung sesi akses maju (FAS): Tidak

Sesi akses teruskan (FAS) menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat Sesi akses terusan.

Peran layanan untuk Amazon Glacier

Mendukung peran layanan: Tidak

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 Glacier. Edit peran layanan hanya jika Amazon Glacier memberikan panduan untuk melakukannya.

Peran terkait layanan untuk Amazon Glacier

Mendukung peran terkait layanan: Tidak

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.

Untuk detail tentang pembuatan atau manajemen peran terkait layanan, lihat Layanan AWS yang berfungsi dengan IAM. Cari layanan dalam tabel yang memiliki `Yes` di kolom Peran terkait layanan. Pilih tautan Ya untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

Contoh kebijakan berbasis identitas untuk Amazon Glacier

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi sumber daya Amazon Glacier. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM.

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 Glacier, termasuk format untuk setiap jenis sumber daya, lihat Kunci tindakan, sumber daya, dan kondisi untuk Amazon Glacier dalam Referensi Otorisasi Layanan. ARNs

Berikut ini adalah contoh kebijakan yang memberikan izin untuk tiga tindakan terkait Amazon Glacier vault (, dan`glacier:ListVaults`) pada sumber daya`glacier:DescribeVault`, menggunakan Nama Sumber Daya Amazon `glacier:CreateVault` (ARN) yang mengidentifikasi semua brankas di Wilayah. `us-west-2` AWS ARNs mengidentifikasi AWS sumber daya secara unik. Untuk informasi lebih lanjut tentang ARNs penggunaan dengan Amazon Glacier, lihat. Sumber daya kebijakan untuk Amazon Glacier

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
         "glacier:CreateVault",
         "glacier:DescribeVault",
         "glacier:ListVaults"
         ],
         "Resource": "arn:aws:glacier:us-west-2:123456789012:vaults/*"
      }
   ]
}
```

Kebijakan tersebut memberikan izin untuk membuat, mencantumkan, dan mendapatkan deskripsi vauld di Wilayah `us-west-2`. Karakter wildcard (*) pada akhir ARN berarti pernyataan ini dapat cocok dengan nama vault.

> **Important**
>
> Ketika Anda memberikan izin untuk membuat vault menggunakan operasi `glacier:CreateVault`, Anda harus menentukan karakter wildcard (*) karena Anda tidak mengetahui nama vault hingga setelah Anda membuat vault.

Topik

- Praktik terbaik kebijakan
- Menggunakan konsol Amazon Glacier
- Mengizinkan pengguna melihat izin mereka sendiri
- Contoh Kebijakan yang Dikelola Pelanggan

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya Amazon Glacier 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 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.

Menggunakan konsol Amazon Glacier

Untuk mengakses konsol Amazon Glacier, Anda harus memiliki set izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang sumber daya Amazon Glacier di Anda. Akun AWS Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana mestinya untuk entitas (pengguna atau peran) dengan kebijakan tersebut.

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebagai gantinya, izinkan akses hanya ke tindakan yang sesuai dengan operasi API yang coba mereka lakukan.

Konsol Amazon Glacier menyediakan lingkungan terintegrasi bagi Anda untuk membuat dan mengelola brankas Amazon Glacier. Minimal identitas IAM yang Anda buat harus diberikan izin untuk `glacier:ListVaults` tindakan untuk melihat konsol Amazon Glacier seperti yang ditunjukkan pada contoh berikut.

JSON

```
{
      "Version":"2012-10-17",
      "Statement": [
         {
            "Action": [
            "glacier:ListVaults"
            ],
            "Effect": "Allow",
            "Resource": "*"
         }
      ]
   }
```

AWS mengatasi banyak kasus penggunaan umum dengan menyediakan kebijakan IAM mandiri yang dibuat dan dikelola oleh. AWS Kebijakan terkelola memberikan izin yang diperlukan untuk kasus penggunaan umum sehingga Anda tidak perlu menyelidiki izin apa yang diperlukan. Untuk informasi selengkapnya, lihat Kebijakan Terkelola AWS dalam Panduan Pengguna IAM.

Kebijakan AWS terkelola berikut, yang dapat Anda lampirkan ke pengguna di akun Anda, khusus untuk Amazon Glacier:

- AmazonGlacierReadOnlyAccess— Hibah hanya membaca akses ke Amazon Konsol Manajemen AWS Glacier melalui.
- AmazonGlacierFullAccess— Memberikan akses penuh ke Amazon Konsol Manajemen AWS Glacier melalui.

Anda juga dapat membuat kebijakan IAM kustom Anda sendiri untuk mengizinkan izin untuk tindakan dan sumber daya Amazon Glacier API. Anda dapat melampirkan kebijakan kustom ini ke peran IAM kustom yang Anda buat untuk brankas Amazon Glacier Anda.

Kedua kebijakan Dikelola Amazon AWS Glacier yang dibahas di bagian berikutnya memberikan izin untuk. `glacier:ListVaults`

Untuk informasi selengkapnya, lihat Menambah izin untuk pengguna dalam Panduan Pengguna IAM.

Mengizinkan pengguna melihat izin mereka sendiri

Contoh ini menunjukkan cara membuat kebijakan yang mengizinkan pengguna IAM melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau menggunakan API atau secara terprogram. AWS CLI AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
```

```
}
```

Contoh Kebijakan yang Dikelola Pelanggan

Di bagian ini, Anda dapat menemukan contoh kebijakan pengguna yang memberikan izin untuk berbagai tindakan Amazon Glacier. Kebijakan ini berfungsi saat Anda menggunakan Amazon Glacier REST API, SDKs Amazon, atau, AWS CLI jika berlaku, konsol manajemen Amazon Glacier.

> **Note**
>
> Semua contoh menggunakan Wilayah Barat AS (Oregon) (`us-west-2`) dan berisi akun fiktif. IDs

Contoh

- Contoh 1: Mengizinkan Pengguna Mengunduh Arsip dari Vault
- Contoh 2: Mengizinkan Pengguna Membuat Vault dan Mengonfigurasi Notifikasi
- Contoh 3: Mengizinkan Pengguna Mengunggah Arsip ke Vault Spesifik
- Contoh 4: Memberikan Izin Pengguna Penuh di Vault Spesifik

Contoh 1: Mengizinkan Pengguna Mengunduh Arsip dari Vault

Untuk mengunduh arsip, Anda memulai tugas untuk mengambil arsip terlebih dahulu. Setelah tugas pengambilan selesai, Anda dapat mengunduh data. Contoh kebijakan berikut memberikan izin untuk tindakan `glacier:InitiateJob` untuk memulai tugas (yang memungkinkan pengguna mengambil arsip atau inventaris vault dari vault), dan izin untuk tindakan `glacier:GetJobOutput` untuk mengunduh data yang diambil. Kebijakan ini juga memberikan izin untuk melakukan tindakan `glacier:DescribeJob` sehingga pengguna bisa mendapatkan status tugas. Untuk informasi selengkapnya, lihat Memulai Tugas (POST jobs).

Kebijakan memberikan izin ini pada vault bernama `examplevault`. Anda bisa mendapatkan ARN vault dari konsol Amazon Glacier, atau secara terprogram dengan memanggil tindakan atau API. Menjelaskan Vault (GET vault) Mencantumkan Vault (GET vaults)

Contoh 2: Mengizinkan Pengguna Membuat Vault dan Mengonfigurasi Notifikasi

Contoh kebijakan berikut memberikan izin untuk membuat vault di Wilayah us-west-2 seperti yang ditentukan dalam elemen `Resource` dan mengonfigurasi notifikasi. Untuk informasi selengkapnya

tentang bekerja dengan notifikasi, lihat Mengkonfigurasi Pemberitahuan Vault di Amazon Glacier. Kebijakan ini juga memberikan izin untuk mencantumkan vault di AWS Wilayah dan mendapatkan deskripsi vault tertentu.

> **Important**
>
> Ketika Anda memberikan izin untuk membuat vault menggunakan operasi `glacier:CreateVault`, Anda harus menentukan karakter wildcard (*) dalam nilai `Resource` karena Anda tidak mengetahui nama vault hingga setelah Anda membuat vault.

Contoh 3: Mengizinkan Pengguna Mengunggah Arsip ke Vault Spesifik

Contoh kebijakan berikut memberikan izin untuk mengunggah arsip ke vault spesifik di Wilayah us-west-2. Izin ini memungkinkan pengguna mengunggah arsip sekaligus menggunakan operasi API Mengunggah Arsip (POST archive) atau di sebagian menggunakan operasi API Memulai Unggahan Multipart (POST multipart-uploads).

Contoh 4: Memberikan Izin Pengguna Penuh di Vault Spesifik

Contoh kebijakan berikut memberikan izin untuk semua tindakan Amazon Glacier pada vault bernama. `examplevault`

Contoh kebijakan berbasis sumber daya untuk Amazon Glacier

Gudang Amazon Glacier dapat memiliki satu kebijakan akses vault dan satu kebijakan Vault Lock yang terkait dengannya. Kebijakan akses Amazon Glacier vault adalah kebijakan berbasis sumber daya yang dapat Anda gunakan untuk mengelola izin ke vault Anda. Kebijakan Vault Lock adalah kebijakan akses vault yang dapat dikunci. Setelah Anda mengunci kebijakan Vault Lock, kebijakan tidak dapat diubah. Anda dapat menggunakan Kebijakan Vault Lock untuk menerapkan kontrol kepatuhan.

Topik

- Kebijakan Akses Vault
- Kebijakan Vault Lock

Kebijakan Akses Vault

Kebijakan akses Amazon Glacier vault adalah kebijakan berbasis sumber daya yang dapat Anda gunakan untuk mengelola izin ke vault.

Anda dapat membuat satu kebijakan akses vault untuk setiap vault guna mengelola izin. Anda dapat memodifikasi izin dalam kebijakan akses vault kapan saja. Amazon Glacier juga mendukung kebijakan Vault Lock pada setiap brankas yang, setelah Anda menguncinya, tidak dapat diubah. Untuk informasi selengkapnya tentang bekerja dengan kebijakan Vault Lock, lihat Kebijakan Vault Lock.

Contoh

- Contoh 1: Berikan Izin Lintas Akun untuk Tindakan Amazon Glacier Tertentu
- Contoh 2: Memberikan Izin Lintas Akun untuk Operasi Hapus MFA

Contoh 1: Berikan Izin Lintas Akun untuk Tindakan Amazon Glacier Tertentu

Contoh kebijakan berikut memberikan izin lintas akun kepada dua Akun AWS untuk satu set operasi Amazon Glacier pada vault bernama. `examplevault`

> **Note**
>
> Akun yang memiliki vault ditagih untuk semua biaya yang terkait dengan vault. Semua permintaan, transfer data, dan biaya pengambilan yang dibuat oleh akun eksternal yang diizinkan ditagih ke akun yang memiliki vault.

Contoh 2: Memberikan Izin Lintas Akun untuk Operasi Hapus MFA

Anda dapat menggunakan otentikasi multi-faktor (MFA) untuk melindungi sumber daya Amazon Glacier Anda. Untuk memberikan tingkat keamanan tambahan, MFA mengharuskan pengguna membuktikan kepemilikan fisik perangkat MFA dengan memberikan kode MFA yang valid. Untuk informasi selengkapnya tentang mengonfigurasi akses MFA, lihat Mengonfigurasi Akses API yang Dilindungi MFA di Panduan Pengguna IAM.

Kebijakan contoh memberikan izin kredenal sementara untuk menghapus arsip dari vault bernama examplevault, asalkan permintaan tersebut diautentikasi dengan perangkat MFA. Akun AWS Kebijakan menggunakan kunci kondisi `aws:MultiFactorAuthPresent` untuk menentukan

persyaratan tambahan ini. Untuk informasi selengkapnya, lihat Kunci yang Tersedia untuk Kondisi di Panduan Pengguna IAM.

Kebijakan Vault Lock

Brankas Amazon Glacier (Amazon Glacier) dapat memiliki satu kebijakan akses brankas berbasis sumber daya dan satu kebijakan Vault Lock yang melekat padanya. Kebijakan Vault Lock adalah kebijakan akses vault yang dapat Anda kunci. Menggunakan kebijakan Vault Lock dapat membantu Anda menerapkan persyaratan peraturan dan kepatuhan. Amazon Glacier menyediakan serangkaian operasi API bagi Anda untuk mengelola kebijakan Vault Lock, lihat. Mengunci Vault dengan Menggunakan Amazon Glacier API

Sebagai contoh kebijakan Vault Lock, misalkan Anda diminta untuk menyimpan arsip selama satu tahun sebelum dapat menghapusnya. Untuk menerapkan persyaratan ini, Anda dapat membuat kebijakan Vault Lock yang menolak izin pengguna untuk menghapus arsip hingga arsip tersedia selama satu tahun. Anda dapat menguji kebijakan ini sebelum menguncinya. Setelah Anda mengunci kebijakan, kebijakan tidak dapat diubah. Untuk informasi selengkapnya tentang proses penguncian, lihat Kebijakan Vault Lock. Jika Anda ingin mengelola izin pengguna lain yang dapat diubah, Anda dapat menggunakan kebijakan akses vault (lihatKebijakan Akses Vault).

Anda dapat menggunakan Amazon Glacier API, Amazon, atau konsol SDKs Amazon AWS CLI Glacier untuk membuat dan mengelola kebijakan Vault Lock. Untuk daftar tindakan Amazon Glacier yang diizinkan untuk kebijakan berbasis sumber daya vault, lihat. Referensi Izin API

Contoh

- Contoh 1: Menolak Izin Penghapusan untuk Arsip Kurang Dari 365 Hari
- Contoh 2: Menolak Izin Penghapusan Berdasarkan Tanda

Contoh 1: Menolak Izin Penghapusan untuk Arsip Kurang Dari 365 Hari

Misalkan Anda memiliki persyaratan peraturan untuk menyimpan arsip hingga satu tahun sebelum Anda dapat menghapusnya. Anda dapat melaksanakan persyaratan tersebut dengan menerapkan kebijakan Vault Lock berikut. Kebijakan menolak tindakan `glacier:DeleteArchive` pada vault examplevault jika arsip yang dihapus kurang dari satu tahun. Kebijakan ini menggunakan kunci kondisi khusus Amazon Glacier `ArchiveAgeInDays` untuk menegakkan persyaratan retensi satu tahun.

Contoh 2: Menolak Izin Penghapusan Berdasarkan Tanda

Misalkan Anda memiliki aturan retensi berbasis waktu yang dapat menghapus arsip jika disimpan kurang dari satu tahun. Pada saat yang sama, misalkan Anda harus menempatkan pengamanan legal pada arsip Anda untuk mencegah penghapusan atau modifikasi selama waktu yang tidak terbatas selama penyelidikan legal. Dalam kasus ini, pengamanan legal diutamakan dari aturan retensi berbasis waktu yang ditentukan dalam kebijakan Vault Lock.

Untuk menempatkan dua aturan ini, kebijakan contoh berikut memiliki dua pernyataan:

- Pernyataan pertama menolak izin penghapusan untuk semua orang, mengunci vault. Kunci ini dilakukan menggunakan tanda `LegalHold`.
- Pernyataan kedua memberikan izin penghapusan ketika arsip kurang dari 365 hari. Namun, meskipun ketika arsip kurang dari 365 hari, tidak ada yang dapat menghapusnya ketika kondisi dalam pernyataan pertama terpenuhi.

Memecahkan masalah identitas dan akses Amazon Glacier

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan Amazon Glacier dan IAM.

Topik

- Saya tidak berwenang untuk melakukan tindakan di Amazon Glacier
- Saya tidak berwenang untuk melakukan iam: PassRole
- Saya ingin mengizinkan orang di luar saya Akun AWS mengakses sumber daya Amazon Glacier saya

Saya tidak berwenang untuk melakukan tindakan di Amazon Glacier

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 `glacier:`*`GetWidget`* rekaan.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 glacier: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 `glacier:`*`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 Glacier.

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 Glacier. 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 Glacier 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 mengetahui apakah Amazon Glacier mendukung fitur-fitur ini, lihat. Bagaimana Amazon Glacier 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.

Referensi Izin API

Ketika Anda mengatur Bagaimana Amazon Glacier bekerja dengan IAM dan menulis kebijakan izin yang dapat Anda lampirkan ke identitas IAM (kebijakan berbasis identitas) atau sumber daya (kebijakan berbasis sumber daya), Anda dapat menggunakan tabel berikut sebagai referensi. Daftar mencakup setiap operasi Amazon Glacier API, tindakan terkait yang dapat Anda berikan izin untuk melakukan tindakan, dan AWS sumber daya yang dapat Anda berikan izin.

Anda menentukan tindakan dalam elemen `Action` kebijakan, dan Anda menentukan nilai sumber daya di dalam elemen `Resource` kebijakan tersebut. Anda juga dapat menggunakan elemen `Condition` bahasa kebijakan IAM untuk menentukan kapan kebijakan harus berlaku.

Untuk menentukan tindakan, gunakan prefiks `glacier:` yang diikuti dengan nama operasi API (misalnya, `glacier:CreateVault`). Untuk sebagian besar tindakan Amazon Glacier`Resource`, adalah brankas tempat Anda ingin memberikan izin. Anda menentukan vault sebagai nilai `Resource` menggunakan ARN vault. Untuk menyatakan kondisi, Anda menggunakan kunci kondisi yang telah ditentukan sebelumnya. Untuk informasi selengkapnya, lihat Kebijakan berbasis sumber daya dalam Amazon Glacier.

Tabel berikut mencantumkan tindakan yang dapat digunakan dengan kebijakan berbasis identitas dan kebijakan berbasis sumber daya.

> **Note**
>
> Beberapa tindakan hanya dapat digunakan dengan kebijakan berbasis identitas. Tindakan ini ditandai dengan tanda bintang (*) setelah nama operasi API di kolom pertama.

Amazon Glacier API dan Izin yang Diperlukan untuk Tindakan

Membatalkan Unggahan Multipart (DELETE uploadID)

Izin yang Diperlukan (Tindakan API): `glacier:AbortMultipartUpload`

Sumber Daya: `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/vault-name`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/example*`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/*`

Kunci Kondisi Amazon Glacier:

Membatalkan Kunci Vault (DELETE lock-policy)

Izin yang Diperlukan (Tindakan API): `glacier:AbortVaultLock`

Sumber Daya:

Kunci Kondisi Amazon Glacier:

Menambahkan Tanda ke Vault (POST tag add)

Izin yang Diperlukan (Tindakan API):`glacier:AddTagsToVault`

Sumber Daya: `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/vault-name`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/example*`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/*`

Kunci Kondisi Amazon Glacier: `glacier:ResourceTag/`*`TagKey`*

Menyelesaikan Unggahan Multipart (POST uploadID)

Izin yang Diperlukan (Tindakan API):`glacier:CompleteMultipartUpload`

Sumber Daya: `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/vault-name`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/example*`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/*`

Kunci Kondisi Amazon Glacier: `glacier:ResourceTag/`*`TagKey`*

Menyelesaikan Kunci Vault (POST lockId)

Izin yang Diperlukan (Tindakan API):`glacier:CompleteVaultLock`

Sumber Daya:

Kunci Kondisi Amazon Glacier: `glacier:ResourceTag/`*`TagKey`*

Membuat Vault (PUT vault) *

Izin yang Diperlukan (Tindakan API):`glacier:CreateVault`

Sumber Daya:

Kunci Kondisi Amazon Glacier:

Menghapus Arsip (DELETE archive)

Izin yang Diperlukan (Tindakan API):`glacier:DeleteArchive`

Sumber Daya: `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/vault-name`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/example*`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/*`

Kunci Kondisi Amazon Glacier:, `glacier:ArchiveAgeInDays` `glacier:ResourceTag/`*`TagKey`*

Menghapus Vault (DELETE vault)

Izin yang Diperlukan (Tindakan API):`glacier:DeleteVault`

Sumber Daya: `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/vault-name`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/example*`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/*`

Kunci Kondisi Amazon Glacier: `glacier:ResourceTag/`*`TagKey`*

Menghapus Kebijakan Akses Vault (DELETE access-policy)

Izin yang Diperlukan (Tindakan API):`glacier:DeleteVaultAccessPolicy`

Sumber Daya: `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/vault-name`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/example*`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/*`

Kunci Kondisi Amazon Glacier: `glacier:ResourceTag/`*`TagKey`*

Menghapus Notifikasi Vault (DELETE notification-configuration)

Izin yang Diperlukan (Tindakan API):`glacier:DeleteVaultNotifications`

Sumber Daya: `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/vault-name`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/example*`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/*`

Kunci Kondisi Amazon Glacier: `glacier:ResourceTag/`*`TagKey`*

Mendeskripsikan Tugas (GET JobID)

Izin yang Diperlukan (Tindakan API):`glacier:DescribeJob`

Sumber Daya: `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/vault-name`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/example*`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/*`

Kunci Kondisi Amazon Glacier:

Menjelaskan Vault (GET vault)

Izin yang Diperlukan (Tindakan API):`glacier:DescribeVault`

Sumber Daya: `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/vault-name`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/example*`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/*`

Kunci Kondisi Amazon Glacier:

Mendapatkan Kebijakan Pengambilan Data (GET policy) *

Izin yang Diperlukan (Tindakan API):`glacier:GetDataRetrievalPolicy`

Sumber Daya: `arn:aws:glacier:`*`region`*`:`*`account-id`*`:policies/retrieval-limit-policy`

Kunci Kondisi Amazon Glacier:

Mendapatkan Output Tugas (GET output)

Izin yang Diperlukan (Tindakan API):`glacier:GetJobOutput`

Sumber Daya: `arn:aws:glacier:region:account-id:vaults/vault-name`, `arn:aws:glacier:region:account-id:vaults/example*`, `arn:aws:glacier:region:account-id:vaults/*`

Kunci Kondisi Amazon Glacier:

Mendapatkan Kebijakan Akses Vault (GET access-policy)

Izin yang Diperlukan (Tindakan API):`glacier:GetVaultAccessPolicy`

Sumber Daya: `arn:aws:glacier:region:account-id:vaults/vault-name`, `arn:aws:glacier:region:account-id:vaults/example*`, `arn:aws:glacier:region:account-id:vaults/*`

Kunci Kondisi Amazon Glacier:

Mendapatkan Kunci Vault (GET lock-policy)

Izin yang Diperlukan (Tindakan API):`glacier:GetVaultLock`

Sumber Daya: `arn:aws:glacier:region:account-id:vaults/vault-name`, `arn:aws:glacier:region:account-id:vaults/example*`, `arn:aws:glacier:region:account-id:vaults/*`

Kunci Kondisi Amazon Glacier:

Mendapatkan Notifikasi Vault (GET notification-configuration)

Izin yang Diperlukan (Tindakan API):`glacier:GetVaultNotifications`

Sumber Daya: `arn:aws:glacier:region:account-id:vaults/vault-name`, `arn:aws:glacier:region:account-id:vaults/example*`, `arn:aws:glacier:region:account-id:vaults/*`

Kunci Kondisi Amazon Glacier:

Memulai Tugas (POST jobs)

Izin yang Diperlukan (Tindakan API):`glacier:InitiateJob`

Sumber Daya: `arn:aws:glacier:region:account-id:vaults/vault-name`, `arn:aws:glacier:region:account-id:vaults/example*`, `arn:aws:glacier:region:account-id:vaults/*`

Kunci Kondisi Amazon Glacier:, `glacier:ArchiveAgeInDays` `glacier:ResourceTag/`*`TagKey`*

Memulai Unggahan Multipart (POST multipart-uploads)

Izin yang Diperlukan (Tindakan API):`glacier:InitiateMultipartUpload`

Sumber Daya: `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/vault-name`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/example*`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/*`

Kunci Kondisi Amazon Glacier: `glacier:ResourceTag/`*`TagKey`*

Memulai Kunci Vault (POST lock-policy)

Izin yang Diperlukan (Tindakan API):`glacier:InitiateVaultLock`

Sumber Daya:

Kunci Kondisi Amazon Glacier: `glacier:ResourceTag/`*`TagKey`*

Mencantumkan Tugas (GET jobs)

Izin yang Diperlukan (Tindakan API):`glacier:ListJobs`

Sumber Daya: `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/vault-name`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/example*`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/*`

Kunci Kondisi Amazon Glacier:

Mencantumkan Unggahan Multipart (GET multipart-uploads)

Izin yang Diperlukan (Tindakan API):`glacier:ListMultipartUploads`

Sumber Daya: `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/vault-name`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/example*`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/*`

Kunci Kondisi Amazon Glacier:

Mencantumkan Bagian (GET uploadID)

Izin yang Diperlukan (Tindakan API):`glacier:ListParts`

Sumber Daya: `arn:aws:glacier:region:account-id:vaults/vault-name`, `arn:aws:glacier:region:account-id:vaults/example*`, `arn:aws:glacier:region:account-id:vaults/*`

Kunci Kondisi Amazon Glacier:

Mencantumkan Tanda untuk Vault (GET tags)

Izin yang Diperlukan (Tindakan API):`glacier:ListTagsForVault`

Sumber Daya: `arn:aws:glacier:region:account-id:vaults/vault-name`, `arn:aws:glacier:region:account-id:vaults/example*`, `arn:aws:glacier:region:account-id:vaults/*`

Kunci Kondisi Amazon Glacier:

Mencantumkan Vault (GET vaults)

Izin yang Diperlukan (Tindakan API):`glacier:ListVaults`

Sumber Daya:

Kunci Kondisi Amazon Glacier:

Hapus Tanda Dari Vault (POST tags remove)

Izin yang Diperlukan (Tindakan API):`glacier:RemoveTagsFromVault`

Sumber Daya: `arn:aws:glacier:region:account-id:vaults/vault-name`, `arn:aws:glacier:region:account-id:vaults/example*`, `arn:aws:glacier:region:account-id:vaults/*`

Kunci Kondisi Amazon Glacier: `glacier:ResourceTag/TagKey`

Mengatur Kebijakan Pengambilan Data (PUT policy) *

Izin yang Diperlukan (Tindakan API):`glacier:SetDataRetrievalPolicy`

Sumber Daya:`arn:aws:glacier:region:account-id:policies/retrieval-limit-policy`

Kunci Kondisi Amazon Glacier:

Mendapatkan Kebijakan Akses Vault (PUT access-policy)

Izin yang Diperlukan (Tindakan API):`glacier:SetVaultAccessPolicy`

Sumber Daya: `arn:aws:glacier:region:account-id:vaults/vault-name`, `arn:aws:glacier:region:account-id:vaults/example*`, `arn:aws:glacier:region:account-id:vaults/*`

Kunci Kondisi Amazon Glacier: `glacier:ResourceTag/TagKey`

Menetapkan Konfigurasi Notifikasi Vault (PUT notification-configuration)

Izin yang Diperlukan (Tindakan API):`glacier:SetVaultNotifications`

Sumber Daya: `arn:aws:glacier:region:account-id:vaults/vault-name`, `arn:aws:glacier:region:account-id:vaults/example*`, `arn:aws:glacier:region:account-id:vaults/*`

Kunci Kondisi Amazon Glacier: `glacier:ResourceTag/TagKey`

Mengunggah Arsip (POST archive)

Izin yang Diperlukan (Tindakan API):`glacier:UploadArchive`

Sumber Daya: `arn:aws:glacier:region:account-id:vaults/vault-name`, `arn:aws:glacier:region:account-id:vaults/example*`, `arn:aws:glacier:region:account-id:vaults/*`

Kunci Kondisi Amazon Glacier: `glacier:ResourceTag/TagKey`

Mengunggah Bagian (PUT UploadID)

Izin yang Diperlukan (Tindakan API):`glacier:UploadMultipartPart`

Sumber Daya: `arn:aws:glacier:region:account-id:vaults/vault-name`, `arn:aws:glacier:region:account-id:vaults/example*`, `arn:aws:glacier:region:account-id:vaults/*`

Kunci Kondisi Amazon Glacier: `glacier:ResourceTag/TagKey`

Penebangan dan Pemantauan di Amazon Glacier

Pemantauan adalah bagian penting dalam menjaga keandalan, ketersediaan, dan kinerja Amazon Glacier (Amazon Glacier) dan solusi Anda. AWS Anda harus mengumpulkan data pemantauan dari semua bagian AWS solusi Anda sehingga Anda dapat lebih mudah mengidentifikasi dan men-debug sumber kegagalan jika terjadi. AWS menyediakan alat berikut untuk memantau sumber daya Amazon Glacier Anda dan menanggapi potensi insiden:

CloudWatch Alarm Amazon

Saat menggunakan Amazon Glacier melalui Amazon S3, Anda dapat menggunakan alarm CloudWatch Amazon untuk menonton satu metrik selama periode waktu yang Anda tentukan. Jika metrik melebihi ambang batas tertentu, pemberitahuan dikirim ke topik atau AWS Auto Scaling kebijakan Amazon SNS. CloudWatch alarm tidak memanggil tindakan karena mereka berada dalam keadaan tertentu. Sebaliknya, status harus diubah dan dipelihara selama jangka waktu 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 Glacier. CloudTrail menangkap semua panggilan API untuk Amazon Glacier sebagai acara, termasuk panggilan dari konsol Amazon Glacier dan dari panggilan kode ke Amazon Glacier. APIs Untuk informasi selengkapnya, lihat Mencatat Panggilan API Amazon Glacier dengan AWS CloudTrail.

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.

Untuk informasi lebih lanjut, lihat AWS Trusted Advisor dalam Panduan Pengguna Dukungan .

Validasi Kepatuhan untuk Amazon Glacier

Keamanan dan kepatuhan Amazon Glacier (Amazon Glacier) dinilai oleh auditor pihak ketiga sebagai bagian dari beberapa program kepatuhan, termasuk yang berikut: AWS

- 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 di AWS Artifact Panduan Pengguna.

Untuk informasi selengkapnya tentang program AWS kepatuhan, lihat Program AWS Kepatuhan.

Tanggung jawab kepatuhan Anda saat menggunakan Amazon Glacier ditentukan oleh sensitivitas data Anda, tujuan kepatuhan organisasi Anda, serta undang-undang dan peraturan yang berlaku. Jika penggunaan Amazon Glacier Anda tunduk pada kepatuhan terhadap standar seperti HIPAA, PCI, atau FedRAMP, sediakan sumber daya untuk membantu: AWS

- Kunci Gletser Amazon Glaciermemungkinkan Anda untuk dengan mudah menerapkan dan menerapkan kontrol kepatuhan untuk kubah Amazon Glacier individual dengan kebijakan kunci vault. Anda dapat menentukan kontrol seperti “tulis sekali baca banyak” (WORM) dalam kebijakan vault lock dan mengunci kebijakan dari pengeditan di masa mendatang. Setelah kebijakan dikunci, kebijakan tidak dapat diubah lagi. Kebijakan kunci Vault dapat membantu Anda mematuhi kerangka peraturan seperti SEC17a -4 dan HIPAA.
- Panduan Memulai Cepat Keamanan dan Kepatuhan 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 Well-Architected Tool AWS (WA Tool) adalah layanan di cloud yang menyediakan proses yang konsisten bagi Anda untuk meninjau dan mengukur arsitektur AWS Anda menggunakan

praktik terbaik. Alat AWS WA memberikan rekomendasi untuk membuat beban kerja Anda lebih andal, aman, efisien, dan hemat biaya.

- AWS Sumber Daya Kepatuhan menyediakan beberapa buku kerja dan panduan berbeda yang mungkin berlaku untuk industri dan lokasi Anda.
- AWS Config dapat membantu Anda menilai seberapa baik konfigurasi sumber daya Anda 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.

Ketahanan di Amazon Glacier

Infrastruktur AWS global dibangun di sekitar Wilayah dan Zona Ketersediaan. AWS Wilayah menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi yang terhubung dengan jaringan latensi rendah, throughput tinggi, dan sangat redundan. Availability Zone 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. Amazon Glacier menyimpan data secara berlebihan di beberapa perangkat yang mencakup minimal tiga Availability Zone. Untuk meningkatkan daya tahan, Amazon Glacier secara serempak menyimpan data Anda di AZs beberapa sebelum mengonfirmasi unggahan yang berhasil.

Untuk informasi selengkapnya tentang AWS Wilayah dan Availability Zone, lihat Infrastruktur AWS Global.

Keamanan Infrastruktur di Amazon Glacier

Sebagai layanan terkelola, Amazon Glacier (Amazon Glacier) dilindungi oleh prosedur keamanan jaringan global AWS yang dijelaskan dalam Amazon Web Services: Tinjauan Proses Keamanan.

Akses ke Amazon Glacier melalui jaringan melalui dipublikasikan. AWS APIs Klien harus mendukung Keamanan Lapisan Pengangkutan (TLS) 1.2. Kami merekomendasikan TLS 1.3 atau versi yang lebih baru. Klien juga harus mendukung cipher suites dengan Perfect Forward Secrecy (PFS) seperti Ephemeral Diffie-Hellman (DHE) atau Elliptic Curve Diffie-Hellman Ephemeral (ECDHE). Sebagian besar sistem modern seperti Java 7 dan versi yang lebih baru support mode ini. Selain itu, permintaan harus ditandatangani menggunakan access key ID dan secret access key yang terkait dengan prinsipal IAM, atau Anda dapat menggunakan AWS Security Token Service (AWS STS) untuk membuat kredensial keamanan sementara untuk menandatangani permintaan.

VPC Endpoint

Titik akhir virtual private cloud (VPC) memungkinkan Anda menghubungkan VPC Anda secara pribadi ke layanan yang didukung AWS dan layanan titik akhir VPC yang didukung AWS PrivateLink tanpa memerlukan gateway internet, perangkat NAT, koneksi VPN, atau koneksi. AWS Direct Connect Meskipun Amazon Glacier tidak mendukung titik akhir VPC secara langsung, Anda dapat memanfaatkan titik akhir Amazon S3 VPC jika Anda mengakses Amazon Glacier sebagai tingkat penyimpanan yang terintegrasi dengan Amazon S3.

Untuk informasi selengkapnya tentang konfigurasi siklus hidup Amazon S3 dan objek transisi ke kelas penyimpanan Amazon Glacier, lihat Objek Manajemen dan Transisi Siklus Hidup Objek di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon. Untuk informasi selengkapnya tentang VPC endpoint, lihat VPC Endpoint dalam Panduan Pengguna Amazon VPC.

Kebijakan Pengambilan Data Amazon Glacier

Dengan kebijakan pengambilan data Amazon Glacier, Anda dapat dengan mudah menyetel kuota pengambilan data dan mengelola aktivitas pengambilan data di masing-masing. Akun AWS AWS RegionUntuk informasi selengkapnya tentang biaya pengambilan data Amazon Glacier, lihat harga Amazon Glacier.

> **Important**
>
> Kebijakan pengambilan data hanya berlaku untuk pengambilan Standar dan mengelola permintaan pengambilan yang dibuat langsung ke Amazon Glacier.
> Untuk informasi selengkapnya tentang kelas penyimpanan Amazon Glacier, lihat Kelas penyimpanan untuk mengarsipkan objek dan objek Transisi di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Topik

- Memilih Kebijakan Pengambilan Data Amazon Glacier
- Menggunakan Konsol Amazon Glacier untuk Mengatur Kebijakan Pengambilan Data
- Menggunakan Amazon Glacier API untuk Mengatur Kebijakan Pengambilan Data

Memilih Kebijakan Pengambilan Data Amazon Glacier

Anda dapat memilih dari tiga jenis kebijakan pengambilan data Amazon Glacier: Tanpa Batas Pengambilan, Hanya Tingkat Gratis, dan Tingkat Pengambilan Maks.

No Retrieval Limit adalah kebijakan pengambilan data default yang digunakan untuk pengambilan. Jika Anda menggunakan kebijakan Tanpa Batas Pengambilan, tidak ada kuota pengambilan yang ditetapkan, dan semua permintaan pengambilan data yang valid akan diterima.

Dengan menggunakan kebijakan Tier Only Gratis, Anda dapat menyimpan pengambilan Anda dalam tunjangan Tingkat AWS Gratis harian Anda dan tidak dikenakan biaya pengambilan data apa pun. Jika Anda ingin mengambil lebih banyak data daripada yang ada di tunjangan Tingkat AWS Gratis, Anda dapat menggunakan kebijakan Tingkat Pengambilan Maks untuk menetapkan kuota tarif pengambilan. bytes-per-hour Kebijakan Max Retrieval Rate memastikan bahwa tingkat pengambilan

puncak dari semua pekerjaan pengambilan di seluruh akun Anda AWS Region tidak melebihi bytes-per-hour kuota yang Anda tetapkan.

Dengan kebijakan Tingkat Gratis Saja dan Tingkat Pengambilan Maks, permintaan pengambilan data yang melebihi kuota pengambilan yang Anda tentukan tidak diterima. Jika Anda menggunakan kebijakan Tier Saja Gratis, Amazon Glacier secara serempak menolak permintaan pengambilan yang melebihi tunjangan Tingkat Gratis Anda. AWS Jika Anda menggunakan kebijakan Tingkat Pengambilan Maks, Amazon Glacier menolak permintaan pengambilan yang menyebabkan tingkat pengambilan puncak pekerjaan yang sedang berlangsung melebihi kuota yang ditetapkan oleh kebijakan. bytes-per-hour Kebijakan ini membantu Anda menyederhanakan manajemen biaya pengambilan data.

Berikut adalah beberapa fakta yang berguna tentang kebijakan pengambilan data:

- Pengaturan kebijakan pengambilan data tidak mengubah periode 3 hingga 5 jam yang diperlukan untuk mengambil data dari Amazon Glacier dengan menggunakan pengambilan Standar.
- Menetapkan kebijakan pengambilan data baru tidak memengaruhi tugas pengambilan yang diterima sebelumnya yang sudah berlangsung.
- Jika permintaan pekerjaan pengambilan ditolak karena kebijakan pengambilan data, Anda tidak dikenakan biaya untuk pekerjaan atau permintaan tersebut.
- Anda dapat menetapkan satu kebijakan pengambilan data untuk masing-masing AWS Region, yang akan mengatur semua aktivitas pengambilan data di AWS Region bawah akun Anda. Kebijakan pengambilan data khusus untuk yang tertentu AWS Region karena biaya pengambilan data bervariasi. Wilayah AWS Untuk informasi selengkapnya, lihat harga Amazon Glacier.

Kebijakan Hanya Tingkat Gratis

Anda dapat menetapkan kebijakan pengambilan data ke Tingkat Gratis Hanya untuk memastikan bahwa pengambilan Anda selalu berada dalam tunjangan Tingkat AWS Gratis Anda, sehingga Anda tidak dikenakan biaya pengambilan data. Jika permintaan pengambilan ditolak, Anda menerima pesan galat yang menyatakan bahwa permintaan tersebut telah ditolak oleh kebijakan pengambilan data saat ini.

Anda dapat menyetel kebijakan pengambilan data ke Tier Gratis Hanya berdasarkan per wilayah. Setelah kebijakan ditetapkan, Anda tidak dapat mengambil lebih banyak data dalam sehari daripada tunjangan pengambilan Tingkat AWS Gratis harian prorata untuk itu. AWS Region Anda juga tidak dikenakan biaya pengambilan data.

Anda juga dapat beralih ke kebijakan Tier Only Gratis setelah Anda mengeluarkan biaya pengambilan data dalam waktu satu bulan. Dalam hal ini, kebijakan Tier Only Gratis berlaku untuk permintaan pengambilan baru, tetapi tidak memengaruhi permintaan sebelumnya. Anda akan ditagih untuk biaya yang dikeluarkan sebelumnya.

Kebijakan Tingkat Pengambilan Maksimal

Anda dapat mengatur kebijakan pengambilan data Anda ke Max Retrieval Rate untuk mengontrol tingkat pengambilan puncak dengan menentukan kuota pengambilan data yang memiliki maksimum. bytes-per-hour Saat Anda menyetel kebijakan pengambilan data ke Tingkat Pengambilan Maks, permintaan pengambilan baru akan ditolak jika akan menyebabkan tingkat pengambilan puncak pekerjaan yang sedang berlangsung melebihi bytes-per-hour kuota yang ditentukan oleh kebijakan. Jika permintaan pekerjaan pengambilan ditolak, Anda menerima pesan galat yang menyatakan bahwa permintaan tersebut telah ditolak oleh kebijakan pengambilan data saat ini.

Menyetel kebijakan pengambilan data Anda ke kebijakan Tingkat Pengambilan Maks dapat memengaruhi jumlah tunjangan Tingkat AWS Gratis yang dapat Anda gunakan dalam sehari. Misalnya, Anda menetapkan Tingkat Pengambilan Maksimal ke 1 MB per jam. Ini kurang dari tarif kebijakan Tingkat AWS Gratis. Untuk memastikan bahwa Anda memanfaatkan tunjangan Tingkat AWS Gratis harian dengan baik, pertama-tama Anda dapat menetapkan kebijakan Anda ke Tier Gratis Saja, lalu beralih ke kebijakan Tingkat Pengambilan Maks nanti jika perlu. Untuk informasi lebih lanjut tentang bagaimana tunjangan pengambilan Anda dihitung, kunjungi Amazon Glacier. FAQs

Kebijakan Tidak Ada Batas Pengambilan

Jika kebijakan pengambilan data Anda disetel ke Tanpa Batas Pengambilan, semua permintaan pengambilan data yang valid akan diterima dan biaya pengambilan data Anda akan bervariasi berdasarkan penggunaan Anda.

Menggunakan Konsol Amazon Glacier untuk Mengatur Kebijakan Pengambilan Data

Untuk membuat kebijakan pengambilan data menggunakan konsol Amazon Glacier

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Glacier di rumah. https://console.aws.amazon.com/glacier/

2. Di bawah Pilih Wilayah, pilih AWS Region dari menu tarik-turun. Anda dapat mengonfigurasi kebijakan pengambilan data untuk masing-masing AWS Region.
3. Di panel navigasi kiri, pilih Pengaturan pengambilan data.
4. Pilih Edit. Halaman Edit kebijakan pengambilan data muncul.
5. Di bawah Kebijakan pengambilan data, pilih kebijakan.

   Anda dapat memilih salah satu dari tiga kebijakan pengambilan data: Tidak ada batas pengambilan, Tier Gratis saja, atau Tentukan tingkat pengambilan maksimal.

   - Jika Anda memilih Tidak ada batas pengambilan, semua permintaan pengambilan data yang valid akan diterima.
   - Jika Anda memilih Tingkat Gratis saja, permintaan pengambilan data yang melebihi Tingkat AWS Gratis tidak akan diterima.
   - Jika Anda memilih Tentukan tingkat pengambilan maksimal, permintaan pengambilan data ditolak jika permintaan tersebut akan menyebabkan tingkat pengambilan puncak pekerjaan yang sedang berlangsung melebihi tingkat pengambilan maksimal yang Anda tentukan. Anda harus menentukan nilai gigabyte (GB) per jam di kotak GB/jam di bawah Tingkat pengambilan maksimum. Saat Anda memasukkan nilai GB/jam, konsol menghitung perkiraan biaya untuk Anda.
6. Pilih Simpan perubahan.

Menggunakan Amazon Glacier API untuk Mengatur Kebijakan Pengambilan Data

Anda dapat melihat dan menyetel kebijakan pengambilan data menggunakan Amazon Glacier REST API atau dengan menggunakan. AWS SDKs

Menggunakan Amazon Glacier REST API untuk Mengatur Kebijakan Pengambilan Data

Anda dapat melihat dan menetapkan kebijakan pengambilan data menggunakan Amazon Glacier REST API. Anda dapat melihat kebijakan pengambilan data yang ada menggunakan operasi Mendapatkan Kebijakan Pengambilan Data (GET policy). Anda menetapkan kebijakan pengambilan data dengan menggunakan Mengatur Kebijakan Pengambilan Data (PUT policy) operasi.

Saat menggunakan operasi PUT kebijakan, Anda memilih jenis kebijakan pengambilan data dengan menyetel nilai `Strategy` bidang JSON ke`BytesPerHour`,`FreeTier`, atau. `None` `BytesPerHour`setara dengan memilih Tentukan tingkat pengambilan maksimal di konsol, `FreeTier` untuk memilih Tingkat Gratis saja, dan `None` memilih Tidak ada batas pengambilan.

Bila Anda menggunakan Memulai Tugas (POST jobs) operasi untuk memulai pekerjaan pengambilan data yang akan melebihi tingkat pengambilan maksimum yang ditetapkan dalam kebijakan pengambilan data Anda, `Initiate Job` operasi akan berhenti dan melempar pengecualian.

Menggunakan AWS SDKs untuk Mengatur Kebijakan Pengambilan Data

AWS menyediakan SDKs bagi Anda untuk mengembangkan aplikasi untuk Amazon Glacier. Ini SDKs menyediakan pustaka yang memetakan ke REST API yang mendasarinya dan menyediakan objek yang memungkinkan Anda membuat permintaan dan respons proses dengan mudah. Lihat informasi yang lebih lengkap di Menggunakan AWS SDKs dengan Amazon Glacier.

Menandai Sumber Daya Amazon Glacier

Tag adalah label yang Anda tetapkan ke AWS sumber daya. Setiap tag terdiri atas sebuah kunci dan sebuah nilai, yang keduanya Anda tentukan. Anda dapat menetapkan tag yang Anda tentukan ke sumber daya vault Amazon Glacier (Amazon Glacier). Menggunakan tag adalah cara sederhana namun ampuh untuk mengelola AWS sumber daya dan mengatur data, termasuk data penagihan.

Topik

- Dasar Penandaan
- Pembatasan Tag
- Melacak Biaya Menggunakan Penandaan
- Mengelola Kontrol Akses dengan Penandaan
- Bagian Terkait

Dasar Penandaan

Anda menggunakan konsol Amazon Glacier AWS Command Line Interface ,AWS CLI(), atau Amazon Glacier API untuk menyelesaikan tugas-tugas berikut:

- Menambahkan tanda ke vault
- Mencantumkan tanda untuk vault
- Menghapus tanda dari vault

Untuk informasi tentang cara menambahkan, mencantumkan, dan menghapus tanda, lihat Menandai Brankas Amazon Glacier Anda.

Anda dapat menggunakan tanda untuk mengategorikan vault Anda. Misalnya, Anda dapat mengategorikan vault berdasarkan tujuan, pemilik, atau lingkungan. Karena Anda menentukan kunci dan nilai untuk setiap tanda, Anda dapat membuat serangkaian kategori khusus untuk memenuhi kebutuhan spesifik Anda. Misalnya, Anda dapat menentukan sekumpulan tanda yang membantu Anda melacak vault berdasarkan pemilik dan tujuan vault. Berikut adalah beberapa contoh tanda:

- Pemilik: Nama
- Tujuan: Arsip video

- Lingkungan: Produksi

Pembatasan Tag

Pembatasan tanda dasar adalah sebagai berikut:

- Jumlah maksimum tanda untuk sumber daya (vault) adalah 50.
- Kunci dan nilai tag peka huruf besar dan kecil.

Pembatasan kunci tanda adalah sebagai berikut:

- Dalam sekumpulan tanda untuk vault, setiap kunci tanda harus unik. Jika Anda menambahkan tanda dengan kunci yang sudah digunakan, tanda baru akan menimpa pasangan nilai-kunci yang sudah ada.
- Kunci tag tidak dapat dimulai `aws:` karena awalan ini dicadangkan untuk digunakan oleh AWS. AWS dapat membuat tag yang dimulai dengan awalan ini atas nama Anda, tetapi Anda tidak dapat mengedit atau menghapusnya.
- Kunci tanda harus memiliki panjang dari 1 hingga 128 karakter Unicode.
- Kunci tanda harus terdiri dari karakter berikut: huruf Unicode, digit, spasi, dan karakter khusus berikut: `_ . / = + - @`.

Pembatasan nilai tanda adalah sebagai berikut:

- Kunci tanda harus memiliki panjang dari 0 hingga 255 karakter Unicode.
- Nilai tanda dapat kosong. Jika tidak, nilai tanda harus terdiri dari karakter berikut: huruf Unicode, digit, spasi, dan salah satu karakter khusus berikut: `_ . / = + - @`.

Melacak Biaya Menggunakan Penandaan

Anda dapat menggunakan tag untuk mengkategorikan dan melacak biaya Anda AWS . Saat Anda menerapkan tag ke AWS sumber daya apa pun, termasuk brankas, laporan alokasi AWS biaya Anda mencakup penggunaan dan biaya yang dikumpulkan berdasarkan tag. Anda dapat menerapkan tanda yang mewakili kategori bisnis (seperti pusat biaya, nama aplikasi, dan pemilik) untuk mengatur biaya Anda di berbagai layanan. Untuk informasi selengkapnya, lihat Menggunakan Tanda Alokasi Biaya untuk Laporan Penagihan Khusus dalam Panduan Pengguna AWS Billing .

Mengelola Kontrol Akses dengan Penandaan

Anda dapat menggunakan tanda sebagai kondisi dalam pernyataan kebijakan akses. Misalnya, Anda dapat menyiapkan tanda pengamanan legal dan menyertakannya sebagai kondisi dalam kebijakan retensi data yang menyatakan bahwa “penghapusan arsip dari semua orang akan ditolak jika nilai tanda pengamanan legal diatur ke `True`.” Anda dapat men-deploy kebijakan retensi data dan menetapkan tanda pengamanan legal ke `False` dalam kondisi normal. Jika data Anda harus diamankan untuk membantu penyelidikan, Anda dapat dengan mudah mengaktifkan pengamanan legal dengan menetapkan nilai tanda ke `True` dan menghapus pengamanan dengan cara yang sama di kemudian hari. Untuk informasi selengkapnya, lihat Mengontrol Akses Menggunakan Tanda di Panduan Pengguna IAM.

Bagian Terkait

- Menandai Brankas Amazon Glacier Anda

Mencatat Panggilan API Amazon Glacier dengan AWS CloudTrail

Amazon Glacier (Amazon Glacier) terintegrasi AWS CloudTrail dengan, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau layanan di Amazon Glacier. AWS CloudTrail menangkap semua panggilan API untuk Amazon Glacier sebagai acara, termasuk panggilan dari konsol Amazon Glacier dan dari panggilan kode ke Amazon Glacier. APIs Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara terus menerus ke bucket Amazon S3, termasuk acara untuk Amazon Glacier. Jika Anda tidak membuat konfigurasi jejak, Anda masih dapat melihat kejadian terbaru dalam konsol CloudTrail di Riwayat peristiwa. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat ke Amazon Glacier, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna.

Informasi Amazon Glacier di CloudTrail

CloudTrail diaktifkan pada Akun AWS saat Anda membuat akun. Ketika aktivitas terjadi di Amazon Glacier, aktivitas tersebut direkam dalam CloudTrail suatu peristiwa bersama dengan peristiwa layanan AWS lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh acara terbaru di situs Anda Akun AWS. Untuk informasi selengkapnya, lihat Melihat Acara dengan Riwayat CloudTrail Acara.

Untuk catatan acara yang sedang berlangsung di Anda Akun AWS, termasuk acara untuk Amazon Glacier, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua AWS Wilayah. Jejak mencatat peristiwa dari semua AWS Wilayah di AWS partisi dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi lebih lanjut, lihat:

- Gambaran Umum untuk Membuat Jejak
- CloudTrail Layanan dan Integrasi yang Didukung
- Mengkonfigurasi Notifikasi Amazon SNS untuk CloudTrail

- Menerima File CloudTrail Log dari Beberapa Wilayah dan Menerima File CloudTrail Log dari Beberapa Akun

Semua tindakan Amazon Glacier dicatat CloudTrail oleh dan didokumentasikan di. Referensi API untuk Amazon Glacier Misalnya, panggilan keMembuat Vault (PUT vault),Menghapus Vault (DELETE vault), dan Mencantumkan Vault (GET vaults) tindakan menghasilkan entri dalam file CloudTrail log.

Setiap entri peristiwa atau log berisi informasi tentang entitas yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan hal berikut ini:

- Apakah permintaan dibuat dengan pengguna root atau kredensional lainnya.
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna gabungan.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi lain, lihat Elemen userIdentity CloudTrail .

Memahami Entri File Log Amazon Glacier

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa mewakili permintaan tunggal dari sumber manapun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, sehingga file tersebut tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkanMembuat Vault (PUT vault),, Menghapus Vault (DELETE vault)Mencantumkan Vault (GET vaults), dan Menjelaskan Vault (GET vault) tindakan.

```
{
    "Records": [
        {
            "awsRegion": "us-east-1",
            "eventID": "52f8c821-002e-4549-857f-8193a15246fa",
            "eventName": "CreateVault",
            "eventSource": "glacier.amazonaws.com",
            "eventTime": "2014-12-10T19:05:15Z",
```

```
            "eventType": "AwsApiCall",
            "eventVersion": "1.02",
            "recipientAccountId": "999999999999",
            "requestID": "HJiLgvfXCY88QJAC6rRoexS9ThvI21Q1Nqukfly02hcUPPo",
            "requestParameters": {
                "accountId": "-",
                "vaultName": "myVaultName"
            },
            "responseElements": {
                "location": "/999999999999/vaults/myVaultName"
            },
            "sourceIPAddress": "127.0.0.1",
            "userAgent": "aws-sdk-java/1.9.6 Mac_OS_X/10.9.5 Java_HotSpot(TM)_64-
Bit_Server_VM/25.25-b02/1.8.0_25",
            "userIdentity": {
                "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
                "accountId": "999999999999",
                "arn": "arn:aws:iam::999999999999:user/myUserName",
                "principalId": "A1B2C3D4E5F6G7EXAMPLE",
                "type": "IAMUser",
                "userName": "myUserName"
            }
        },
        {
            "awsRegion": "us-east-1",
            "eventID": "cdd33060-4758-416a-b7b9-dafd3afcec90",
            "eventName": "DeleteVault",
            "eventSource": "glacier.amazonaws.com",
            "eventTime": "2014-12-10T19:05:15Z",
            "eventType": "AwsApiCall",
            "eventVersion": "1.02",
            "recipientAccountId": "999999999999",
            "requestID": "GGdw-VfhVfLCFwAM6iVUvMQ6-fMwSqSO9FmRd0eRSa_Fc7c",
            "requestParameters": {
                "accountId": "-",
                "vaultName": "myVaultName"
            },
            "responseElements": null,
            "sourceIPAddress": "127.0.0.1",
            "userAgent": "aws-sdk-java/1.9.6 Mac_OS_X/10.9.5 Java_HotSpot(TM)_64-
Bit_Server_VM/25.25-b02/1.8.0_25",
            "userIdentity": {
                "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
                "accountId": "999999999999",
```

```
                "arn": "arn:aws:iam::999999999999:user/myUserName",
                "principalId": "A1B2C3D4E5F6G7EXAMPLE",
                "type": "IAMUser",
                "userName": "myUserName"
            }
        },
        {
            "awsRegion": "us-east-1",
            "eventID": "355750b4-e8b0-46be-9676-e786b1442470",
            "eventName": "ListVaults",
            "eventSource": "glacier.amazonaws.com",
            "eventTime": "2014-12-10T19:05:15Z",
            "eventType": "AwsApiCall",
            "eventVersion": "1.02",
            "recipientAccountId": "999999999999",
            "requestID": "yPTs22ghTsWprFivb-2u30FAaDALIZP17t4jM_xL9QJQyVA",
            "requestParameters": {
                "accountId": "-"
            },
            "responseElements": null,
            "sourceIPAddress": "127.0.0.1",
            "userAgent": "aws-sdk-java/1.9.6 Mac_OS_X/10.9.5 Java_HotSpot(TM)_64-
Bit_Server_VM/25.25-b02/1.8.0_25",
            "userIdentity": {
                "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
                "accountId": "999999999999",
                "arn": "arn:aws:iam::999999999999:user/myUserName",
                "principalId": "A1B2C3D4E5F6G7EXAMPLE",
                "type": "IAMUser",
                "userName": "myUserName"
            }
        },
        {
            "awsRegion": "us-east-1",
            "eventID": "569e830e-b075-4444-a826-aa8b0acad6c7",
            "eventName": "DescribeVault",
            "eventSource": "glacier.amazonaws.com",
            "eventTime": "2014-12-10T19:05:15Z",
            "eventType": "AwsApiCall",
            "eventVersion": "1.02",
            "recipientAccountId": "999999999999",
            "requestID": "QRt1ZdFLGn0TCm784HmKafBmcB2lVaV81UU3fsOR3PtoIiM",
            "requestParameters": {
                "accountId": "-",
```

```
                "vaultName": "myVaultName"
            },
            "responseElements": null,
            "sourceIPAddress": "127.0.0.1",
            "userAgent": "aws-sdk-java/1.9.6 Mac_OS_X/10.9.5 Java_HotSpot(TM)_64-
Bit_Server_VM/25.25-b02/1.8.0_25",
            "userIdentity": {
                "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
                "accountId": "999999999999",
                "arn": "arn:aws:iam::999999999999:user/myUserName",
                "principalId": "A1B2C3D4E5F6G7EXAMPLE",
                "type": "IAMUser",
                "userName": "myUserName"
            }
        }
    ]
}
```

Referensi API untuk Amazon Glacier

Amazon Glacier mendukung serangkaian operasi—khususnya, serangkaian panggilan API—yang memungkinkan Anda RESTful berinteraksi dengan layanan.

Anda dapat menggunakan pustaka pemrograman apa pun yang dapat mengirim permintaan HTTP untuk mengirim permintaan REST Anda ke Amazon Glacier. Saat mengirim permintaan REST, Amazon Glacier mengharuskan Anda mengautentikasi setiap permintaan dengan menandatangani permintaan. Selain itu, ketika mengunggah arsip, Anda juga harus menghitung checksum muatan dan memasukkannya ke dalam permintaan Anda. Untuk informasi selengkapnya, lihat Menandatangani Permintaan.

Jika terjadi kesalahan, Anda perlu tahu apa yang dikirimkan Amazon Glacier dalam respons kesalahan sehingga Anda dapat memprosesnya. Bagian ini menyediakan semua informasi ini, selain untuk mendokumentasikan operasi REST, agar Anda dapat membuat panggilan REST API langsung.

Anda dapat menggunakan panggilan REST API secara langsung atau menggunakan Amazon SDKs yang menyediakan pustaka pembungkus. Pustaka ini menandatangani setiap permintaan yang Anda kirim dan menghitung checksum muatan dalam permintaan Anda. Oleh karena itu, menggunakan Amazon SDKs menyederhanakan tugas pengkodean Anda. Panduan pengembang ini memberikan contoh kerja operasi Amazon Glacier dasar menggunakan AWS SDK for Java dan .NET. Untuk informasi selengkapnya, lihat Menggunakan AWS SDKs dengan Amazon Glacier.

Topik

- Header Permintaan Umum
- Header Respons Umum
- Menandatangani Permintaan
- Checksum Komputasi
- Respons Kesalahan
- Operasi Vault
- Operasi Arsip
- Operasi Unggahan Multipart
- Operasi Tugas
- Tipe Data yang Digunakan dalam Operasi Tugas

- Operasi Pengambilan Data

Header Permintaan Umum

Permintaan REST Amazon Glacier (Amazon Glacier) menyertakan header yang berisi informasi dasar tentang permintaan tersebut. Tabel berikut menjelaskan header yang dapat digunakan oleh semua permintaan Amazon Glacier REST.

Nama Header	Deskripsi	Wajib
`Authorization`	Header yang diperlukan untuk menandatangani permintaan. Amazon Glacier membutuhkan Signature Versi 4. Untuk informasi selengkapnya, lihat Menandatangani Permintaan. Tipe: String	Ya
`Content-Length`	Panjang isi permintaan (tanpa header). Tipe: String Syarat: Diperlukan hanya untuk API Mengunggah Arsip (POST archive).	Bersyarat
`Date`	Tanggal yang dapat digunakan untuk membuat tanda tangan yang terdapat dalam header `Authorization` . Jika header `Date` digunakan untuk penandatanganan, header harus ditentuka n dalam format dasar ISO 8601. Dalam hal ini, header `x-amz-date` tidak diperlukan. Perhatikan bahwa ketika `x-amz-date` tersedia, nilai header `Date` akan selalu ditimpa. Jika header Tanggal tidak digunakan untuk penandatanganan, header tersebut bisa menjadi salah satu dari format tanggal penuh yang ditentukan oleh RFC 2616, bagian 3.3. Misalnya, tanggal/waktu berikut `Wed, 10 Feb 2017`	Bersyarat

Nama Header	Deskripsi	Wajib
	`12:00:00 GMT` adalah date/time header yang valid untuk digunakan dengan Amazon Glacier. Jika Anda menggunakan header `Date` untuk penandatanganan, header tersebut harus dalam format `YYYYMMDD'T'HHMMSS'Z'` dasar ISO 8601. Tipe: String Syarat: Jika `Date` ditentukan, tetapi tidak dalam format dasar ISO 8601, Anda juga harus menyertakan header `x-amz-date` . Jika `Date` ditentukan dalam format dasar ISO 8601, ini cukup untuk menandatangani permintaan dan Anda tidak memerlukan header `x-amz-date` . Untuk informasi selengkapnya, lihat Menangani Tanggal dalam Signature Versi 4 di Glosarium Amazon Web Services.	
`Host`	Header ini menentukan titik akhir layanan tempat Anda mengirim permintaan Anda. Nilai harus dari bentuk "`glacier.`*`region`*`.amazonaws.com` “, di *`region`* mana diganti dengan sebutan AWS Region seperti`us-west-2` . Tipe: String	Ya

Nama Header	Deskripsi	Wajib
`x-amz-content-sha256`	SHA256 Checksum yang dihitung dari seluruh muatan yang diunggah dengan salah satu atau. Mengunggah Arsip (POST archive) Mengunggah Bagian (PUT UploadID) Header ini tidak sama dengan header `x-amz-sha256-tree-hash` , meskipun, untuk beberapa muatan kecil, nilai-nil ainya sama. Ketika `x-amz-content-sha256` wajib diisi, `x-amz-content-sha256` dan `x-amz-sha256-tree-hash` harus ditentukan. Tipe: String Syarat: Diperlukan untuk streaming API,Mengunggah Arsip (POST archive) dan Mengunggah Bagian (PUT UploadID).	Bersyarat
`x-amz-date`	Tanggal yang digunakan untuk membuat tanda tangan di header Otorisasi. Format harus berupa ISO 8601 dasar dalam format `YYYYMMDD'T'HHMMSS'Z'` . Misalnya, berikut ini date/time `20170210T120000Z` valid `x-amz-date` untuk digunakan dengan Amazon Glacier. Tipe: String Syarat: `x-amz-date` bersifat opsional untuk semua permintaan; ini dapat digunakan untuk menimpa tanggal yang digunakan untuk menandatangani permintaan. Jika `Date` ditentukan dalam format dasar ISO 8601, `x-amz-date` tidak diperlukan. Ketika `x-amz-date` tersedia, nilai header `Date` akan selalu ditimpa. Untuk informasi selengkapnya, lihat Menangani Tanggal dalam Signature Versi 4 di Glosarium Amazon Web Services.	Bersyarat

Nama Header	Deskripsi	Wajib
`x-amz-glacier-version`	Versi Amazon Glacier API untuk digunakan. Versi saat ini adalah `2012-06-01` . Tipe: String	Ya
`x-amz-sha256-tree-hash`	Checksum SHA256 hash pohon yang dihitung untuk arsip () atau bagian arsip (Mengunggah Arsip (POST archive)) yang diunggah. Mengunggah Bagian (PUT UploadID) Untuk informasi selengkapnya tentang menghitung checksum ini, lihat Checksum Komputasi. Tipe: String Default: Tidak ada Syarat: Diperlukan untuk Mengunggah Arsip (POST archive) dan Mengunggah Bagian (PUT UploadID).	Bersyarat

Header Respons Umum

Tabel berikut menjelaskan header respons yang umum untuk sebagian besar respons API.

Nama	Deskripsi
`Content-Length`	Panjang dalam byte isi respons. Tipe: String
`Date`	Tanggal dan waktu Amazon Glacier (Amazon Glacier) merespons, misalnya,. `Wed, 10 Feb 2017 12:00:00 GMT` Format tanggal harus salah satu dari format tanggal penuh yang ditentukan oleh RFC 2616, bagian 3.3. Perhatikan bahwa `Date` yang dikembalikan mungkin sedikit menyimpang dari tanggal lain, misalnya, tanggal yang dikembalikan dari permintaan Mengunggah

Nama	Deskripsi
	Arsip (POST archive) mungkin tidak cocok dengan tanggal yang ditampilkan untuk arsip dalam daftar inventaris untuk vault. Tipe: String
`x-amzn-RequestId`	Nilai yang dibuat oleh Amazon Glacier yang secara unik mengidentifikasi permintaan Anda. Jika Anda memiliki masalah dengan Amazon Glacier AWS , dapat menggunakan nilai ini untuk memecahkan masalah. Sebaiknya catat nilai-nilai ini. Tipe: String
`x-amz-sha256-tree-hash`	Checksum SHA256 hash pohon dari arsip atau badan inventaris. Untuk informasi selengkapnya tentang menghitung checksum ini, lihat Checksum Komputasi. Tipe: String

Menandatangani Permintaan

Amazon Glacier mengharuskan Anda mengautentikasi setiap permintaan yang Anda kirim dengan menandatangani permintaan. Untuk menandatangani permintaan, Anda menghitung tanda tangan digital menggunakan fungsi hash kriptografi. Hash kriptografi adalah fungsi yang mengembalikan nilai hash unik berdasarkan input. Input ke fungsi hash termasuk teks permintaan Anda dan secret access key Anda. Fungsi hash mengembalikan nilai hash yang Anda sertakan dalam permintaan sebagai tanda tangan Anda. Tanda tangan adalah bagian header `Authorization` dari permintaan Anda.

Setelah menerima permintaan Anda, Amazon Glacier menghitung ulang tanda tangan menggunakan fungsi hash yang sama dan masukan yang Anda gunakan untuk menandatangani permintaan. Jika tanda tangan yang dihasilkan cocok dengan tanda tangan dalam permintaan, Amazon Glacier memproses permintaan tersebut. Jika tidak, permintaan ditolak.

Amazon Glacier mendukung otentikasi AWS menggunakan Signature Version 4. Proses untuk menghitung tanda tangan dapat dibagi menjadi tiga tugas:

- Tugas 1: Membuat Permintaan Kanonik

Atur ulang permintaan HTTP Anda ke dalam format kanonik. Menggunakan bentuk kanonik diperlukan karena Amazon Glacier menggunakan bentuk kanonik yang sama ketika menghitung ulang tanda tangan untuk dibandingkan dengan yang Anda kirim.

- Tugas 2: Membuat String to Sign

  Buat string yang akan Anda gunakan sebagai salah satu nilai input untuk fungsi hash kriptografi Anda. String, yang disebut string to sign, adalah rangkaian dari nama algoritme hash, tanggal permintaan, string cakupan kredensial, dan permintaan kanonikalisasi dari tugas sebelumnya. String lingkup kredensi itu sendiri adalah rangkaian tanggal, AWS Wilayah, dan informasi layanan.

- Tugas 3: Membuat tanda tangan

  Buat tanda tangan untuk permintaan Anda menggunakan fungsi hash kriptografi yang menerima dua string input: string to sign dan kunci turunan. Kunci turunan dihitung dengan memulai dengan kunci akses rahasia Anda dan menggunakan string cakupan kredensi untuk membuat serangkaian kode otentikasi pesan berbasis hash (). HMACs Perhatikan bahwa fungsi hash yang digunakan dalam langkah penandatanganan ini bukanlah algoritma hash pohon yang digunakan di Amazon Glacier yang mengunggah data. APIs

Topik

- Contoh Perhitungan Tanda Tangan
- Menghitung Tanda Tangan untuk Operasi Streaming

Contoh Perhitungan Tanda Tangan

Contoh berikut memandu Anda melalui detail pembuatan tanda tangan untuk Membuat Vault (PUT vault). Contoh dapat digunakan sebagai referensi untuk memeriksa metode perhitungan tanda tangan Anda. Untuk informasi selengkapnya, lihat Menandatangani permintaan AWS API di Panduan Pengguna IAM.

Contoh tersebut mengasumsikan sebagai berikut:

- Stempel waktu permintaan adalah `Fri, 25 May 2012 00:24:53 GMT`.
- Titik akhir adalah Wilayah US East (N. Virginia) `us-east-1`.

Sintaks permintaan umum (termasuk isi JSON) adalah:

```
PUT /-/vaults/examplevault HTTP/1.1
Host: glacier.us-east-1.amazonaws.com
Date: Fri, 25 May 2012 00:24:53 GMT
Authorization: SignatureToBeCalculated
x-amz-glacier-version: 2012-06-01
```

Bentuk kanonik permintaan yang dihitung untuk Tugas 1: Membuat Permintaan Kanonik adalah:

```
PUT
/-/vaults/examplevault

host:glacier.us-east-1.amazonaws.com
x-amz-date:20120525T002453Z
x-amz-glacier-version:2012-06-01

host;x-amz-date;x-amz-glacier-version
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

Baris terakhir dari permintaan kanonik adalah hash dari isi permintaan. Selain itu, perhatikan baris ketiga kosong dalam permintaan kanonik. Hal ini karena tidak ada parameter query untuk API ini.

String to sign untuk Tugas 2: Membuat String to Sign adalah:

```
AWS4-HMAC-SHA256
20120525T002453Z
20120525/us-east-1/glacier/aws4_request
5f1da1a2d0feb614dd03d71e87928b8e449ac87614479332aced3a701f916743
```

Baris pertama string to sign adalah algoritme, baris kedua adalah stempel waktu, baris ketiga adalah cakupan kredensial, dan baris terakhir adalah hash permintaan kanonik dari Tugas 1: Membuat Permintaan Kanonik. Nama layanan untuk digunakan dalam cakupan kredensial adalah `glacier`.

Untuk Tugas 3: Membuat tanda tangan, kunci turunan dapat digambarkan sebagai:

```
derived key = HMAC(HMAC(HMAC(HMAC("AWS4" + YourSecretAccessKey,"20120525"),"us-
east-1"),"glacier"),"aws4_request")
```

Jika secret access key, `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY` , digunakan, tanda tangan yang dihitung adalah:

```
3ce5b2f2fffac9262b4da9256f8d086b4aaf42eba5f111c21681a65a127b7c2a
```

Langkah terakhir adalah membangun header `Authorization`. Untuk access key demonstrasi `AKIAIOSFODNN7EXAMPLE`, header (dengan jeda baris yang ditambahkan untuk keterbacaan) adalah:

```
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20120525/us-east-1/
glacier/aws4_request,
SignedHeaders=host;x-amz-date;x-amz-glacier-version,
Signature=3ce5b2f2fffac9262b4da9256f8d086b4aaf42eba5f111c21681a65a127b7c2a
```

Menghitung Tanda Tangan untuk Operasi Streaming

Mengunggah Arsip (POST archive) dan Mengunggah Bagian (PUT UploadID) adalah operasi streaming yang mengharuskan Anda menyertakan header tambahan `x-amz-content-sha256` saat menandatangani dan mengirim permintaan Anda. Langkah-langkah penandatanganan untuk operasi streaming sama persis dengan operasi lain, dengan penambahan header streaming.

Perhitungan header streaming `x-amz-content-sha256` didasarkan pada SHA256 hash dari seluruh konten (payload) yang akan diunggah. Perhatikan bahwa perhitungan ini berbeda dari hash SHA256 pohon (Checksum Komputasi). Selain kasus sepele, nilai hash SHA 256 dari data payload akan berbeda dari hash SHA256 pohon dari data payload.

Jika data payload ditentukan sebagai array byte, Anda dapat menggunakan cuplikan kode Java berikut untuk menghitung hash. SHA256

```
public static byte[] computePayloadSHA256Hash2(byte[] payload) throws
 NoSuchAlgorithmException, IOException {
    BufferedInputStream bis =
       new BufferedInputStream(new ByteArrayInputStream(payload));
    MessageDigest messageDigest = MessageDigest.getInstance("SHA-256");
    byte[] buffer = new byte[4096];
```

```
    int bytesRead = -1;
    while ( (bytesRead = bis.read(buffer, 0, buffer.length)) != -1 ) {
        messageDigest.update(buffer, 0, bytesRead);
    }
    return messageDigest.digest();
}
```

Demikian pula, dalam C # Anda dapat menghitung SHA256 hash dari data payload seperti yang ditunjukkan pada cuplikan kode berikut.

```
public static byte[] CalculateSHA256Hash(byte[] payload)
{
    SHA256 sha256 = System.Security.Cryptography.SHA256.Create();
    byte[] hash = sha256.ComputeHash(payload);

    return hash;
}
```

Contoh Perhitungan Tanda Tangan untuk API Streaming

Contoh berikut memandu Anda melalui detail pembuatan tanda tangan untukMengunggah Arsip (POST archive), salah satu dari dua streaming APIs di Amazon Glacier. Contoh tersebut mengasumsikan sebagai berikut:

- Stempel waktu permintaan adalah `Mon, 07 May 2012 00:00:00 GMT`.
- Titik akhir adalah Wilayah US East (N. Virginia), us-east-1.
- Muatan konten adalah string “Selamat datang di Amazon Glacier.”

Sintaks permintaan umum (termasuk isi JSON) ditampilkan dalam contoh di bawah ini. Perhatikan bahwa header `x-amz-content-sha256` disertakan. Dalam contoh sederhana ini, `x-amz-sha256-tree-hash` dan `x-amz-content-sha256` adalah nilai yang sama. Namun, tidak demikian untuk unggahan arsip yang lebih besar dari 1 MB.

```
POST /-/vaults/examplevault HTTP/1.1
Host: glacier.us-east-1.amazonaws.com
Date: Mon, 07 May 2012 00:00:00 GMT
x-amz-archive-description: my archive
x-amz-sha256-tree-hash: SHA256 tree hash
x-amz-content-sha256: SHA256 payload hash
```

```
Authorization: SignatureToBeCalculated
x-amz-glacier-version: 2012-06-01
```

Bentuk kanonik permintaan yang dihitung untuk Tugas 1: Membuat Permintaan Kanonik ditampilkan di bawah ini. Perhatikan bahwa header streaming `x-amz-content-sha256` disertakan dengan nilainya. Ini berarti Anda harus membaca payload dan menghitung SHA256 hash terlebih dahulu dan kemudian menghitung tanda tangan.

```
POST
/-/vaults/examplevault

host:glacier.us-east-1.amazonaws.com
x-amz-content-sha256:726e392cb4d09924dbad1cc0ba3b00c3643d03d14cb4b823e2f041cff612a628
x-amz-date:20120507T000000Z
x-amz-glacier-version:2012-06-01

host;x-amz-content-sha256;x-amz-date;x-amz-glacier-version
726e392cb4d09924dbad1cc0ba3b00c3643d03d14cb4b823e2f041cff612a628
```

Sisa perhitungan tanda tangan mengikuti langkah-langkah yang diuraikan dalam Contoh Perhitungan Tanda Tangan. Header `Authorization` yang menggunakan secret access key `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY` dan access key `AKIAIOSFODNN7EXAMPLE` ditunjukkan di bawah ini (dengan jeda baris yang ditambahkan untuk keterbacaan):

```
Authorization=AWS4-HMAC-SHA256
Credential=AKIAIOSFODNN7EXAMPLE/20120507/us-east-1/glacier/aws4_request,
SignedHeaders=host;x-amz-content-sha256;x-amz-date;x-amz-glacier-version,
Signature=b092397439375d59119072764a1e9a144677c43d9906fd98a5742c57a2855de6
```

Checksum Komputasi

Saat mengunggah arsip, Anda harus menyertakan header `x-amz-sha256-tree-hash` dan `x-amz-content-sha256`. Header `x-amz-sha256-tree-hash` adalah checksum muatan di isi permintaan Anda. Topik ini menjelaskan cara menghitung header `x-amz-sha256-tree-hash`. Header `x-amz-content-sha256` adalah hash seluruh muatan dan diperlukan untuk otorisasi. Untuk informasi selengkapnya, lihat Contoh Perhitungan Tanda Tangan untuk API Streaming.

Muatan permintaan Anda dapat berupa:

- Seluruh arsip— Saat mengunggah arsip dalam satu permintaan menggunakan API Unggap Arsip, Anda mengirim seluruh arsip di isi permintaan. Dalam hal ini, Anda harus menyertakan checksum seluruh arsip.
- Bagian arsip— Saat mengunggah arsip di bagian menggunakan API unggahan multipart, Anda hanya mengirim sebagian arsip di isi permintaan. Dalam hal ini, Anda menyertakan checksum bagian arsip. Dan setelah Anda mengunggah semua bagian, Anda mengirim permintaan Complete Multipart Upload (Selesaikan Unggahan Multipart), yang harus mencakup checksum seluruh arsip.

Checksum muatan adalah hash tree SHA-256. Ini disebut hash tree karena dalam proses komputasi checksum, Anda menghitung pohon nilai hash SHA-256. Nilai hash pada akar adalah checksum untuk seluruh arsip.

> **Note**
>
> Bagian ini menjelaskan cara menghitung hash pohon SHA-256. Namun, Anda dapat menggunakan prosedur apa pun selama hasilnya yang sama.

Anda menghitung hash tree SHA-256 sebagai berikut:

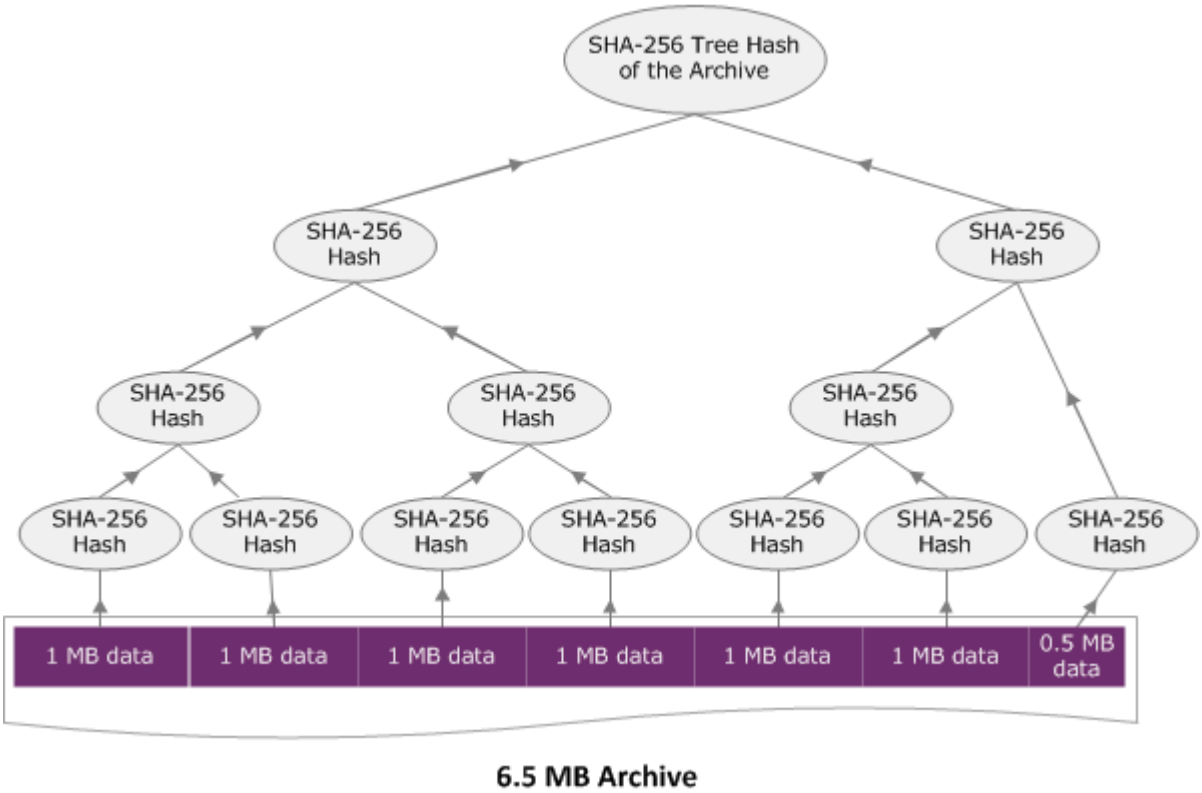
1. Untuk setiap potongan data muatan 1 MB, hitung hash SHA-256. Potongan data terakhir dapat kurang dari 1 MB. Misalnya, jika Anda mengunggah arsip berukuran 3,2 MB, Anda menghitung nilai hash SHA-256 untuk masing-masing tiga potongan data 1 MB pertama, lalu menghitung hash SHA-256 dari 0,2 MB data yang tersisa. Nilai hash ini membentuk node daun pokok.
2. Bangun tingkat berikutnya dari pohon.
   a. Gabungkan dua nilai hash node turunan berturut-turut dan hitung hash SHA-256 dari gabungan nilai hash. Rangkaian dan pembuatan hash SHA-256 ini menghasilkan node induk untuk dua node turunan.
   b. Jika hanya satu node anak yang tersisa, promosikan nilai hash itu ke tingkat berikutnya di pohon.
3. Ulangi langkah 2 sampai pohon yang dihasilkan memiliki akar. Akar pohon menyediakan hash seluruh arsip dan akar subpohom yang sesuai menyediakan hash untuk bagian dalam upload multipart.

Topik

- Contoh Hash Pohon 1: Mengunggah arsip dalam satu permintaan
- Contih Hash Pohon 2: Mengunggah arsip menggunakan unggahan multipart
- Menghitung Hash Pohon File
- Menerima Checksum Saat Mengunduh Data

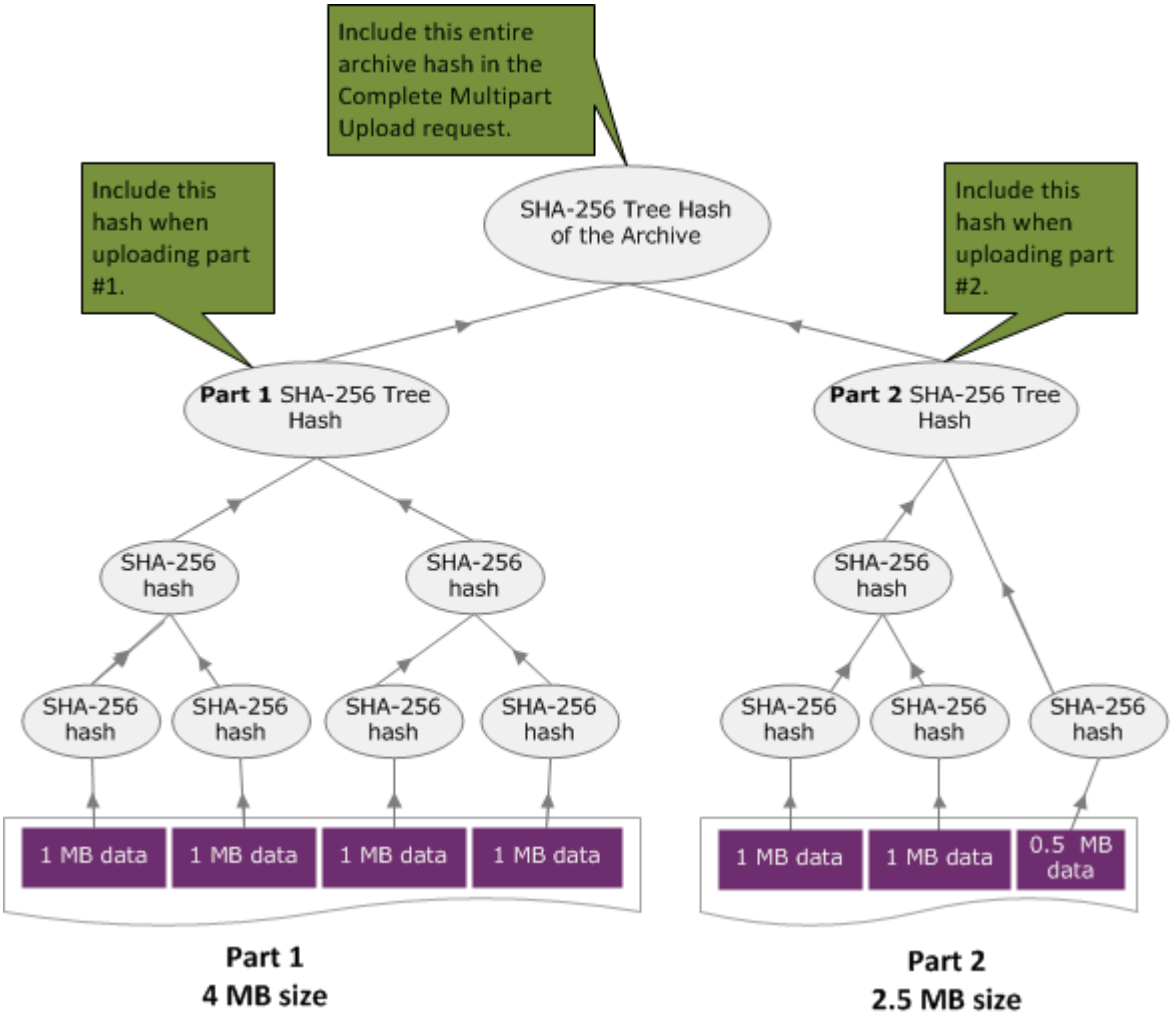
Contoh Hash Pohon 1: Mengunggah arsip dalam satu permintaan

Saat Anda mengunggah arsip dalam permintaan tunggal menggunakan API Unggah Arsip (lihat Mengunggah Arsip (POST archive)), muatan permintaan menyertakan seluruh arsip. Dengan demikian, Anda harus menyertakan hash pohon seluruh arsip di header permintaan `x-amz-sha256-tree-hash`. Misalkan Anda ingin mengunggah arsip sebesar 6,5 MB. Diagram berikut menggambarkan proses pembuatan hash SHA-256 arsip. Anda membaca arsip dan menghitung hash SHA-256 untuk setiap potongan 1 MB. Anda juga menghitung hash untuk 0,5 MB data yang tersisa, lalu membangun pohon seperti yang diuraikan dalam prosedur sebelumnya.



Contih Hash Pohon 2: Mengunggah arsip menggunakan unggahan multipart

Proses komputasi hash pohon saat mengunggah arsip menggunakan unggahan multipart sama seperti saat mengunggah arsip dalam permintaan tunggal. Satu-satunya perbedaan adalah dalam unggahan multipart, Anda hanya mengunggah sebagian arsip dalam setiap permintaan (menggunakan API Mengunggah Bagian (PUT UploadID)), dan karena itu, Anda memberikan checksum hanya bagian dalam header permintaan `x-amz-sha256-tree-hash`. Namun, setelah Anda mengunggah semua bagian, Anda harus mengirim permintaan Complete Multipart Upload (Selesaikan Unggahan Multipart) (lihat Menyelesaikan Unggahan Multipart (POST uploadID)) dengan pohon hash seluruh arsip di header permintaan `x-amz-sha256-tree-hash`.



Menghitung Hash Pohon File

Algoritma yang ditampilkan di sini dipilih untuk tujuan demonstrasi. Anda dapat mengoptimalkan kode yang diperlukan untuk skenario implementasi Anda. Jika Anda menggunakan Amazon SDK untuk memprogram melawan Amazon Glacier (Amazon Glacier), perhitungan hash pohon dilakukan untuk Anda dan Anda hanya perlu memberikan referensi file.

Example 1: Contoh Java

Contoh berikut menunjukkan bagaimana menghitung hash SHA256 pohon dari file menggunakan Java. Anda dapat menjalankan contoh ini dengan menyediakan lokasi file sebagai argumen atau Anda dapat menggunakan metode `TreeHashExample.computeSHA256TreeHash` langsung dari kode Anda.

```
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

public class TreeHashExample {

static final int ONE_MB = 1024 * 1024;

    /**
     * Compute the Hex representation of the SHA-256 tree hash for the specified
     * File
     *
     * @param args
     *            args[0]: a file to compute a SHA-256 tree hash for
     */
    public static void main(String[] args) {

        if (args.length < 1) {
            System.err.println("Missing required filename argument");
            System.exit(-1);
        }

        File inputFile = new File(args[0]);
        try {

            byte[] treeHash = computeSHA256TreeHash(inputFile);
```

```
            System.out.printf("SHA-256 Tree Hash = %s\n", toHex(treeHash));

        } catch (IOException ioe) {
            System.err.format("Exception when reading from file %s: %s", inputFile,
                    ioe.getMessage());
            System.exit(-1);

        } catch (NoSuchAlgorithmException nsae) {
            System.err.format("Cannot locate MessageDigest algorithm for SHA-256: %s",
                    nsae.getMessage());
            System.exit(-1);
        }
    }

    /**
     * Computes the SHA-256 tree hash for the given file
     *
     * @param inputFile
     *            a File to compute the SHA-256 tree hash for
     * @return a byte[] containing the SHA-256 tree hash
     * @throws IOException
     *             Thrown if there's an issue reading the input file
     * @throws NoSuchAlgorithmException
     */
    public static byte[] computeSHA256TreeHash(File inputFile) throws IOException,
            NoSuchAlgorithmException {

        byte[][] chunkSHA256Hashes = getChunkSHA256Hashes(inputFile);
        return computeSHA256TreeHash(chunkSHA256Hashes);
    }

    /**
     * Computes a SHA256 checksum for each 1 MB chunk of the input file. This
     * includes the checksum for the last chunk even if it is smaller than 1 MB.
     *
     * @param file
     *            A file to compute checksums on
     * @return a byte[][] containing the checksums of each 1 MB chunk
     * @throws IOException
     *             Thrown if there's an IOException when reading the file
     * @throws NoSuchAlgorithmException
     *             Thrown if SHA-256 MessageDigest can't be found
     */
    public static byte[][] getChunkSHA256Hashes(File file) throws IOException,
```

```
            NoSuchAlgorithmException {

        MessageDigest md = MessageDigest.getInstance("SHA-256");

        long numChunks = file.length() / ONE_MB;
        if (file.length() % ONE_MB > 0) {
            numChunks++;
        }

        if (numChunks == 0) {
            return new byte[][] { md.digest() };
        }

        byte[][] chunkSHA256Hashes = new byte[(int) numChunks][];
        FileInputStream fileStream = null;

        try {
            fileStream = new FileInputStream(file);
            byte[] buff = new byte[ONE_MB];

            int bytesRead;
            int idx = 0;
            int offset = 0;

            while ((bytesRead = fileStream.read(buff, offset, ONE_MB)) > 0) {
                md.reset();
                md.update(buff, 0, bytesRead);
                chunkSHA256Hashes[idx++] = md.digest();
                offset += bytesRead;
            }

            return chunkSHA256Hashes;

        } finally {
            if (fileStream != null) {
                try {
                    fileStream.close();
                } catch (IOException ioe) {
                    System.err.printf("Exception while closing %s.\n %s",
file.getName(),
                            ioe.getMessage());
                }
            }
        }
```

```
    }

    /**
     * Computes the SHA-256 tree hash for the passed array of 1 MB chunk
     * checksums.
     *
     * This method uses a pair of arrays to iteratively compute the tree hash
     * level by level. Each iteration takes two adjacent elements from the
     * previous level source array, computes the SHA-256 hash on their
     * concatenated value and places the result in the next level's destination
     * array. At the end of an iteration, the destination array becomes the
     * source array for the next level.
     *
     * @param chunkSHA256Hashes
     *            An array of SHA-256 checksums
     * @return A byte[] containing the SHA-256 tree hash for the input chunks
     * @throws NoSuchAlgorithmException
     *             Thrown if SHA-256 MessageDigest can't be found
     */
    public static byte[] computeSHA256TreeHash(byte[][] chunkSHA256Hashes)
            throws NoSuchAlgorithmException {

        MessageDigest md = MessageDigest.getInstance("SHA-256");

        byte[][] prevLvlHashes = chunkSHA256Hashes;

        while (prevLvlHashes.length > 1) {

            int len = prevLvlHashes.length / 2;
            if (prevLvlHashes.length % 2 != 0) {
                len++;
            }

            byte[][] currLvlHashes = new byte[len][];

            int j = 0;
            for (int i = 0; i < prevLvlHashes.length; i = i + 2, j++) {

                // If there are at least two elements remaining
                if (prevLvlHashes.length - i > 1) {

                    // Calculate a digest of the concatenated nodes
                    md.reset();
                    md.update(prevLvlHashes[i]);
```

```
                    md.update(prevLvlHashes[i + 1]);
                    currLvlHashes[j] = md.digest();

                } else { // Take care of remaining odd chunk
                    currLvlHashes[j] = prevLvlHashes[i];
                }
            }

            prevLvlHashes = currLvlHashes;
        }

        return prevLvlHashes[0];
    }

    /**
     * Returns the hexadecimal representation of the input byte array
     *
     * @param data
     *            a byte[] to convert to Hex characters
     * @return A String containing Hex characters
     */
    public static String toHex(byte[] data) {
        StringBuilder sb = new StringBuilder(data.length * 2);

        for (int i = 0; i < data.length; i++) {
            String hex = Integer.toHexString(data[i] & 0xFF);

            if (hex.length() == 1) {
                // Append leading zero.
                sb.append("0");
            }
            sb.append(hex);
        }
        return sb.toString().toLowerCase();
    }
}
```

Example 2: Contoh .NET C#

Contoh berikut menunjukkan bagaimana menghitung hash SHA256 pohon dari sebuah file. Anda dapat menjalankan contoh ini dengan menyediakan lokasi file sebagai argumen.

```
using System;
```

```
using System.IO;

using System.Security.Cryptography;

namespace ExampleTreeHash
{
    class Program
    {
        static int ONE_MB = 1024 * 1024;

        /**
        * Compute the Hex representation of the SHA-256 tree hash for the
        * specified file
        *
        * @param args
        *            args[0]: a file to compute a SHA-256 tree hash for
        */
        public static void Main(string[] args)
        {
            if (args.Length < 1)
            {
                Console.WriteLine("Missing required filename argument");
                Environment.Exit(-1);
            }
            FileStream inputFile = File.Open(args[0], FileMode.Open, FileAccess.Read);
            try
            {
                byte[] treeHash = ComputeSHA256TreeHash(inputFile);
                Console.WriteLine("SHA-256 Tree Hash = {0}",
 BitConverter.ToString(treeHash).Replace("-", "").ToLower());
                Console.ReadLine();
                Environment.Exit(-1);
            }
            catch (IOException ioe)
            {
                Console.WriteLine("Exception when reading from file {0}: {1}",
                    inputFile, ioe.Message);
                Console.ReadLine();
                Environment.Exit(-1);
            }
            catch (Exception e)
            {
                Console.WriteLine("Cannot locate MessageDigest algorithm for SHA-256:
 {0}",
```

```
                e.Message);
            Console.WriteLine(e.GetType());
            Console.ReadLine();
            Environment.Exit(-1);
        }
        Console.ReadLine();
    }


    /**
     * Computes the SHA-256 tree hash for the given file
     *
     * @param inputFile
     *            A file to compute the SHA-256 tree hash for
     * @return a byte[] containing the SHA-256 tree hash
     */
    public static byte[] ComputeSHA256TreeHash(FileStream inputFile)
    {
        byte[][] chunkSHA256Hashes = GetChunkSHA256Hashes(inputFile);
        return ComputeSHA256TreeHash(chunkSHA256Hashes);
    }


    /**
     * Computes a SHA256 checksum for each 1 MB chunk of the input file. This
     * includes the checksum for the last chunk even if it is smaller than 1 MB.
     *
     * @param file
     *            A file to compute checksums on
     * @return a byte[][] containing the checksums of each 1MB chunk
     */
    public static byte[][] GetChunkSHA256Hashes(FileStream file)
    {
        long numChunks = file.Length / ONE_MB;
        if (file.Length % ONE_MB > 0)
        {
            numChunks++;
        }

        if (numChunks == 0)
        {
            return new byte[][] { CalculateSHA256Hash(null, 0) };
        }
        byte[][] chunkSHA256Hashes = new byte[(int)numChunks][];
```

```
            try
            {
                byte[] buff = new byte[ONE_MB];

                int bytesRead;
                int idx = 0;

                while ((bytesRead = file.Read(buff, 0, ONE_MB)) > 0)
                {
                    chunkSHA256Hashes[idx++] = CalculateSHA256Hash(buff, bytesRead);
                }
                return chunkSHA256Hashes;
            }
            finally
            {
                if (file != null)
                {
                    try
                    {
                        file.Close();
                    }
                    catch (IOException ioe)
                    {
                        throw ioe;
                    }
                }
            }

        }

        /**
         * Computes the SHA-256 tree hash for the passed array of 1MB chunk
         * checksums.
         *
         * This method uses a pair of arrays to iteratively compute the tree hash
         * level by level. Each iteration takes two adjacent elements from the
         * previous level source array, computes the SHA-256 hash on their
         * concatenated value and places the result in the next level's destination
         * array. At the end of an iteration, the destination array becomes the
         * source array for the next level.
         *
         * @param chunkSHA256Hashes
         *            An array of SHA-256 checksums
```

```
         * @return A byte[] containing the SHA-256 tree hash for the input chunks
         */
        public static byte[] ComputeSHA256TreeHash(byte[][] chunkSHA256Hashes)
        {
            byte[][] prevLvlHashes = chunkSHA256Hashes;
            while (prevLvlHashes.GetLength(0) > 1)
            {

                int len = prevLvlHashes.GetLength(0) / 2;
                if (prevLvlHashes.GetLength(0) % 2 != 0)
                {
                    len++;
                }

                byte[][] currLvlHashes = new byte[len][];

                int j = 0;
                for (int i = 0; i < prevLvlHashes.GetLength(0); i = i + 2, j++)
                {

                    // If there are at least two elements remaining
                    if (prevLvlHashes.GetLength(0) - i > 1)
                    {

                        // Calculate a digest of the concatenated nodes
                        byte[] firstPart = prevLvlHashes[i];
                        byte[] secondPart = prevLvlHashes[i + 1];
                        byte[] concatenation = new byte[firstPart.Length +
secondPart.Length];
                        System.Buffer.BlockCopy(firstPart, 0, concatenation, 0,
firstPart.Length);
                        System.Buffer.BlockCopy(secondPart, 0, concatenation,
firstPart.Length, secondPart.Length);

                        currLvlHashes[j] = CalculateSHA256Hash(concatenation,
concatenation.Length);

                    }
                    else
                    { // Take care of remaining odd chunk
                        currLvlHashes[j] = prevLvlHashes[i];
                    }
                }
```

```
                prevLvlHashes = currLvlHashes;
            }

            return prevLvlHashes[0];
        }

        public static byte[] CalculateSHA256Hash(byte[] inputBytes, int count)
        {
            SHA256 sha256 = System.Security.Cryptography.SHA256.Create();
            byte[] hash = sha256.ComputeHash(inputBytes, 0, count);
            return hash;
        }
    }
}
```

Menerima Checksum Saat Mengunduh Data

Ketika Anda mengambil arsip menggunakan API Mulai Tugas (lihat Memulai Tugas (POST jobs)), Anda dapat secara opsional menentukan rentang untuk mengambil arsip. Demikian pula, ketika Anda mengunduh data Anda menggunakan API Dapatkan Hasil Tugas (lihat Mendapatkan Output Tugas (GET output)), Anda dapat secara opsional menentukan rentang data yang akan diunduh. Ada dua karakteristik dari rentang ini yang penting untuk dipahami saat Anda mengambil dan mengunduh data arsip Anda. Rentang yang diambil harus megabyte selaras dengan arsip. Kedua rentang yang diambil dan kisaran yang diunduh harus hash pohon yang diseleraskan untuk menerima nilai checksum saat Anda mengunduh data Anda. Definisi dua tipe penyelarasan rentang ini adalah sebagai berikut:

- Megabyte selaras - Rentang [StartByte, EndBytes] adalah megabyte (1024*1024) sejajar bila habis StartBytesdibagi 1 MB dan EndBytesditambah 1 habis dibagi 1 MB atau sama dengan akhir arsip yang ditentukan (ukuran byte arsip dikurangi 1). Rentang yang digunakan dalam API Mulai Tugas, jika ditentukan, harus megabyte selaras.
- Tree-hash aligned - Rentang [StartBytes, EndBytes] adalah hash pohon yang disejajarkan sehubungan dengan arsip jika dan hanya jika akar hash pohon yang dibangun di atas rentang setara dengan simpul di hash pohon dari seluruh arsip. Kedua rentang yang diambil dan kisaran yang diunduh harus hash pohon selaras agar dapat menerima nilai checksum untuk data yang Anda unduh. Untuk contoh rentang dan hubungannya dengan hash pohon arsip, lihat Contoh Hash Pohon: Mengambil rentang arsip yang merupakan hash pohon selaras.

Perhatikan bahwa rentang yang merupakan hash pohon selaras juga megabyte selaras. Namun, rentang megabyte selaras belum tentu hash pohon selaras.

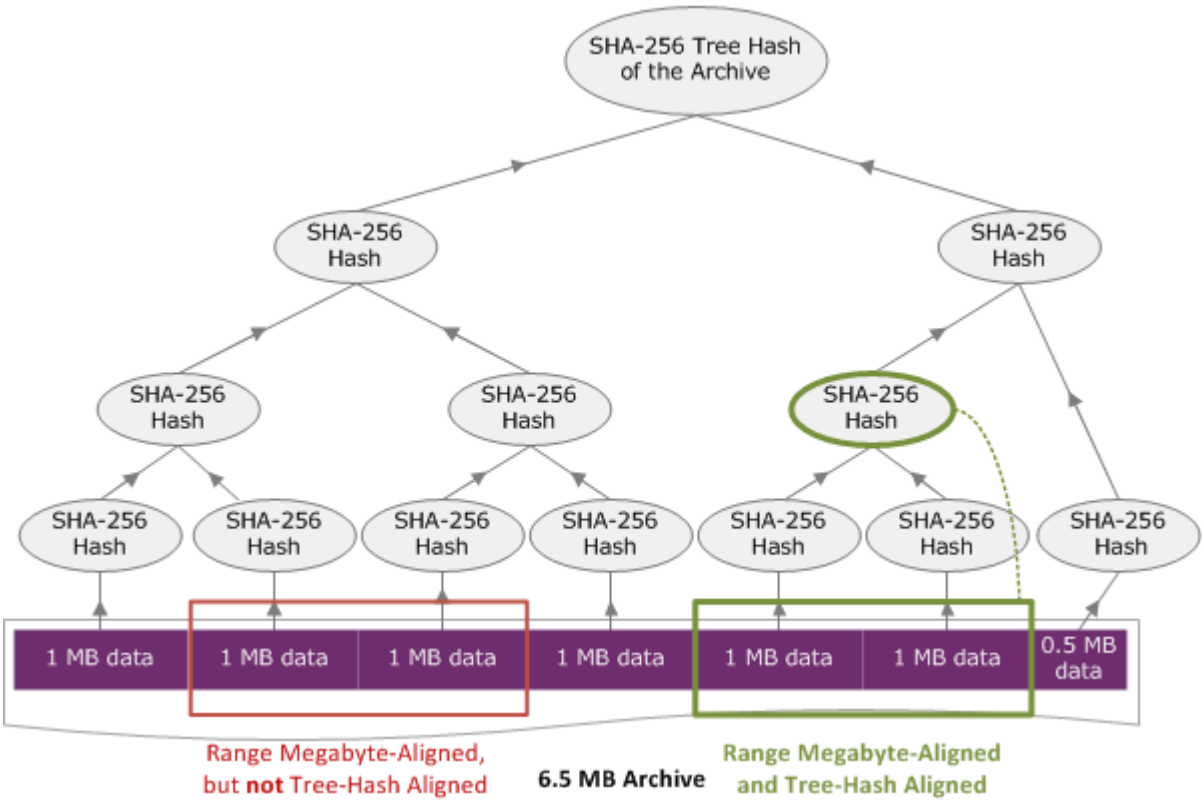
Kasus berikut menjelaskan ketika Anda menerima nilai checksum saat Anda mengunduh data arsip Anda:

- Jika Anda tidak menentukan rentang yang diambil dalam permintaan Initiate Job (Mulai Tugas) dan Anda mengunduh seluruh arsip dalam Permintaan Get Job (Dapatkan Tugas).
- Jika Anda tidak menentukan rentang yang diambil dalam permintaan Initiate Job (Mulai Tugas) dan Anda menentukan rentang hash pohon selaras yang diunduh dalam Permintaan Get Job (Dapatkan Tugas).
- Jika Anda menentukan rentang hash pohon selaras yang diambil dalam permintaan Initiate Job (Mulai Tugas) dan Anda mengunduh seluruh rentang dalam Permintaan Get Job (Dapatkan Tugas).
- Jika Anda menentukan rentang hash pohon selaras yang diambil dalam permintaan Initiate Job (Mulai Tugas) dan Anda menentukan rentang hash tree selaras yang diunduh di Permintaan Get Job (Dapatkan Tugas).

Jika Anda menentukan rentang yang diambil dalam permintaan Initiate Job (Mulai Tugas) yang bukan hash pohon selaras, Anda masih bisa mendapatkan data arsip Anda, tetapi tidak ada nilai checksum yang dikembalikan ketika Anda mengunggah data dalam Permintaan Get Job (Dapatkan Tugas).

Contoh Hash Pohon: Mengambil rentang arsip yang merupakan hash pohon selaras

Misalkan Anda memiliki arsip sebesar 6,5 MB di vault Anda dan Anda ingin mengambil arsip sebesar 2 MB. Cara Anda menentukan rentang 2 MB dalam permintaan Initiate Job (Mulai Tugas) menentukan apakah Anda menerima nilai checksum data saat Anda mengunduh data. Diagram berikut menggambarkan dua rentang 2 MB untuk arsip berukuran 6,5 MB yang dapat Anda unduh. Kedua rentang adalah megabyte selaras, tetapi hanya satu yang merupakan hash pohon selaras.



Spesifikasi Rentang Hash Pohon Selaras

Bagian ini memberikan spesifikasi yang tepat untuk apa yang merupakan rentang hash pohon selaras. Rentang hash pohon selaras penting ketika Anda mengunduh sebagian arsip dan Anda menentukan rentang data yang diambil dan rentang yang diunduh dari data yang diambil. Jika kedua rentang ini adalah hash pohon selaras, Anda akan menerima data checksum ketika Anda mengunduh data.

Rentang [A, B] adalah hash pohon selaras yang berkenaan dengan arsip jika dan hanya jika hash pohon baru dibangun di atas [A, B], akar hash pohon dari rentang tersebut sama dengan node di hash pohon seluruh arsip. Anda dapat melihat ini ditunjukkan dalam diagram di Contoh Hash Pohon: Mengambil rentang arsip yang merupakan hash pohon selaras. Pada bagian ini, kami menyediakan spesifikasi untuk penyelarasan hash pohon.

Pertimbangkan [P, Q) sebagai kueri rentang untuk arsip N megabyte (MB) dan P serta Q adalah kelipatan satu MB. Perhatikan bahwa rentang inklusif sebenarnya adalah [P MB, Q MB – 1 byte], tetapi agar ringkas, kami menampilkannya sebagai [P, Q). Dengan pertimbangan ini, maka

- Jika P adalah angka ganjil, hanya ada satu kemungkinan rentang hash pohon selaras—yaitu [P, P + 1 MB).

- Jika P adalah bilangan genap dan k adalah jumlah maksimum, dengan P yang dapat ditulis sebagai 2k * X, ada rentang hash pohon selaras k paling banyak yang dimulai dengan P. X adalah bilangan bulat yang lebih besar dari 0. Rentang hash pohon selaras masuk dalam kategori berikut:
  - Untuk setiap i, dengan (0 <= i <= k) dan dengan $P + 2^i < N$, maka $[P, Q + 2^i)$ adalah rentang hash pohon selaras.
  - P = 0 adalah kasus khusus dengan A = 2[lgN]*0

Respons Kesalahan

Jika terjadi kesalahan, API mengembalikan salah satu pengecualian berikut:

Kode	Deskripsi	Kode Status HTTP	Tipe
`AccessDeniedException`	Dikembalikan jika ada upaya untuk mengakses sumber daya yang tidak diizinkan oleh kebijakan AWS Identity and Access Management (IAM), atau Akun AWS ID yang salah digunakan dalam URI permintaan. Untuk informasi selengkapnya, lihat Identity and Access Management untuk Amazon Glacier.	`403 Forbidden`	Klien
`BadRequest`	Dikembalikan jika permintaan tidak dapat diproses.	`400 Bad Request`	Klien
`ExpiredTokenException`	Dikembalikan jika token keamanan yang digunakan dalam permintaan telah berakhir.	`403 Forbidden`	Klien
`InsufficientCapacityException`	Dikembalikan jika kapasitas tidak memadai untuk memproses permintaan yang dipercepat. Kesalahan ini hanya berlaku	`503 Service Unavailable`	Server

Kode	Deskripsi	Kode Status HTTP	Tipe
	untuk pengambilan yang dipercepat dan bukan untuk pengambilan standar atau massal.		
`InvalidParameterValueException`	Dikembalikan jika parameter permintaan salah ditentukan.	`400 Bad Request`	Klien
`InvalidSignatureException`	Dikembalikan jika tanda tangan permintaan tidak valid.	`403 Forbidden`	Klien
`LimitExceededException`	Dikembalikan jika hasil permintaan di salah satu batas berikut terlampaui, batas vault, batas tanda, atau batas kapasitas yang disediakan.	`400 Bad Request`	Klien
`MissingAuthenticationTokenException`	Dikembalikan jika tidak tidak ditemukan data autentikasi untuk permintaan.	`400 Bad Request`	Klien
`MissingParameterValueException`	Dikembalikan jika header atau parameter yang diperlukan hilang dari permintaan.	`400 Bad Request`	Klien
`PolicyEnforcedException`	Dikembalikan jika tugas pengambilan akan melebihi batas tingkat pengambilan kebijakan data saat ini. Untuk informasi selengkapnya tentang kebijakan pengambilan data, lihat Kebijakan Pengambilan Data Amazon Glacier.	`400 Bad Request`	Klien

Kode	Deskripsi	Kode Status HTTP	Tipe
`ResourceNotFoundException`	Dikembalikan jika sumber daya yang ditentukan seperti vault, ID unggahan, atau ID tugas tidak ada.	`404 Not Found`	Klien
`RequestTimeoutException`	Dikembalikan jika mengunggah arsip dan Amazon Glacier (Amazon Glacier) habis waktu saat menerima unggahan.	`408 Request Timeout`	Klien
`SerializationException`	Dikembalikan jika isi permintaan tidak valid. Jika termasuk muatan JSON, periksa bahwa muatan JSON tersebut memiliki bentuk yang baik.	`400 Bad Request`	Klien
`ServiceUnavailableException`	Dikembalikan jika layanan tidak dapat menyelesaikan permintaan.	`500 Internal Server Error`	Server
`ThrottlingException`	Dikembalikan jika Anda perlu mengurangi tingkat permintaan Anda ke Amazon Glacier.	`400 Bad Request`	Klien
`UnrecognizedClientException`	Dikembalikan jika Access Key ID atau token keamanan tidak valid.	`400 Bad Request`	Klien

Berbagai Amazon APIs Glacier mengembalikan pengecualian yang sama, tetapi dengan pesan pengecualian yang berbeda untuk membantu Anda memecahkan masalah kesalahan tertentu yang dihadapi.

Amazon Glacier mengembalikan informasi kesalahan di badan respons. Contoh berikut menunjukkan beberapa respons kesalahan.

Contoh 1: Menjelaskan permintaan Tugas dengan ID tugas yang tidak ada

Misalkan Anda mengirim permintaan Mendeskripsikan Tugas (GET JobID) untuk tugas yang tidak ada. Artinya, Anda menentukan ID tugas yang tidak ada.

```
GET /-/vaults/examplevault/jobs/HkF9p6o7yjhFx-
K3CGl6fuSm6VzW9T7esGQfco8nUXVYwS0jlb5gq1JZ55yHgt5vP54ZShjoQzQVVEXAMPLEbadJobID HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
Date: 20170210T120000Z
x-amz-glacier-version: 2012-06-01
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
```

Sebagai tanggapan, Amazon Glacier mengembalikan respons kesalahan berikut.

```
HTTP/1.1 404 Not Found
x-amzn-RequestId: AAABaZ9N92Iiyv4N7sru3ABEpSQkuFtmH3NP6aAC51ixfjg
Content-Type: application/json
Content-Length: 185
Date: Wed, 10 Feb 2017 12:00:00 GMT
{
  "code": "ResourceNotFoundException",
  "message": "The job ID was not found: HkF9p6o7yjhFx-
K3CGl6fuSm6VzW9T7esGQfco8nUXVYwS0jlb5gq1JZ55yHgt5vP54ZShjoQzQVVEXAMPLEbadJobID",
  "type": "Client"
  }
```

Di mana:

Kode

Salah satu pengecualian umum.

Tipe: String

Pesan

Deskripsi umum tentang kondisi kesalahan yang khusus untuk API yang mengembalikan kesalahan.

Tipe: String

Jenis

Sumber kesalahan. Bidang dapat menjadi salah satu dari nilai berikut: `Client`, `Server`, atau `Unknown`.

Tipe: String.

Perhatikan hal berikut dalam respons sebelumnya:

- Untuk respons kesalahan, Amazon Glacier mengembalikan nilai kode status dan. `4xx` `5xx` Dalam contoh ini, kode status adalah `404 Not Found`.
- Nilai header `Content-Type` `application/json` menunjukkan JSON dalam isi
- JSON dalam isi menyediakan informasi kesalahan.

Dalam permintaan sebelumnya, bukan ID tugas yang buruk, misalkan Anda menentukan vault yang tidak ada. Respons mengembalikan pesan yang berbeda.

```
HTTP/1.1 404 Not Found
x-amzn-RequestId: AAABBeC9Zw0rp_5D0L8VfB3FA_WlTupqTKAUehMcPhdgni0
Content-Type: application/json
Content-Length: 154
Date: Wed, 10 Feb 2017 12:00:00 GMT
{
  "code": "ResourceNotFoundException",
  "message": "Vault not found for ARN: arn:aws:glacier:us-west-2:012345678901:vaults/
examplevault",
  "type": "Client"
}
```

Contoh 2: Mencantumkan permintaan Tugas dengan nilai yang tidak valid untuk parameter permintaan

Dalam contoh ini Anda mengirim permintaan Mencantumkan Tugas (GET jobs) untuk mengambil tugas vault dengan `statuscode` tertentu, dan Anda memberikan nilai `statuscode` `finished` yang salah, bukan nilai `InProgress` ,`Succeeded`, atau `Failed` yang dapat diterima.

```
GET /-/vaults/examplevault/jobs?statuscode=finished HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
```

```
Date: 20170210T120000Z
x-amz-glacier-version: 2012-06-01
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
```

Amazon Glacier mengembalikan pesan `InvalidParameterValueException` yang sesuai.

```
HTTP/1.1 400 Bad Request
x-amzn-RequestId: AAABaZ9N92Iiyv4N7sru3ABEpSQkuFtmH3NP6aAC51ixfjg
Content-Type: application/json
Content-Length: 141
Date: Wed, 10 Feb 2017 12:00:00 GMT
{
  "code": "InvalidParameterValueException",
  "message": "The job status code is not valid: finished",
  "type: "Client"
}
```

Operasi Vault

Berikut ini adalah operasi brankas yang tersedia di Amazon Glacier.

Topik

- Membatalkan Kunci Vault (DELETE lock-policy)
- Menambahkan Tanda ke Vault (POST tag add)
- Membuat Vault (PUT vault)
- Menyelesaikan Kunci Vault (POST lockId)
- Menghapus Vault (DELETE vault)
- Menghapus Kebijakan Akses Vault (DELETE access-policy)
- Menghapus Notifikasi Vault (DELETE notification-configuration)
- Menjelaskan Vault (GET vault)
- Mendapatkan Kebijakan Akses Vault (GET access-policy)
- Mendapatkan Kunci Vault (GET lock-policy)
- Mendapatkan Notifikasi Vault (GET notification-configuration)

- Memulai Kunci Vault (POST lock-policy)
- Mencantumkan Tanda untuk Vault (GET tags)
- Mencantumkan Vault (GET vaults)
- Hapus Tanda Dari Vault (POST tags remove)
- Mendapatkan Kebijakan Akses Vault (PUT access-policy)
- Menetapkan Konfigurasi Notifikasi Vault (PUT notification-configuration)

Membatalkan Kunci Vault (DELETE lock-policy)

Deskripsi

Operasi ini menghentikan proses penguncian vault jika kunci vault tidak berada dalam status `Locked`. Jika status kunci vault adalah `Locked` ketika operasi ini diminta, operasi mengembalikan kesalahan `AccessDeniedException`. Menghentikan proses penguncian vault menghapus kebijakan kunci vault dari vault yang ditentukan.

Kunci vault ditetapkan ke dalam status `InProgress` dengan memanggil Memulai Kunci Vault (POST lock-policy). Kunci vault ditetapkan ke dalam status `Locked` dengan memanggil Menyelesaikan Kunci Vault (POST lockId). Anda bisa mendapatkan status kunci vault dengan memanggil Mendapatkan Kunci Vault (GET lock-policy). Untuk informasi selengkapnya tentang proses penguncian vault, lihat Kunci Gletser Amazon Glacier. Untuk informasi selengkapnya tentang kebijakan kunci vault, lihat Kebijakan Vault Lock.

Operasi ini bersifat idempoten. Anda dapat berhasil memanggil operasi ini beberapa kali, jika status kunci vault adalah `InProgress` atau jika tidak ada kebijakan yang terkait dengan vault.

Permintaan

Untuk menghapus kebijakan kunci vault, kirim permintaan `DELETE` HTTP ke URI subsumber daya `lock-policy` vault.

Sintaksis

```
DELETE /AccountId/vaults/vaultName/lock-policy HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: Date
Authorization: SignatureValue
```

```
x-amz-glacier-version: 2012-06-01
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID. Nilai ini harus cocok dengan Akun AWS ID yang terkait dengan kredensional yang digunakan untuk menandatangani permintaan. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menentukan ID akun, jangan sertakan tanda hubung ('-') di ID.

Parameter Permintaan

Operasi ini tidak menggunakan parameter permintaan.

Header Permintaan

Operasi ini hanya menggunakan header permintaan yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header permintaan umum, lihat Header Permintaan Umum.

Isi Permintaan

Operasi ini tidak memiliki isi permintaan.

Respons

Jika kebijakan berhasil dihapus, Amazon Glacier mengembalikan respons. `HTTP 204 No Content`

Sintaksis

```
HTTP/1.1 204 No Content
x-amzn-RequestId: x-amzn-RequestId
Date: Date
```

Header Respons

Operasi ini hanya menggunakan header respons yang umum untuk sebagian besar respons. Untuk informasi tentang header permintaan umum, lihat Header Respons Umum.

Isi Respons

Operasi ini tidak mengembalikan isi respons.

Kesalahan

Untuk informasi tentang pengecualian Amazon Glacier dan pesan kesalahan, lihat. Respons Kesalahan

Contoh

Contoh berikut menunjukkan cara menghentikan proses penguncian vault.

Contoh Permintaan

Dalam contoh ini, permintaan `DELETE` dikirim ke subsumber daya `lock-policy` dari vault bernama **`examplevault`**.

```
DELETE /-/vaults/examplevault/lock-policy HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
x-amz-glacier-version: 2012-06-01
```

Contoh Respons

Jika kebijakan berhasil dihapus Amazon Glacier mengembalikan respons, seperti `HTTP 204 No Content` yang ditunjukkan pada contoh berikut.

```
HTTP/1.1 204 No Content
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:00:00 GMT
```

Bagian Terkait

- Menyelesaikan Kunci Vault (POST lockId)

- Mendapatkan Kunci Vault (GET lock-policy)

- Memulai Kunci Vault (POST lock-policy)

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu Amazon khusus bahasa SDKs, lihat berikut ini:

- AWS Command Line Interface

Menambahkan Tanda ke Vault (POST tag add)

Operasi ini menambahkan tanda tertentu ke vault. Setiap tanda terdiri dari kunci dan nilai. Setiap vault dapat memiliki hingga 50 tanda. Jika permintaan Anda akan menyebabkan batas tanda untuk vault terlampaui, operasi melempar kesalahan `LimitExceededException`.

Jika tanda sudah ada di vault di bawah kunci tertentu, nilai kunci yang ada akan ditimpa. Untuk informasi selengkapnya tentang tanda, lihat Menandai Sumber Daya Amazon Glacier.

Sintaks Permintaan

Untuk menambahkan tag ke vault, kirim permintaan HTTP POST ke tag URI seperti yang ditunjukkan dalam contoh sintaks berikut.

```
POST /AccountId/vaults/vaultName/tags?operation=add HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: Date
Authorization: SignatureValue
Content-Length: Length
x-amz-glacier-version: 2012-06-01

{
   "Tags":
      {
         "string": "string",
         "string": "string"
      }
}
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID. Nilai ini harus cocok dengan Akun AWS ID yang terkait dengan kredensional yang digunakan untuk menandatangani permintaan. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menentukan ID akun, jangan sertakan tanda hubung ('-') di ID.

Parameter Permintaan

Nama	Deskripsi	Wajib
`operation=add`	Parameter string kueri tunggal `operation` dengan nilai `add` untuk membedakannya dari Hapus Tanda Dari Vault (POST tags remove).	Ya

Header Permintaan

Operasi ini hanya menggunakan header permintaan yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header permintaan umum, lihat Header Permintaan Umum.

Isi Permintaan

Isi permintaan berisi bidang JSON berikut.

Tag

Tanda untuk ditambahkan ke vault. Setiap tanda terdiri dari kunci dan nilai. Nilai bisa string kosong.

Tipe: Peta String ke String

Panjang batasan: Panjang minimum 1. Panjang maksimum 10.

Wajib: Ya

Respons

Jika permintaan operasi berhasil, layanan mengembalikan respons `204 No Content` HTTP.

Sintaksis

```
HTTP/1.1 204 No Content
x-amzn-RequestId: x-amzn-RequestId
Date: Date
```

Header Respons

Operasi ini hanya menggunakan header respons yang umum untuk sebagian besar respons. Untuk informasi tentang header permintaan umum, lihat Header Respons Umum.

Isi Respons

Operasi ini tidak mengembalikan isi respons.

Kesalahan

Untuk informasi tentang pengecualian Amazon Glacier dan pesan kesalahan, lihat. Respons Kesalahan

Contoh

Contoh Permintaan

Contoh berikut mengirimkan permintaan HTTP POST dengan tanda untuk ditambahkan ke vault.

```
POST /-/vaults/examplevault/tags?operation=add HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
Content-Length: length
x-amz-glacier-version: 2012-06-01

{
  "Tags":
    {
```

```
        "examplekey1": "examplevalue1",
        "examplekey2": "examplevalue2"
    }
}
```

Contoh Respons

Jika permintaan berhasil Amazon Glacier mengembalikan seperti `HTTP 204 No Content` yang ditunjukkan pada contoh berikut.

```
HTTP/1.1 204 No Content
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:02:00 GMT
```

Bagian Terkait

- Mencantumkan Tanda untuk Vault (GET tags)
- Hapus Tanda Dari Vault (POST tags remove)

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu Amazon khusus bahasa SDKs, lihat berikut ini:

- AWS Command Line Interface

Membuat Vault (PUT vault)

Deskripsi

Operasi ini membuat vault baru dengan nama tertentu. Nama brankas harus unik di dalam AWS Wilayah untuk sebuah Akun AWS. Anda dapat membuat hingga 1.000 vault per akun. Untuk informasi tentang membuat lebih banyak brankas, buka halaman detail produk Amazon Glacier.

Anda harus menggunakan pedoman berikut ketika memberi nama vault.

- Panjang karakter nama dapat antara 1 dan 255.
- Karakter yang diperbolehkan adalah a–z, A–Z, 0–9, '_' (garis bawah), '-' (tanda hubung), dan '.' (titik).

Operasi ini idempoten, Anda dapat mengirim permintaan yang sama beberapa kali dan tidak memiliki efek lebih lanjut setelah pertama kali Amazon Glacier (Amazon Glacier) membuat brankas yang ditentukan.

Permintaan

Sintaksis

Untuk membuat vault, kirim permintaan HTTP PUT ke URI vault yang akan dibuat.

```
PUT /AccountId/vaults/VaultName HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: Date
Authorization: SignatureValue
Content-Length: Length
x-amz-glacier-version: 2012-06-01
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID. Nilai ini harus cocok dengan Akun AWS ID yang terkait dengan kredensional yang digunakan untuk menandatangani permintaan. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menentukan ID akun, jangan sertakan tanda hubung ('-') di ID.

Parameter Permintaan

Operasi ini tidak menggunakan parameter permintaan.

Header Permintaan

Operasi ini hanya menggunakan header permintaan yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header permintaan umum, lihat Header Permintaan Umum.

Isi Permintaan

Isi permintaan untuk operasi ini harus kosong (0 byte).

Respons

Sintaksis

```
HTTP/1.1 201 Created
x-amzn-RequestId: x-amzn-RequestId
Date: Date
Location: Location
```

Header Respons

Respons yang sukses termasuk header respon berikut, selain header respons yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header respons umum, lihat Header Respons Umum.

Nama	Deskripsi
`Location`	Jalur URI relatif dari vault yang dibuat. Tipe: String

Isi Respons

Operasi ini tidak mengembalikan isi respons.

Kesalahan

Untuk informasi tentang pengecualian Amazon Glacier dan pesan kesalahan, lihat. Respons Kesalahan

Contoh

Contoh Permintaan

Contoh berikut mengirimkan permintaan HTTP PUT untuk membuat vault bernama `examplevault`.

```
PUT /-/vaults/examplevault HTTP/1.1
```

```
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
x-amz-glacier-version: 2012-06-01
Content-Length: 0
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
```

Contoh Respons

Amazon Glacier membuat vault dan mengembalikan jalur URI relatif dari vault di header. `Location` ID akun selalu ditampilkan di header `Location` terlepas dari apakah ID akun atau tanda hubung ('-') ditentukan dalam permintaan.

```
HTTP/1.1 201 Created
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:02:00 GMT
Location: /111122223333/vaults/examplevault
```

Bagian Terkait

- Mencantumkan Vault (GET vaults)
- Menghapus Vault (DELETE vault)
- Identity and Access Management untuk Amazon Glacier

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu Amazon khusus bahasa SDKs, lihat berikut ini:

- AWS Command Line Interface

Menyelesaikan Kunci Vault (POST lockId)

Deskripsi

Operasi ini menyelesaikan proses penguncian vault dengan mengalihkan kunci vault dari status `InProgress` ke status `Locked`, yang menyebabkan kebijakan kunci vault menjadi tidak dapat

berubah. Kunci vault ditetapkan ke dalam status `InProgress` dengan memanggil Memulai Kunci Vault (POST lock-policy). Anda bisa mendapatkan status kunci vault dengan memanggil Mendapatkan Kunci Vault (GET lock-policy). Untuk informasi selengkapnya tentang proses penguncian vault, lihat Kunci Gletser Amazon Glacier.

Operasi ini bersifat idempoten. Permintaan ini selalu berhasil jika status kunci vault adalah `Locked` dan ID kunci yang disediakan cocok dengan ID kunci yang awalnya digunakan untuk mengunci vault.

Jika ID kunci yang tidak valid diteruskan dalam permintaan ketika status kunci vault adalah `Locked`, operasi mengembalikan kesalahan `AccessDeniedException`. Jika ID kunci yang tidak valid diteruskan dalam permintaan ketika status kunci vault adalah `InProgress`, operasi menampilkan kesalahan `InvalidParameter`.

Permintaan

Untuk menyelesaikan proses penguncian vault, kirim permintaan `POST` HTTP ke URI subsumber daya `lock-policy` vault dengan ID kunci yang valid.

Sintaksis

```
POST /AccountId/vaults/vaultName/lock-policy/lockId HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: Date
Authorization: SignatureValue
Content-Length: Length
x-amz-glacier-version: 2012-06-01
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID. Nilai ini harus cocok dengan Akun AWS ID yang terkait dengan kredensional yang digunakan untuk menandatangani permintaan. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menentukan ID akun, jangan sertakan tanda hubung ('-') di ID.

Nilai `lockId` adalah ID kunci yang diperoleh dari permintaan Memulai Kunci Vault (POST lock-policy).

Parameter Permintaan

Header Permintaan

Operasi ini hanya menggunakan header permintaan yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header permintaan umum, lihat Header Permintaan Umum.

Isi Permintaan

Operasi ini tidak memiliki isi permintaan.

Respons

Jika permintaan operasi berhasil, layanan mengembalikan respons `204 No Content` HTTP.

Sintaksis

```
HTTP/1.1 204 No Content
x-amzn-RequestId: x-amzn-RequestId
Date: Date
```

Header Respons

Operasi ini hanya menggunakan header respons yang umum untuk sebagian besar respons. Untuk informasi tentang header permintaan umum, lihat Header Respons Umum.

Isi Respons

Operasi ini tidak mengembalikan isi respons.

Kesalahan

Untuk informasi tentang pengecualian Amazon Glacier dan pesan kesalahan, lihat. Respons Kesalahan

Contoh

Contoh Permintaan

Contoh berikut mengirimkan permintaan HTTP POST dengan ID kunci untuk menyelesaikan proses penguncian vault.

```
POST /-/vaults/examplevault/lock-policy/AE863rKkWZU53SLW5be4DUcW HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
```

```
x-amz-Date: 20170210T120000Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
Content-Length: length
x-amz-glacier-version: 2012-06-01
```

Contoh Respons

Jika permintaan berhasil, Amazon Glacier (Amazon Glacier) mengembalikan respons, seperti yang ditunjukkan pada contoh `HTTP 204 No Content` berikut.

```
HTTP/1.1 204 No Content
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:02:00 GMT
```

Bagian Terkait

- Membatalkan Kunci Vault (DELETE lock-policy)
- Mendapatkan Kunci Vault (GET lock-policy)
- Memulai Kunci Vault (POST lock-policy)

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu Amazon khusus bahasa SDKs, lihat berikut ini:

- AWS Command Line Interface

Menghapus Vault (DELETE vault)

Deskripsi

Operasi ini menghapus vault. Amazon Glacier (Amazon Glacier) akan menghapus lemari besi hanya jika tidak ada arsip di lemari besi sesuai inventaris terakhir dan belum ada penulisan ke lemari besi

sejak inventaris terakhir. Jika salah satu dari kondisi ini tidak terpenuhi, penghapusan vault gagal (yaitu, brankas tidak dihapus) dan Amazon Glacier mengembalikan kesalahan.

Anda dapat menggunakan Menjelaskan Vault (GET vault) operasi yang menyediakan informasi brankas, termasuk jumlah arsip di brankas; namun, informasi tersebut didasarkan pada inventaris brankas Amazon Glacier yang terakhir dihasilkan.

Operasi ini bersifat idempoten.

> **Note**
>
> Ketika Anda menghapus vault, kebijakan akses vault yang dilampirkan di vault juga dihapus. Untuk informasi selengkapnya tentang kebijakan akses vault, lihat Kebijakan Akses Vault.

Permintaan

Untuk menghapus vault, kirim permintaan `DELETE` ke URI sumber daya vault.

Sintaksis

```
DELETE /AccountId/vaults/VaultName HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: Date
Authorization: SignatureValue
x-amz-glacier-version: 2012-06-01
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID akun yang memiliki brankas. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menggunakan ID akun, jangan sertakan tanda hubung ('-') di ID.

Parameter Permintaan

Operasi ini tidak menggunakan parameter permintaan.

Header Permintaan

Operasi ini hanya menggunakan header permintaan yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header permintaan umum, lihat Header Permintaan Umum.

Isi Permintaan

Operasi ini tidak memiliki isi permintaan.

Respons

Sintaksis

```
HTTP/1.1 204 No Content
x-amzn-RequestId: x-amzn-RequestId
Date: Date
```

Header Respons

Operasi ini hanya menggunakan header respons yang umum untuk sebagian besar respons. Untuk informasi tentang header permintaan umum, lihat Header Respons Umum.

Isi Respons

Operasi ini tidak mengembalikan isi respons.

Kesalahan

Untuk informasi tentang pengecualian Amazon Glacier dan pesan kesalahan, lihat. Respons Kesalahan

Contoh

Contoh Permintaan

Contoh berikut menghapus vault bernama `examplevault`. Contoh permintaan adalah permintaan `DELETE` ke URI dari sumber daya (vault) untuk menghapus.

```
DELETE /-/vaults/examplevault HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
```

```
x-amz-glacier-version: 2012-06-01
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
```

Contoh Respons

```
HTTP/1.1 204 No Content
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:02:00 GMT
```

Bagian Terkait

- Membuat Vault (PUT vault)
- Mencantumkan Vault (GET vaults)
- Memulai Tugas (POST jobs)
- Identity and Access Management untuk Amazon Glacier

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu Amazon khusus bahasa SDKs, lihat berikut ini:

- AWS Command Line Interface

Menghapus Kebijakan Akses Vault (DELETE access-policy)

Deskripsi

Operasi ini akan menghapus kebijakan akses yang terkait dengan vault yang ditentukan. Operasi pada akhirnya konsisten—yaitu, mungkin perlu beberapa waktu bagi Amazon Glacier (Amazon Glacier) untuk sepenuhnya menghapus kebijakan akses, dan Anda mungkin masih melihat efek kebijakan untuk waktu yang singkat setelah Anda mengirim permintaan penghapusan.

Operasi ini bersifat idempoten. Anda dapat memanggil hapus beberapa kali, bahkan jika tidak ada kebijakan yang terkait dengan vault. Untuk informasi selengkapnya tentang kebijakan akses vault, lihat Kebijakan Akses Vault.

Permintaan

Untuk menghapus kebijakan akses vault saat ini, kirim permintaan `DELETE` HTTP ke URI subsumber daya `access-policy` vault.

Sintaksis

```
DELETE /AccountId/vaults/vaultName/access-policy HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: Date
Authorization: SignatureValue
x-amz-glacier-version: 2012-06-01
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID akun yang memiliki brankas. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menggunakan ID akun, jangan sertakan tanda hubung ('-') di ID.

Parameter Permintaan

Operasi ini tidak menggunakan parameter permintaan.

Header Permintaan

Operasi ini hanya menggunakan header permintaan yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header permintaan umum, lihat Header Permintaan Umum.

Isi Permintaan

Operasi ini tidak memiliki isi permintaan.

Respons

Sebagai tanggapan, Amazon Glacier `204 No Content` kembali jika kebijakan berhasil dihapus.

Sintaksis

```
HTTP/1.1 204 No Content
x-amzn-RequestId: x-amzn-RequestId
Date: Date
```

Header Respons

Operasi ini hanya menggunakan header respons yang umum untuk sebagian besar respons. Untuk informasi tentang header permintaan umum, lihat Header Respons Umum.

Isi Respons

Operasi ini tidak mengembalikan isi respons.

Kesalahan

Untuk informasi tentang pengecualian Amazon Glacier dan pesan kesalahan, lihat. Respons Kesalahan

Contoh

Contoh berikut menunjukkan cara menghapus kebijakan akses vault.

Contoh Permintaan

Dalam contoh ini, permintaan `DELETE` dikirim ke subsumber daya `access-policy` dari vault bernama **`examplevault`**.

```
DELETE /-/vaults/examplevault/access-policy HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
x-amz-glacier-version: 2012-06-01
```

Contoh Respons

Sebagai tanggapan, jika kebijakan berhasil dihapus Amazon Glacier mengembalikan seperti `204 No Content` yang ditunjukkan pada contoh berikut.

```
HTTP/1.1 204 No Content
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:00:00 GMT
```

Bagian Terkait

- Mendapatkan Kebijakan Akses Vault (GET access-policy)
- Mendapatkan Kebijakan Akses Vault (PUT access-policy)

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu Amazon khusus bahasa SDKs, lihat berikut ini:

- AWS Command Line Interface

Menghapus Notifikasi Vault (DELETE notification-configuration)

Deskripsi

Operasi ini menghapus set konfigurasi notifikasi untuk vault Menetapkan Konfigurasi Notifikasi Vault (PUT notification-configuration). Operasi ini pada akhirnya konsisten—yaitu, mungkin perlu beberapa waktu untuk Amazon Glacier (Amazon Glacier) untuk sepenuhnya menonaktifkan notifikasi, dan Anda mungkin masih menerima beberapa pemberitahuan untuk waktu yang singkat setelah Anda mengirim permintaan penghapusan.

Permintaan

Untuk menghapus konfigurasi notifikasi vault, kirim permintaan `DELETE` ke subsumber daya `notification-configuration` vault.

Sintaksis

```
DELETE /AccountId/vaults/VaultName/notification-configuration HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: Date
```

```
Authorization: SignatureValue
x-amz-glacier-version: 2012-06-01
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID akun yang memiliki brankas. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menggunakan ID akun, jangan sertakan tanda hubung ('-') di ID.

Parameter Permintaan

Operasi ini tidak menggunakan parameter permintaan.

Header Permintaan

Operasi ini hanya menggunakan header permintaan yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header permintaan umum, lihat Header Permintaan Umum.

Isi Permintaan

Operasi ini tidak memiliki isi permintaan.

Respons

Sintaksis

```
HTTP/1.1 204 No Content
x-amzn-RequestId: x-amzn-RequestId
Date: Date
```

Header Respons

Operasi ini hanya menggunakan header respons yang umum untuk sebagian besar respons. Untuk informasi tentang header permintaan umum, lihat Header Respons Umum.

Isi Respons

Operasi ini tidak mengembalikan isi respons.

Kesalahan

Untuk informasi tentang pengecualian Amazon Glacier dan pesan kesalahan, lihat. Respons Kesalahan

Contoh

Contoh berikut menunjukkan cara menghapus konfigurasi notifikasi untuk vault.

Contoh Permintaan

Dalam contoh ini, permintaan `DELETE` dikirim ke subsumber daya `notification-configuration` dari vault yang disebut `examplevault`.

```
DELETE /111122223333/vaults/examplevault/notification-configuration HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
x-amz-glacier-version: 2012-06-01
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
```

Contoh Respons

```
HTTP/1.1 204 No Content
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:00:00 GMT
```

Bagian Terkait

- Mendapatkan Notifikasi Vault (GET notification-configuration)
- Menetapkan Konfigurasi Notifikasi Vault (PUT notification-configuration)
- Identity and Access Management untuk Amazon Glacier

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu Amazon khusus bahasa SDKs, lihat berikut ini:

- AWS Command Line Interface

Menjelaskan Vault (GET vault)

Deskripsi

Operasi ini mengembalikan informasi tentang vault, termasuk Amazon Resource Name (ARN) vault, tanggal vault dibuat, jumlah arsip yang terdapat di dalam vault, dan ukuran total semua arsip di vault. Jumlah arsip dan ukuran totalnya adalah pada inventaris brankas terakhir Amazon Glacier (Amazon Glacier) yang dihasilkan (lihat). Bekerja dengan Vaults di Amazon Glacier Amazon Glacier menghasilkan inventaris lemari besi kira-kira setiap hari. Ini berarti jika Anda menambahkan atau menghapus arsip dari vault, lalu segera mengirim permintaan Jelaskan Vault, respons mungkin tidak mencerminkan perubahan.

Permintaan

Untuk mendapatkan informasi tentang vault, kirim permintaan `GET` ke URI dari sumber daya vault tertentu.

Sintaksis

```
GET /AccountId/vaults/VaultName HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: Date
Authorization: SignatureValue
x-amz-glacier-version: 2012-06-01
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID akun yang memiliki brankas. Anda dapat menentukan Akun AWS ID atau secara opsional satu '`-`' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menggunakan ID akun, jangan sertakan tanda hubung ('-') di ID.

Parameter Permintaan

Operasi ini tidak menggunakan parameter permintaan.

Header Permintaan

Operasi ini hanya menggunakan header permintaan yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header permintaan umum, lihat Header Permintaan Umum.

Isi Permintaan

Operasi ini tidak memiliki isi permintaan.

Respons

Sintaksis

```
HTTP/1.1 200 OK
x-amzn-RequestId: x-amzn-RequestId
Date: Date
Content-Type: application/json
Content-Length: Length

{
  "CreationDate" : String,
  "LastInventoryDate" : String,
  "NumberOfArchives" : Number,
  "SizeInBytes" : Number,
  "VaultARN" : String,
  "VaultName" : String
}
```

Header Respons

Operasi ini hanya menggunakan header respons yang umum untuk sebagian besar respons. Untuk informasi tentang header permintaan umum, lihat Header Respons Umum.

Isi Respons

Isi respons berisi bidang JSON berikut.

CreationDate

Tanggal UTC saat vault dibuat.

Tipe: Representasi string dalam format tanggal ISO 8601, misalnya `2013-03-20T17:03:43.221Z`.

LastInventoryDate

Tanggal UTC ketika Amazon Glacier menyelesaikan inventaris brankas terakhir. Untuk informasi tentang memulai inventaris untuk vault, lihat Memulai Tugas (POST jobs).

Tipe: Representasi string dalam format tanggal ISO 8601, misalnya `2013-03-20T17:03:43.221Z`.

NumberOfArchives

Jumlah arsip di vault sesuai inventaris vault terakhir. Bidang ini akan mengembalikan nol jika inventaris belum berjalan di vault, misalnya, jika Anda baru saja membuat vault.

Tipe: Angka

SizeInBytes

Ukuran total dalam byte dari arsip di vault termasuk setiap overhead per arsip, yang dimulai dari tanggal inventaris terakhir. Bidang ini akan mengembalikan `null` jika inventaris belum berjalan di vault, misalnya, jika Anda baru saja membuat vault.

Tipe: Angka

Vaultarn

Amazon Resource Name (ARN) dari vault.

Tipe: String

VaultName

Nama vault yang ditentukan saat pembuatan. Nama vault juga disertakan dalam ARN vault tersebut.

Tipe: String

Kesalahan

Untuk informasi tentang pengecualian Amazon Glacier dan pesan kesalahan, lihat. Respons Kesalahan

Contoh

Contoh Permintaan

Contoh berikut menunjukkan cara mendapatkan informasi tentang vault bernama `examplevault`.

```
GET /-/vaults/examplevault HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
x-amz-glacier-version: 2012-06-01
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
```

Contoh Respons

```
HTTP/1.1 200 OK
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:02:00 GMT
Content-Type: application/json
Content-Length: 260

{
  "CreationDate" : "2012-02-20T17:01:45.198Z",
  "LastInventoryDate" : "2012-03-20T17:03:43.221Z",
  "NumberOfArchives" : 192,
  "SizeInBytes" : 78088912,
  "VaultARN" : "arn:aws:glacier:us-west-2:012345678901:vaults/examplevault",
  "VaultName" : "examplevault"
}
```

Bagian Terkait

- Membuat Vault (PUT vault)
- Mencantumkan Vault (GET vaults)
- Menghapus Vault (DELETE vault)
- Memulai Tugas (POST jobs)
- Identity and Access Management untuk Amazon Glacier

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu Amazon khusus bahasa SDKs, lihat berikut ini:

- AWS Command Line Interface

Mendapatkan Kebijakan Akses Vault (GET access-policy)

Deskripsi

Operasi ini mengambil subsumber daya `access-policy` yang ditetapkan di vault—untuk informasi selengkapnya tentang pengaturan subsumber daya ini, lihat Mendapatkan Kebijakan Akses Vault (PUT access-policy). Jika tidak ada kebijakan akses yang ditetapkan di vault, operasi mengembalikan kesalahan `404 Not found`. Untuk informasi selengkapnya tentang kebijakan akses vault, lihat Kebijakan Akses Vault.

Permintaan

Untuk mengembalikan kebijakan akses vault saat ini, kirim permintaan `GET` HTTP ke URI subsumber daya `access-policy` vault.

Sintaksis

```
GET /AccountId/vaults/vaultName/access-policy HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: Date
Authorization: SignatureValue
x-amz-glacier-version: 2012-06-01
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID akun yang memiliki brankas. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menggunakan ID akun, jangan sertakan tanda hubung ('-') di ID.

Parameter Permintaan

Operasi ini tidak menggunakan parameter permintaan.

Header Permintaan

Operasi ini hanya menggunakan header permintaan yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header permintaan umum, lihat Header Permintaan Umum.

Isi Permintaan

Operasi ini tidak memiliki isi permintaan.

Respons

Sebagai tanggapan, Amazon Glacier (Amazon Glacier) mengembalikan kebijakan akses vault dalam format JSON di badan respons.

Sintaksis

```
HTTP/1.1 200 OK
x-amzn-RequestId: x-amzn-RequestId
Date: Date
Content-Type: application/json
Content-Length: length

{
  "Policy": "string"
}
```

Header Respons

Operasi ini hanya menggunakan header respons yang umum untuk sebagian besar respons. Untuk informasi tentang header permintaan umum, lihat Header Respons Umum.

Isi Respons

Isi respons berisi bidang JSON berikut.

Kebijakan

Kebijakan akses vault sebagai string JSON, yang menggunakan "\" sebagai karakter escape.

Tipe: String

Kesalahan

Untuk informasi tentang pengecualian Amazon Glacier dan pesan kesalahan, lihat. Respons Kesalahan

Contoh

Contoh berikut menunjukkan cara mendapatkan kebijakan akses vault.

Contoh Permintaan

Dalam contoh ini, permintaan GET dikirim ke URI dari subsumber daya `access-policy` vault.

```
GET /-/vaults/examplevault/access-policy HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
x-amz-glacier-version: 2012-06-01
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
```

Contoh Respons

Jika permintaan berhasil, Amazon Glacier mengembalikan kebijakan akses vault sebagai string JSON di badan respons. String JSON yang dikembalikan menggunakan "\" sebagai karakter escape, seperti yang ditunjukkan dalam contoh Mendapatkan Kebijakan Akses Vault (PUT access-policy). Namun, contoh berikut menunjukkan string JSON yang dikembalikan tanpa karakter escape agar mudah dibaca.

```
HTTP/1.1 200 OK
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:00:00 GMT
Content-Type: application/json
Content-Length: length

{
  "Policy": "
    {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Sid": "allow-time-based-deletes",
          "Principal": {
            "AWS": "999999999999"
          },
          "Effect": "Allow",
          "Action": "glacier:Delete*",
          "Resource": [
```

```
            "arn:aws:glacier:us-west-2:999999999999:vaults/examplevault"
          ],
          "Condition": {
            "DateGreaterThan": {
              "aws:CurrentTime": "2018-12-31T00:00:00Z"
            }
          }
        }
      ]
    }
  "
}
```

Bagian Terkait

- Menghapus Kebijakan Akses Vault (DELETE access-policy)
- Mendapatkan Kebijakan Akses Vault (PUT access-policy)

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu Amazon khusus bahasa SDKs, lihat berikut ini:

- AWS Command Line Interface

Mendapatkan Kunci Vault (GET lock-policy)

Deskripsi

Operasi ini mengambil atribut berikut dari subsumber daya `lock-policy` yang ditetapkan pada vault yang ditentukan:

- Kebijakan kunci vault yang ditetapkan di vault.
- Status kunci vault, yakni `InProgess` atau `Locked`.
- Saat ID kunci kedaluwarsa. ID kunci digunakan untuk menyelesaikan proses penguncian vault.
- Ketika kunci vault dimulai dan ditetapkan ke status `InProgress`.

Kunci vault ditetapkan ke dalam status `InProgress` dengan memanggil Memulai Kunci Vault (POST lock-policy). Kunci vault ditetapkan ke dalam status `Locked` dengan memanggil Menyelesaikan Kunci Vault (POST lockId). Anda dapat menghentikan proses penguncian vault dengan memanggil Membatalkan Kunci Vault (DELETE lock-policy). Untuk informasi selengkapnya tentang proses penguncian vault, lihat Kunci Gletser Amazon Glacier.

Jika tidak ada kebijakan kunci vault yang ditetapkan di vault, operasi mengembalikan kesalahan `404 Not found`. Untuk informasi selengkapnya tentang kebijakan kunci vault, lihat Kebijakan Vault Lock.

Permintaan

Untuk mengembalikan kebijakan kunci vault saat ini dan atribut lainnya, kirim permintaan `GET` HTTP untuk URI dari subsumber daya `lock-policy` vault seperti yang ditunjukkan dalam contoh sintaks berikut.

Sintaksis

```
GET /AccountId/vaults/vaultName/lock-policy HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: Date
Authorization: SignatureValue
x-amz-glacier-version: 2012-06-01
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID akun yang memiliki brankas. Anda dapat menentukan Akun AWS ID atau secara opsional satu '`-`' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menggunakan ID akun, jangan sertakan tanda hubung ('-') di ID.

Parameter Permintaan

Operasi ini tidak menggunakan parameter permintaan.

Header Permintaan

Operasi ini hanya menggunakan header permintaan yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header permintaan umum, lihat Header Permintaan Umum.

Isi Permintaan

Operasi ini tidak memiliki isi permintaan.

Respons

Sebagai tanggapan, Amazon Glacier (Amazon Glacier) mengembalikan kebijakan akses vault dalam format JSON di badan respons.

Sintaksis

```
HTTP/1.1 200 OK
x-amzn-RequestId: x-amzn-RequestId
Date: Date
Content-Type: application/json
Content-Length: length

{
  "Policy": "string",
  "State": "string",
  "ExpirationDate": "string",
  "CreationDate":"string"
}
```

Header Respons

Operasi ini hanya menggunakan header respons yang umum untuk sebagian besar respons. Untuk informasi tentang header permintaan umum, lihat Header Respons Umum.

Isi Respons

Isi respons berisi bidang JSON berikut.

Kebijakan

Kebijakan kunci vault sebagai string JSON, yang menggunakan "\" sebagai karakter escape.

Tipe: String

Status

Status kunci vault.

Tipe: String

Nilai yang valid: `InProgress|Locked`

ExpirationDate

Tanggal dan waktu UTC saat ID kunci kedaluwarsa. Nilai ini bisa `null` jika status kunci vault adalah `Locked`.

Tipe: Representasi string dalam format tanggal ISO 8601, misalnya `2013-03-20T17:03:43.221Z`.

CreationDate

Tanggal dan waktu UTC saat status kunci vault ditetapkan ke `InProgress`.

Tipe: Representasi string dalam format tanggal ISO 8601, misalnya `2013-03-20T17:03:43.221Z`.

Kesalahan

Untuk informasi tentang pengecualian Amazon Glacier dan pesan kesalahan, lihat. Respons Kesalahan

Contoh

Contoh berikut menunjukkan cara mendapatkan kebijakan kunci vault.

Contoh Permintaan

Dalam contoh ini, permintaan `GET` dikirim ke URI dari subsumber daya `lock-policy` vault.

```
GET /-/vaults/examplevault/lock-policy HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
x-amz-glacier-version: 2012-06-01
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
```

Contoh Respons

Jika permintaan berhasil, Amazon Glacier mengembalikan kebijakan akses vault sebagai string JSON di badan respons. String JSON yang dikembalikan menggunakan "\" sebagai karakter escape, seperti yang ditunjukkan dalam contoh permintaan Memulai Kunci Vault (POST lock-policy). Namun,

contoh berikut menunjukkan string JSON yang dikembalikan tanpa karakter escape agar mudah dibaca.

```
HTTP/1.1 200 OK
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:00:00 GMT
Content-Type: application/json
Content-Length: length

{
  "Policy": "
    {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Sid": "Define-vault-lock",
          "Principal": {
            "AWS": "arn:aws:iam::999999999999:root"
          },
          "Effect": "Deny",
          "Action": "glacier:DeleteArchive",
          "Resource": [
            "arn:aws:glacier:us-west-2:999999999999:vaults/examplevault"
          ],
          "Condition": {
            "NumericLessThanEquals": {
              "glacier:ArchiveAgeInDays": "365"
            }
          }
        }
      ]
    }
  ",
  "State": "InProgress",
  "ExpirationDate": "exampledate",
  "CreationDate": "exampledate"
}
```

Bagian Terkait

- Membatalkan Kunci Vault (DELETE lock-policy)

- Menyelesaikan Kunci Vault (POST lockId)

- Memulai Kunci Vault (POST lock-policy)

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu Amazon khusus bahasa SDKs, lihat berikut ini:

- AWS Command Line Interface

Mendapatkan Notifikasi Vault (GET notification-configuration)

Deskripsi

Operasi ini mengambil subsumber daya `notification-configuration` yang ditetapkan di vault (lihat Menetapkan Konfigurasi Notifikasi Vault (PUT notification-configuration). Jika konfigurasi notifikasi untuk vault tidak ditetapkan, operasi mengembalikan kesalahan `404 Not Found`. Untuk informasi selengkapnya tentang notifikasi vault, lihat Mengkonfigurasi Pemberitahuan Vault di Amazon Glacier.

Permintaan

Untuk mengambil informasi konfigurasi notifikasi, kirim permintaan `GET` ke URI dari subsumber daya `notification-configuration` vault.

Sintaksis

```
GET /AccountId/vaults/VaultName/notification-configuration HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: Date
Authorization: SignatureValue
x-amz-glacier-version: 2012-06-01
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID akun yang memiliki brankas. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini

> Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menggunakan ID akun, jangan sertakan tanda hubung ('-') di ID.

Parameter Permintaan

Operasi ini tidak menggunakan parameter permintaan.

Header Permintaan

Operasi ini hanya menggunakan header permintaan yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header permintaan umum, lihat Header Permintaan Umum.

Isi Permintaan

Operasi ini tidak memiliki isi permintaan.

Respons

Sintaksis

```
HTTP/1.1 200 OK
x-amzn-RequestId: x-amzn-RequestId
Date: Date
Content-Type: application/json
Content-Length: length
{
  "Events": [
    String,
    ...
  ],
  "SNSTopic": String
}
```

Header Respons

Operasi ini hanya menggunakan header respons yang umum untuk sebagian besar respons. Untuk informasi tentang header permintaan umum, lihat Header Respons Umum.

Isi Respons

Isi respons berisi bidang JSON berikut.

Peristiwa

Daftar satu atau lebih acara di mana Amazon Glacier (Amazon Glacier) akan mengirim pemberitahuan ke topik Amazon SNS yang ditentukan. Untuk informasi tentang peristiwa vault tempat Anda dapat mengonfigurasi vault untuk menerbitkan notifikasi, lihat Menetapkan Konfigurasi Notifikasi Vault (PUT notification-configuration).

Tipe: Array

SNSTopic

Amazon Resource Name (ARN topik Amazon Simple Notification Service (Amazon SNS). Untuk informasi selengkapnya, lihat Memulai Amazon SNS di Panduan Memulai Amazon Simple Notification Service.

Tipe: String

Kesalahan

Untuk informasi tentang pengecualian Amazon Glacier dan pesan kesalahan, lihat. Respons Kesalahan

Contoh

Contoh berikut menunjukkan cara mengambil konfigurasi notifikasi untuk vault.

Contoh Permintaan

Dalam contoh ini, permintaan `GET` dikirim ke subsumber daya `notification-configuration` dari vault.

```
GET /-/vaults/examplevault/notification-configuration HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
x-amz-glacier-version: 2012-06-01
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
```

Contoh Respons

Respons yang berhasil menunjukkan dokumen konfigurasi pencatatan audit di isi respons dalam format JSON. Dalam contoh ini, konfigurasi menunjukkan notifikasi untuk dua peristiwa

(`ArchiveRetrievalCompleted` dan `InventoryRetrievalCompleted`) dikirim ke topik Amazon SNS `arn:aws:sns:us-west-2:012345678901:mytopic`.

```
HTTP/1.1 200 OK
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:00:00 GMT
Content-Type: application/json
Content-Length: 150

{
  "Events": [
    "ArchiveRetrievalCompleted",
    "InventoryRetrievalCompleted"
  ],
  "SNSTopic": "arn:aws:sns:us-west-2:012345678901:mytopic"
}
```

Bagian Terkait

- Menghapus Notifikasi Vault (DELETE notification-configuration)
- Menetapkan Konfigurasi Notifikasi Vault (PUT notification-configuration)
- Identity and Access Management untuk Amazon Glacier

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu Amazon khusus bahasa SDKs, lihat berikut ini:

- AWS Command Line Interface

Memulai Kunci Vault (POST lock-policy)

Deskripsi

Operasi ini memulai proses penguncian vault dengan melakukan hal berikut:

- Menempatkan kebijakan kunci vault di vault yang ditentukan.
- Mengatur status kunci dari kunci vault ke `InProgress`.

- Mengembalikan ID kunci, yang digunakan untuk menyelesaikan proses penguncian vault.

Anda dapat menetapkan satu kebijakan kunci vault untuk setiap vault dan kebijakan ini dapat berukuran hingga 20 KB. Untuk informasi selengkapnya tentang kebijakan kunci vault, lihat Kebijakan Vault Lock.

Anda harus menyelesaikan proses penguncian vault dalam waktu 24 jam setelah kunci vault memasuki status `InProgress`. Setelah 24 jam berakhir, ID kunci akan kedaluwarsa, vault secara otomatis keluar dari status `InProgress`, dan kebijakan kunci vault dihapus dari vault. Anda memanggil Menyelesaikan Kunci Vault (POST lockId) untuk menyelesaikan proses penguncian vault dengan menetapkan status kunci vault ke `Locked`.

> **Note**
>
> Setelah kunci vault berada dalam status `Locked`, Anda tidak dapat memulai kunci vault baru untuk vault.

Anda dapat menghentikan proses penguncian vault dengan memanggil Membatalkan Kunci Vault (DELETE lock-policy). Anda bisa mendapatkan status kunci vault dengan memanggil Mendapatkan Kunci Vault (GET lock-policy). Untuk informasi selengkapnya tentang proses penguncian vault, lihat Kunci Gletser Amazon Glacier.

Jika operasi ini dipanggil saat kunci vault dalam status `InProgress`, operasi mengembalikan kesalahan `AccessDeniedException`. Saat kunci vault berada dalam status `InProgress`, Anda harus memanggil Membatalkan Kunci Vault (DELETE lock-policy) sebelum Anda dapat memulai kebijakan kunci vault baru.

Permintaan

Untuk memulai proses penguncian vault, kirim permintaan `POST` HTTP ke URI dari subsumber daya `lock-policy` vault, seperti yang ditunjukkan dalam contoh sintaks berikut.

Sintaksis

```
POST /AccountId/vaults/vaultName/lock-policy HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: Date
Authorization: SignatureValue
Content-Length: Length
```

```
x-amz-glacier-version: 2012-06-01

{
  "Policy": "string"
}
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID. Nilai ini harus cocok dengan Akun AWS ID yang terkait dengan kredensional yang digunakan untuk menandatangani permintaan. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menentukan ID akun, jangan sertakan tanda hubung ('-') di ID.

Parameter Permintaan

Operasi ini tidak menggunakan parameter permintaan.

Header Permintaan

Operasi ini hanya menggunakan header permintaan yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header permintaan umum, lihat Header Permintaan Umum.

Isi Permintaan

Isi permintaan berisi bidang JSON berikut.

Kebijakan

Kebijakan kunci vault sebagai string JSON, yang menggunakan "\" sebagai karakter escape.

Tipe: String

Diperlukan: Ya

Respons

Amazon Glacier (Amazon Glacier) mengembalikan `HTTP 201 Created` respons, jika kebijakan tersebut diterima.

Sintaksis

```
HTTP/1.1 201 Created
x-amzn-RequestId: x-amzn-RequestId
Date: Date
x-amz-lock-id: lockId
```

Header Respons

Respons yang sukses termasuk header respon berikut, selain header respons yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header respons umum, lihat Header Respons Umum.

Nama	Deskripsi
`x-amz-lock-id`	ID kunci, yang digunakan untuk menyelesaikan proses penguncian vault. Tipe: String

Isi Respons

Operasi ini tidak mengembalikan isi respons.

Kesalahan

Untuk informasi tentang pengecualian Amazon Glacier dan pesan kesalahan, lihat. Respons Kesalahan

Contoh

Contoh Permintaan

Contoh berikut mengirimkan permintaan PUT HTTP ke URI dari subsumber daya `lock-policy` vault. String JSON `Policy` menggunakan "\" sebagai karakter escape.

```
PUT /-/vaults/examplevault/lock-policy HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
```

```
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
Content-Length: length
x-amz-glacier-version: 2012-06-01

{"Policy":"{\"Version\":\"2012-10-17\",       \"Statement\":[{\"Sid\":\"Define-vault-
lock\",\"Effect\":\"Deny\",\"Principal\":{\"AWS\":\"arn:aws:iam::999999999999:root
\"},\"Action\":\"glacier:DeleteArchive\",\"Resource\":\"arn:aws:glacier:us-
west-2:999999999999:vaults/examplevault\",\"Condition\":{\"NumericLessThanEquals\":
{\"glacier:ArchiveAgeinDays\":\"365\"}}}]}"}
```

Contoh Respons

Jika permintaan berhasil, Amazon Glacier mengembalikan respons, seperti `HTTP 201 Created` yang ditunjukkan pada contoh berikut.

```
HTTP/1.1 201 Created
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:02:00 GMT
x-amz-lock-id: AE863rKkWZU53SLW5be4DUcW
```

Bagian Terkait

- Membatalkan Kunci Vault (DELETE lock-policy)
- Menyelesaikan Kunci Vault (POST lockId)
- Mendapatkan Kunci Vault (GET lock-policy)

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu Amazon khusus bahasa SDKs, lihat berikut ini:

- AWS Command Line Interface

Mencantumkan Tanda untuk Vault (GET tags)

Operasi ini mencantumkan semua tag yang dilampirkan di vault. Operasi ini mengembalikan peta kosong jika tidak ada tanda. Untuk informasi selengkapnya tentang tanda, lihat Menandai Sumber Daya Amazon Glacier.

Sintaks Permintaan

Untuk mencantumkan tanda untuk vault, kirim permintaan HTTP GET ke URI tanda seperti yang ditunjukkan dalam contoh sintaks berikut.

```
GET /AccountId/vaults/vaultName/tags HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: Date
Authorization: SignatureValue
x-amz-glacier-version: 2012-06-01
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID. Nilai ini harus cocok dengan Akun AWS ID yang terkait dengan kredensional yang digunakan untuk menandatangani permintaan. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menentukan ID akun, jangan sertakan tanda hubung ('-') di ID.

Parameter Permintaan

Operasi ini tidak menggunakan parameter permintaan.

Header Permintaan

Operasi ini hanya menggunakan header permintaan yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header permintaan umum, lihat Header Permintaan Umum.

Isi Permintaan

Operasi ini tidak memiliki isi permintaan.

Respons

Jika operasi berhasil, layanan mengirimkan kembali respon `200 OK` HTTP.

Sintaks Respons

```
HTTP/1.1 200 OK
x-amzn-RequestId: x-amzn-RequestId
Date: Date
Content-Type: application/json
Content-Length: Length
{
   "Tags":
      {
         "string" : "string",
         "string" : "string"
      }
}
```

Header Respons

Operasi ini hanya menggunakan header respons yang umum untuk sebagian besar respons. Untuk informasi tentang header permintaan umum, lihat Header Respons Umum.

Isi Respons

Isi respons berisi bidang JSON berikut.

Tag

Tanda yang dilampirkan ke vault. Setiap tanda terdiri dari kunci dan nilai.

Tipe: Peta String ke String

Wajib: Ya

Kesalahan

Untuk informasi tentang pengecualian Amazon Glacier dan pesan kesalahan, lihat. Respons Kesalahan

Contoh

Contoh: Mencantumkan Tnada untuk Vault

Contoh berikut mencantumkan tanda untuk vault.

Contoh Permintaan

Dalam contoh ini, permintaan GET dikirim untuk mengambil daftar tanda dari vault yang ditentukan.

```
GET /-/vaults/examplevault/tags HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
x-amz-glacier-version: 2012-06-01
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
```

Contoh Respons

Jika permintaan berhasil, Amazon Glacier (Amazon Glacier) mengembalikan dengan daftar tag untuk `HTTP 200 OK` vault seperti yang ditunjukkan pada contoh berikut.

```
HTTP/1.1 200 OK
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:02:00 GMT
Content-Type: application/json
Content-Length: length

{
   "Tags",
      {
         "examplekey1": "examplevalue1",
         "examplekey2": "examplevalue2"
      }
}
```

Bagian Terkait

- Menambahkan Tanda ke Vault (POST tag add)

- Hapus Tanda Dari Vault (POST tags remove)

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu Amazon khusus bahasa SDKs, lihat berikut ini:

- AWS Command Line Interface

Mencantumkan Vault (GET vaults)

Deskripsi

Operasi ini mencantumkan semua vault yang dimiliki oleh akun pengguna yang memanggil. Daftar yang dikembalikan dalam respons diurutkan ASCII berdasarkan nama vault.

Secara default, operasi ini mengembalikan hingga 10 item per permintaan. Jika ada lebih banyak vault yang dicantumkan, bidang `marker` respons dalam isi respons berisi vault Amazon Resource Name (ARN) tempat untuk melanjutkan daftar dengan permintaan Cantumkan Vault baru; jika tidak, bidang `marker` adalah `null`. Dalam permintaan List Vaults berikutnya, Anda menetapkan `marker` parameter ke nilai Amazon Glacier (Amazon Glacier) yang dikembalikan dalam tanggapan terhadap permintaan List Vaults sebelumnya. Anda juga dapat membatasi jumlah vault yang dikembalikan dalam respons dengan menentukan parameter `limit` dalam permintaan.

Permintaan

Untuk mendapatkan daftar vault, Anda mengirim permintaan `GET` ke sumber daya vault.

Sintaksis

```
GET /AccountId/vaults HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: Date
Authorization: SignatureValue
x-amz-glacier-version: 2012-06-01
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID. Nilai ini harus cocok dengan Akun AWS ID yang terkait dengan kredensional yang digunakan untuk menandatangani permintaan. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menentukan ID akun, jangan sertakan tanda hubung ('-') di ID.

Parameter Permintaan

Operasi ini menggunakan parameter permintaan berikut.

Nama	Deskripsi	Wajib
`limit`	Jumlah maksimum vault yang dikembalikan. Batas default adalah 10. Jumlah vault yang dikembalikan mungkin kurang dari batas yang ditentukan, tetapi jumlah vault yang dikembali kan tidak pernah melebihi batas. Tipe: String Batasan: Nilai bilangan bulat minimum 1. Nilai bilangan bulat maksimum 10.	Tidak
`marker`	String yang digunakan untuk pemberian nomor halaman. `marker` menentukan ARN vault setelah cantuman vault harus dimulai. (Vault yang ditentukan oleh `marker` tidak disertakan dalam daftar yang dikembalikan.) Dapatkan nilai `marker` dari respons Cantumkan Vault sebelumnya. Anda harus menyertak an `marker` hanya jika Anda melanjutkan pemberian nomor halaman dari hasil yang dimulai dalam permintaan Cantumkan Vault sebelumnya. Menentukan nilai kosong ("") untuk penanda mengembalikan daftar vault yang dimulai dari vault pertama. Tipe: String	Tidak

Nama	Deskripsi	Wajib
	Batasan: Tidak Ada	

Header Permintaan

Operasi ini hanya menggunakan header permintaan yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header permintaan umum, lihat Header Permintaan Umum.

Isi Permintaan

Operasi ini tidak memiliki isi permintaan.

Respons

Sintaksis

```
HTTP/1.1 200 OK
x-amzn-RequestId: x-amzn-RequestId
Date: Date
Content-Type: application/json
Content-Length: Length

{
  "Marker": String
  "VaultList": [
   {
    "CreationDate": String,
    "LastInventoryDate": String,
    "NumberOfArchives": Number,
    "SizeInBytes": Number,
    "VaultARN": String,
    "VaultName": String
   },
   ...
  ]
}
```

Header Respons

Operasi ini hanya menggunakan header respons yang umum untuk sebagian besar respons. Untuk informasi tentang header permintaan umum, lihat Header Respons Umum.

Isi Respons

Isi respons berisi bidang JSON berikut.

CreationDate

Tanggal vault dibuat, dalam Waktu Universal Terkoordinasi (UTC).

Tipe: String. Representasi string dalam format tanggal ISO 8601, misalnya `2013-03-20T17:03:43.221Z`.

LastInventoryDate

Tanggal inventaris vault terakhir, dalam Waktu Universal Terkoordinasi (UTC). Bidang ini dapat nol jika inventaris belum berjalan di vault, misalnya, jika Anda baru saja membuat vault. Untuk informasi tentang memulai inventaris untuk vault, lihat Memulai Tugas (POST jobs).

Tipe: Representasi string dalam format tanggal ISO 8601, misalnya `2013-03-20T17:03:43.221Z`.

Penanda

`vaultARN` yang mewakili tempat untuk melanjutkan pemberian nomor halaman hasil. Anda menggunakan `marker` dalam permintaan Cantumkan Vault lainnya untuk mendapatkan lebih banyak vault dalam daftar. Jika tidak ada lagi vault, nilai ini adalah `null`.

Tipe: String

NumberOfArchives

Jumlah arsip di vault mulai dari tanggal inventaris terakhir.

Tipe: Angka

SizeInBytes

Ukuran total, dalam byte, dari semua arsip di vault termasuk overhead per arsip apa pun, yang dimulai dari tanggal inventaris terakhir.

Tipe: Angka

Vaultarn

Amazon Resource Name (ARN) dari vault.

Tipe: String

VaultList

Array objek apa pun, dengan masing-masing objek yang memberikan deskripsi vault.

Tipe: Array

VaultName

Nama vault.

Tipe: String

Kesalahan

Untuk informasi tentang pengecualian Amazon Glacier dan pesan kesalahan, lihat. Respons Kesalahan

Contoh

Contoh: Mencantumkan Semua Vault

Contoh berikut mencantumkan vault. Karena parameter `marker` dan `limit` tidak ditentukan dalam permintaan, hingga 10 vault dikembalikan.

Contoh Permintaan

```
GET /-/vaults HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
x-amz-glacier-version: 2012-06-01
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
```

Contoh Respons

`Marker` adalah `null` yang menunjukkan tidak ada lagi vault yang dicantumkan.

```
HTTP/1.1 200 OK
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:02:00 GMT
```

```
Content-Type: application/json
Content-Length: 497

{
  "Marker": null,
  "VaultList": [
   {
    "CreationDate": "2012-03-16T22:22:47.214Z",
    "LastInventoryDate": "2012-03-21T22:06:51.218Z",
    "NumberOfArchives": 2,
    "SizeInBytes": 12334,
    "VaultARN": "arn:aws:glacier:us-west-2:012345678901:vaults/examplevault1",
    "VaultName": "examplevault1"
   },
   {
    "CreationDate": "2012-03-19T22:06:51.218Z",
    "LastInventoryDate": "2012-03-21T22:06:51.218Z",
    "NumberOfArchives": 0,
    "SizeInBytes": 0,
    "VaultARN": "arn:aws:glacier:us-west-2:012345678901:vaults/examplevault2",
    "VaultName": "examplevault2"
   },
   {
    "CreationDate": "2012-03-19T22:06:51.218Z",
    "LastInventoryDate": "2012-03-25T12:14:31.121Z",
    "NumberOfArchives": 0,
    "SizeInBytes": 0,
    "VaultARN": "arn:aws:glacier:us-west-2:012345678901:vaults/examplevault3",
    "VaultName": "examplevault3"
   }
  ]
}
```

Contoh: Daftar Parsial Vault

Contoh berikut mengembalikan dua vault yang dimulai dari vault yang ditentukan oleh `marker`.

Contoh Permintaan

```
GET /-/vaults?limit=2&marker=arn:aws:glacier:us-west-2:012345678901:vaults/
examplevault1 HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
x-amz-glacier-version: 2012-06-01
```

```
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
```

Contoh Respons

Dua vault dikembalikan dalam daftar. `Marker` berisi ARN vault untuk melanjutkan pemberian nomor halaman dalam permintaan Cantumkan Vault lainnya.

```
HTTP/1.1 200 OK
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:02:00 GMT
Content-Type: application/json
Content-Length: 497

{
  "Marker": "arn:aws:glacier:us-west-2:012345678901:vaults/examplevault3",
  "VaultList": [
   {
    "CreationDate": "2012-03-16T22:22:47.214Z",
    "LastInventoryDate": "2012-03-21T22:06:51.218Z",
    "NumberOfArchives": 2,
    "SizeInBytes": 12334,
    "VaultARN": "arn:aws:glacier:us-west-2:012345678901:vaults/examplevault1",
    "VaultName": "examplevault1"
   },
   {
    "CreationDate": "2012-03-19T22:06:51.218Z",
    "LastInventoryDate": "2012-03-21T22:06:51.218Z",
    "NumberOfArchives": 0,
    "SizeInBytes": 0,
    "VaultARN": "arn:aws:glacier:us-west-2:012345678901:vaults/examplevault2",
    "VaultName": "examplevault2"
   }
  ]
}
```

Bagian Terkait

- Membuat Vault (PUT vault)
- Menghapus Vault (DELETE vault)

- Memulai Tugas (POST jobs)
- Identity and Access Management untuk Amazon Glacier

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu Amazon khusus bahasa SDKs, lihat berikut ini:

- AWS Command Line Interface

Hapus Tanda Dari Vault (POST tags remove)

Operasi ini menghapus satu tanda atau lebih dari serangkaian tanda yang dilampirkan ke vault. Untuk informasi selengkapnya tentang tanda, lihat Menandai Sumber Daya Amazon Glacier.

Operasi ini bersifat idempoten. Operasi akan berhasil, meskipun tidak ada tanda yang dilampirkan ke vault.

Sintaks Permintaan

Untuk menghapus tag dari vault, kirim permintaan HTTP POST ke tag URI seperti yang ditunjukkan dalam contoh sintaks berikut.

```
POST /AccountId/vaults/vaultName/tags?operation=remove HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: Date
Authorization: SignatureValue
Content-Length: Length
x-amz-glacier-version: 2012-06-01
{
   "TagKeys": [
      "string",
      "string"
   ]
}
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID. Nilai ini harus cocok dengan Akun AWS ID yang terkait dengan kredensional yang digunakan untuk menandatangani permintaan. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menentukan ID akun, jangan sertakan tanda hubung ('-') di ID.

Parameter Permintaan

Nama	Deskripsi	Wajib
`operation=remove`	Parameter string kueri tunggal `operation` dengan nilai `remove` untuk membedakannya dari Menambahkan Tanda ke Vault (POST tag add).	Ya

Header Permintaan

Operasi ini hanya menggunakan header permintaan yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header permintaan umum, lihat Header Permintaan Umum.

Isi Permintaan

Isi permintaan berisi bidang JSON berikut.

TagKeys

Daftar kunci tanda. Setiap tanda yang sesuai dihapus dari vault.

Tipe: Array string

Batasan panjang: Minimum 1 item dalam daftar. Maksimum 10 item dalam daftar.

Wajib: Ya

Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP `204 No Content` dengan isi HTTP kosong.

Sintaksis

```
HTTP/1.1 204 No Content
x-amzn-RequestId: x-amzn-RequestId
Date: Date
```

Header Respons

Operasi ini hanya menggunakan header respons yang umum untuk sebagian besar respons. Untuk informasi tentang header permintaan umum, lihat Header Respons Umum.

Isi Respons

Operasi ini tidak mengembalikan isi respons.

Kesalahan

Untuk informasi tentang pengecualian Amazon Glacier dan pesan kesalahan, lihat. Respons Kesalahan

Contoh

Contoh Permintaan

Contoh berikut mengirimkan permintaan HTTP POST untuk menghapus tanda yang ditentukan.

```
POST /-/vaults/examplevault/tags?operation=remove HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
Content-Length: length
x-amz-glacier-version: 2012-06-01

{
```

```
   "TagsKeys": [
      "examplekey1",
      "examplekey2"
   ]
}
```

Contoh Respons

Jika permintaan berhasil Amazon Glacier (Amazon Glacier) mengembalikan `HTTP 204 No Content` seperti yang ditunjukkan pada contoh berikut.

```
HTTP/1.1 204 No Content
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:02:00 GMT
```

Bagian Terkait

- Menambahkan Tanda ke Vault (POST tag add)
- Mencantumkan Tanda untuk Vault (GET tags)

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu Amazon khusus bahasa SDKs, lihat berikut ini:

- AWS Command Line Interface

Mendapatkan Kebijakan Akses Vault (PUT access-policy)

Deskripsi

Operasi ini mengonfigurasi kebijakan akses untuk vault dan akan menimpa kebijakan yang ada. Untuk mengonfigurasi kebijakan akses vault, kirim permintaan PUT ke subsumber daya `access-policy`dari vault. Anda dapat menetapkan satu kebijakan akses per vault dan kebijakan dapat berukuran hingga 20 KB. Untuk informasi selengkapnya tentang kebijakan akses vault, lihat Kebijakan Akses Vault.

Permintaan

Sintaksis

Untuk menetapkan kebijakan akses, kirim permintaan PUT HTTP ke URI dari subsumber daya `access-policy` vault seperti yang ditunjukkan dalam contoh sintaks berikut.

```
PUT /AccountId/vaults/vaultName/access-policy HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: Date
Authorization: SignatureValue
Content-Length: Length
x-amz-glacier-version: 2012-06-01

{
  "Policy": "string"
}
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID akun yang memiliki brankas. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menggunakan ID akun, jangan sertakan tanda hubung ('-') di ID.

Parameter Permintaan

Operasi ini tidak menggunakan parameter permintaan.

Header Permintaan

Operasi ini hanya menggunakan header permintaan yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header permintaan umum, lihat Header Permintaan Umum.

Isi Permintaan

Isi permintaan berisi bidang JSON berikut.

Kebijakan

Kebijakan akses vault sebagai string JSON, yang menggunakan "\" sebagai karakter escape.

Tipe: String

Diperlukan: Ya

Respons

Sebagai tanggapan, Amazon Glacier `204 No Content` kembali jika kebijakan tersebut diterima.

Sintaksis

```
HTTP/1.1 204 No Content
x-amzn-RequestId: x-amzn-RequestId
Date: Date
```

Header Respons

Operasi ini hanya menggunakan header respons yang umum untuk sebagian besar respons. Untuk informasi tentang header permintaan umum, lihat Header Respons Umum.

Isi Respons

Operasi ini tidak mengembalikan isi respons.

Kesalahan

Untuk informasi tentang pengecualian Amazon Glacier dan pesan kesalahan, lihat. Respons Kesalahan

Contoh

Contoh Permintaan

Contoh berikut mengirimkan permintaan PUT HTTP ke URI dari subsumber daya `access-policy` vault. String JSON `Policy` menggunakan "\" sebagai karakter escape.

```
PUT /-/vaults/examplevault/access-policy HTTP/1.1
```

```
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
Content-Length: length
x-amz-glacier-version: 2012-06-01

{"Policy":"{\"Version\":\"2012-10-17\",       \"Statement\":[{\"Sid\":
\"Define-owner-access-rights\",\"Effect\":\"Allow\",\"Principal\":{\"AWS\":
\"arn:aws:iam::999999999999:root\"},\"Action\":\"glacier:DeleteArchive\",\"Resource\":
\"arn:aws:glacier:us-west-2:999999999999:vaults/examplevault\"}]}"}
```

Contoh Respons

Jika permintaan berhasil, Amazon Glacier (Amazon Glacier) mengembalikan `HTTP 204 No Content` seperti yang ditunjukkan pada contoh berikut.

```
HTTP/1.1 204 No Content
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:02:00 GMT
```

Bagian Terkait

- Menghapus Kebijakan Akses Vault (DELETE access-policy)
- Mendapatkan Kebijakan Akses Vault (GET access-policy)

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu Amazon khusus bahasa SDKs, lihat berikut ini:

- AWS Command Line Interface

Menetapkan Konfigurasi Notifikasi Vault (PUT notification-configuration)

Deskripsi

Mengambil arsip dan inventaris brankas adalah operasi asinkron di Amazon Glacier (Amazon Glacier) di mana Anda harus terlebih dahulu memulai pekerjaan dan menunggu pekerjaan selesai sebelum Anda dapat mengunduh hasil pekerjaan. Anda dapat mengonfigurasi vault untuk mengirim pesan ke topik Amazon Simple Notification Service (Amazon SNS) ketika tugas ini selesai. Anda dapat menggunakan operasi ini untuk menetapkan konfigurasi notifikasi di vault. Untuk informasi selengkapnya, lihat Mengkonfigurasi Pemberitahuan Vault di Amazon Glacier.

Untuk mengonfigurasi notifikasi vault, kirim permintaan PUT ke subsumber daya `notification-configuration` dari vault. Konfigurasi notifikasi spesifik dengan vault; Oleh karena itu, juga disebut sebagai subsumber daya vault. Permintaan harus menyertakan dokumen JSON yang menyediakan topik Amazon Simple Notification Service (Amazon SNS) dan acara yang Anda inginkan Amazon Glacier mengirim notifikasi ke topik tersebut.

Anda dapat mengonfigurasi vault untuk menerbitkan notifikasi untuk peristiwa vault berikut:

- **`ArchiveRetrievalCompleted`**— Peristiwa ini terjadi ketika tugas yang dimulai untuk pengambilan arsip selesai (Memulai Tugas (POST jobs)). Status tugas yang selesai dapat `Succeeded` atau `Failed`. Notifikasi yang dikirim ke topik SNS adalah output yang sama seperti yang dikembalikan dari Mendeskripsikan Tugas (GET JobID).
- **`InventoryRetrievalCompleted`**— Peristiwa ini terjadi ketika tugas yang dimulai untuk pengambilan inventaris selesai (Memulai Tugas (POST jobs)). Status tugas yang selesai dapat `Succeeded` atau `Failed`. Notifikasi yang dikirim ke topik SNS adalah output yang sama seperti yang dikembalikan dari Mendeskripsikan Tugas (GET JobID).

Topik Amazon SNS harus memberikan izin ke vault agar diizinkan untuk menerbitkan notifikasi ke topik.

Permintaan

Untuk menetapkan konfigurasi notifikasi di vault, kirim permintaan PUT ke URI dari subsumber daya `notification-configuration` dari vault. Anda menentukan konfigurasi dalam isi permintaan. Konfigurasi termasuk nama topik Amazon SNS dan array peristiwa yang memicu notifikasi untuk setiap topik.

Sintaksis

```
PUT /AccountId/vaults/VaultName/notification-configuration HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: Date
Authorization: SignatureValue
x-amz-glacier-version: 2012-06-01

{
   "SNSTopic": String,
   "Events":[String, ...]
}
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID akun yang memiliki brankas. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menggunakan ID akun, jangan sertakan tanda hubung ('-') di ID.

Parameter Permintaan

Operasi ini tidak menggunakan parameter permintaan.

Header Permintaan

Operasi ini hanya menggunakan header permintaan yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header permintaan umum, lihat Header Permintaan Umum.

Isi Permintaan

JSON dalam isi permintaan berisi bidang berikut.

Peristiwa

Sebuah array dari satu atau beberapa acara yang Anda ingin Amazon Glacier untuk mengirim pemberitahuan.

Nilai yang Valid: `ArchiveRetrievalCompleted` | `InventoryRetrievalCompleted`

Wajib: ya

Tipe: Array

SNSTopic

ARN topik Amazon SNS. Untuk informasi selengkapnya, buka Memulai Amazon SNS di Panduan Memulai Amazon Simple Notification Service.

Wajib: ya

Tipe: String

Respons

Sebagai tanggapan, Amazon Glacier (Amazon Glacier) `204 No Content` kembali jika konfigurasi notifikasi diterima.

Sintaksis

```
HTTP/1.1 204 No Content
x-amzn-RequestId: x-amzn-RequestId
Date: Date
```

Header Respons

Operasi ini hanya menggunakan header permintaan yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header permintaan umum, lihat Header Permintaan Umum.

Isi Respons

Operasi ini tidak mengembalikan isi respons.

Kesalahan

Untuk informasi tentang pengecualian Amazon Glacier dan pesan kesalahan, lihat. Respons Kesalahan

Contoh

Contoh berikut menunjukkan cara mengonfigurasi notifikasi vault.

Contoh Permintaan

Permintaan berikut menetapkan konfigurasi notifikasi `examplevault` sehingga notifikasi untuk dua peristiwa (`ArchiveRetrievalCompleted` dan `InventoryRetrievalCompleted`) dikirim ke topik Amazon SNS `arn:aws:sns:us-west-2:012345678901:mytopic`.

```
PUT /-/vaults/examplevault/notification-policy HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
x-amz-glacier-version: 2012-06-01
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2

{
   "Events": ["ArchiveRetrievalCompleted", "InventoryRetrievalCompleted"],
   "SNSTopic": "arn:aws:sns:us-west-2:012345678901:mytopic"
}
```

Contoh Respons

Respons yang berhasil mengembalikan `204 No Content`.

```
HTTP/1.1 204 No Content
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:00:00 GMT
```

Bagian Terkait

- Mendapatkan Notifikasi Vault (GET notification-configuration)
- Menghapus Notifikasi Vault (DELETE notification-configuration)
- Identity and Access Management untuk Amazon Glacier

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu Amazon khusus bahasa SDKs, lihat berikut ini:

- AWS Command Line Interface

Operasi Arsip

Berikut ini adalah operasi arsip yang tersedia untuk digunakan di Amazon Glacier.

Topik

- Menghapus Arsip (DELETE archive)
- Mengunggah Arsip (POST archive)

Menghapus Arsip (DELETE archive)

Deskripsi

Operasi ini menghapus arsip dari vault. Anda dapat menghapus satu arsip sekaligus dari vault. Untuk menghapus arsip, Anda harus memberikan ID arsip dalam permintaan hapus. Anda bisa mendapatkan ID arsip dengan mengunduh inventaris vault untuk vault yang berisi arsip. Untuk informasi selengkapnya tentang mengunduh inventaris vault, lihat Mengunduh Inventaris Vault di Amazon Glacier.

Setelah menghapus arsip, Anda mungkin masih dapat membuat permintaan yang sukses untuk memulai tugas guna mengambil arsip yang dihapus, tetapi tugas pengambilan arsip akan gagal.

Pengambilan arsip yang sedang berlangsung untuk ID arsip ketika Anda menghapus arsip mungkin atau mungkin tidak berhasil menurut skenario berikut:

- Jika pekerjaan pengambilan arsip secara aktif menyiapkan data untuk diunduh saat Amazon Glacier (Amazon Glacier) menerima permintaan hapus arsip, operasi pengambilan arsip mungkin gagal.
- Jika pekerjaan pengambilan arsip telah berhasil menyiapkan arsip untuk diunduh saat Amazon Glacier menerima permintaan hapus arsip, Anda akan dapat mengunduh hasilnya.

Untuk informasi selengkapnya tentang pengambilan arsip, lihat Mengunduh Arsip di Amazon Glacier.

Operasi ini bersifat idempoten. Mencoba untuk menghapus arsip yang sudah dihapus tidak menyebabkan kesalahan.

Permintaan

Untuk menghapus arsip, Anda mengirim permintaan `DELETE` ke URI sumber daya arsip.

Sintaksis

```
DELETE /AccountId/vaults/VaultName/archives/ArchiveID HTTP/1.1
Host: glacier.Region.amazonaws.com
x-amz-Date: Date
Authorization: SignatureValue
x-amz-glacier-version: 2012-06-01
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID akun yang memiliki brankas. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menggunakan ID akun, jangan sertakan tanda hubung ('-') di ID.

Parameter Permintaan

Operasi ini tidak menggunakan parameter permintaan.

Header Permintaan

Operasi ini hanya menggunakan header permintaan yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header permintaan umum, lihat Header Permintaan Umum.

Isi Permintaan

Operasi ini tidak memiliki isi permintaan.

Respons

Sintaksis

```
HTTP/1.1 204 No Content
x-amzn-RequestId: x-amzn-RequestId
Date: Date
```

Header Respons

Operasi ini hanya menggunakan header respons yang umum untuk sebagian besar respons. Untuk informasi tentang header permintaan umum, lihat Header Respons Umum.

Isi Respons

Operasi ini tidak mengembalikan isi respons.

Kesalahan

Untuk informasi tentang pengecualian Amazon Glacier dan pesan kesalahan, lihat. Respons Kesalahan

Contoh

Contoh berikut menunjukkan cara menghapus arsip dari vault bernama `examplevault`.

Contoh Permintaan

ID dari arsip yang akan dihapus ditentukan sebagai subsumber daya dari `archives`.

```
DELETE /-/vaults/examplevault/archives/NkbByEejwEggmBz2fTHgJrg0XBoDfjP4q6iu87-
TjhqG6eGoOY9Z8i1_AUyUsuhPAdTqLHy8pTl5nfCFJmDl2yEZONi5L26Omw12vcs01MNGntHEQL8MBfGlqrEXAMPLEArchi
 HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
x-amz-glacier-version: 2012-06-01
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
```

Contoh Respons

Jika permintaan berhasil, Amazon Glacier merespons `204 No Content` dengan untuk menunjukkan bahwa arsip dihapus.

```
HTTP/1.1 204 No Content
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:00:00 GMT
```

Bagian Terkait

- Memulai Unggahan Multipart (POST multipart-uploads)
- Mengunggah Arsip (POST archive)
- Identity and Access Management untuk Amazon Glacier

Mengunggah Arsip (POST archive)

Deskripsi

Operasi ini menambahkan arsip ke vault. Untuk pengunggahan yang berhasil, data Anda akan bertahan lama. Sebagai tanggapan, Amazon Glacier (Amazon Glacier) mengembalikan ID arsip di header respons. `x-amz-archive-id` Anda harus menyimpan ID arsip yang dikembalikan sehingga Anda dapat mengakses arsip nanti.

Anda harus memberikan hash SHA256 pohon dari data yang Anda unggah. Untuk informasi tentang menghitung hash SHA256 pohon, lihatChecksum Komputasi.

> **Note**
>
> Hash SHA256 pohon hanya diperlukan untuk tindakan Unggah Arsip (arsip POST) saat menggunakan API. Hal ini tidak diperlukan saat menggunakan AWS CLI.

Saat mengunggah arsip, Anda dapat secara opsional menentukan deskripsi arsip hingga 1.024 karakter ASCII yang dapat dicetak. Amazon Glacier mengembalikan deskripsi arsip saat Anda mengambil arsip atau mendapatkan inventaris brankas. Amazon Glacier tidak menafsirkan deskripsi dengan cara apa pun. Deskripsi arsip tidak harus unik. Anda tidak dapat menggunakan deskripsi untuk mengambil atau mengurutkan daftar arsip.

Kecuali untuk deskripsi arsip opsional, Amazon Glacier tidak mendukung metadata tambahan apa pun untuk arsip. ID arsip adalah urutan buram karakter tempat Anda tidak dapat menyimpulkan makna apa pun tentang arsip. Jadi, Anda dapat mempertahankan metadata tentang arsip di sisi klien. Untuk informasi selengkapnya, lihat Bekerja dengan Arsip di Amazon Glacier.

Arsip bersifat tetap. Setelah mengunggah arsip, Anda tidak dapat mengedit arsip atau deskripsinya.

Permintaan

Untuk mengunggah arsip, Anda menggunakan metode `POST` HTTP dan mencakup permintaan ke subsumber daya `archives` dari vault tempat Anda ingin menyimpan arsip. Permintaan harus

menyertakan ukuran muatan arsip, checksum (hash SHA256 pohon), dan secara opsional dapat menyertakan deskripsi arsip.

Sintaksis

```
POST /AccountId/vaults/VaultName/archives
Host: glacier.Region.amazonaws.com
x-amz-glacier-version: 2012-06-01
Date: Date
Authorization: SignatureValue
x-amz-archive-description: Description
x-amz-sha256-tree-hash: SHA256 tree hash
x-amz-content-sha256: SHA256 linear hash
Content-Length: Length

<Request body.>
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID akun yang memiliki brankas. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menggunakan ID akun, jangan sertakan tanda hubung ('-') di ID.

Parameter Permintaan

Implementasi operasi ini tidak menggunakan parameter permintaan.

Header Permintaan

Operasi ini menggunakan header permintaan berikut, selain header permintaan yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header permintaan umum, lihat Header Permintaan Umum.

Nama	Deskripsi	Wajib
`Content-Length`		Ya

Nama	Deskripsi	Wajib
	Ukuran objek, dalam byte. Untuk informasi lebih lanjut, kunjungi http://www.w3. org/Protocols/rfc2616/rfc2616-sec14. html #sec14 .13. Jenis: Angka Default: Tidak ada Batasan: Tidak Ada	
`x-amz-archive-description`	Deskripsi opsional dari arsip yang Anda unggah. Ini bisa menjadi deskripsi bahasa biasa atau beberapa pengident ifikasi yang Anda pilih untuk ditetapkan. Deskripsi tidak harus unik di seluruh arsip. Saat Anda mengambil inventaris vault (lihat Memulai Tugas (POST jobs)), inventaris mencakup deskripsi ini untuk masing-masing arsip yang dikembalikan sebagai respons. Tipe: String Default: Tidak ada Batasan: Deskripsi harus kurang dari atau sama dengan 1.024 karakter. Karakter yang diizinkan adalah ASCII 7-bit tanpa kode kontrol, khususnya nilai ASCII 32—126 desimal atau 0x20—0x7E heksadesimal.	Tidak
`x-amz-content-sha256`	SHA256 Checksum (hash linier) dari payload. Ini bukan nilai yang sama seperti yang Anda tentukan di header `x-amz-sha256-tree-hash` . Tipe: String Default: Tidak ada Batasan: Tidak Ada	Ya

Nama	Deskripsi	Wajib
`x-amz-sha256-tree-hash`	Checksum yang dihitung pengguna, hash SHA256 pohon, dari payload. Untuk informasi tentang menghitung hash SHA256 pohon, lihatChecksum Komputasi. Jika Amazon Glacier menghitung checksum payload yang berbeda, itu akan menolak permintaan tersebut. Tipe: String Default: Tidak ada Batasan: Tidak Ada	Ya

Isi Permintaan

Isi permintaan berisi data untuk diunggah.

Respons

Sebagai tanggapan, Amazon Glacier menyimpan arsip dengan tahan lama dan mengembalikan jalur URI ke ID arsip.

Sintaksis

```
HTTP/1.1 201 Created
x-amzn-RequestId: x-amzn-RequestId
Date: Date
x-amz-sha256-tree-hash: ChecksumComputedByAmazonGlacier
Location: Location
x-amz-archive-id: ArchiveId
```

Header Respons

Respons yang sukses termasuk header respon berikut, selain header respons yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header respons umum, lihat Header Respons Umum.

Nama	Deskripsi
`Location`	Jalur URI relatif dari sumber daya arsip yang baru ditambahkan. Tipe: String
`x-amz-archive-id`	ID arsip. Nilai ini juga disertakan sebagai bagian dari header `Location`. Tipe: String
`x-amz-sha256-tree-hash`	Checksum arsip dihitung oleh Amazon Glacier. Tipe: String

Isi Respons

Operasi ini tidak mengembalikan isi respons.

Kesalahan

Untuk informasi tentang pengecualian Amazon Glacier dan pesan kesalahan, lihat. Respons Kesalahan

Contoh

Contoh Permintaan

Contoh berikut menunjukkan permintaan untuk mengunggah arsip.

```
POST /-/vaults/examplevault/archives HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
x-amz-sha256-tree-hash:
 beb0fe31a1c7ca8c6c04d574ea906e3f97b31fdca7571defb5b44dca89b5af60
x-amz-content-sha256: 7f2fe580edb35154041fa3d4b41dd6d3adaef0c85d2ff6309f1d4b520eeecda3
Content-Length: 2097152
x-amz-glacier-version: 2012-06-01
Authorization: Authorization=AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-content-sha256;x-amz-date;x-
```

```
amz-glacier-
version,Signature=16b9a9e220a37e32f2e7be196b4ebb87120ca7974038210199ac5982e792cace

<Request body (2097152 bytes).>
```

Contoh Respons

Respons yang berhasil di bawah ini memiliki `Location` header tempat Anda bisa mendapatkan ID yang ditetapkan Amazon Glacier ke arsip.

```
HTTP/1.1 201 Created
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:00:00 GMT
x-amz-sha256-tree-hash:
 beb0fe31a1c7ca8c6c04d574ea906e3f97b31fdca7571defb5b44dca89b5af60
Location: /111122223333/vaults/examplevault/archives/
NkbByEejwEggmBz2fTHgJrg0XBoDfjP4q6iu87-
TjhqG6eGoOY9Z8i1_AUyUsuhPAdTqLHy8pTl5nfCFJmDl2yEZONi5L26Omw12vcs01MNGntHEQL8MBfGlqrEXAMPLEArchi
x-amz-archive-id: NkbByEejwEggmBz2fTHgJrg0XBoDfjP4q6iu87-
TjhqG6eGoOY9Z8i1_AUyUsuhPAdTqLHy8pTl5nfCFJmDl2yEZONi5L26Omw12vcs01MNGntHEQL8MBfGlqrEXAMPLEArchi
```

Bagian Terkait

- Bekerja dengan Arsip di Amazon Glacier
- Mengunggah Arsip Besar dalam Beberapa Bagian (Unggahan Multipart)
- Menghapus Arsip (DELETE archive)
- Identity and Access Management untuk Amazon Glacier

Operasi Unggahan Multipart

Berikut ini adalah operasi pengunggahan multibagian yang tersedia untuk digunakan di Amazon Glacier.

Topik

- Membatalkan Unggahan Multipart (DELETE uploadID)
- Menyelesaikan Unggahan Multipart (POST uploadID)
- Memulai Unggahan Multipart (POST multipart-uploads)
- Mencantumkan Bagian (GET uploadID)

- Mencantumkan Unggahan Multipart (GET multipart-uploads)
- Mengunggah Bagian (PUT UploadID)

Membatalkan Unggahan Multipart (DELETE uploadID)

Deskripsi

Perintah untuk operasi unggahan multipart ini menghentikan unggahan multipart yang diidentifikasi oleh ID unggahan.

Setelah permintaan Hentikan Unggahan Multipart berhasil, Anda tidak dapat menggunakan ID unggah untuk mengunggah lebih banyak bagian atau melakukan operasi lainnya. Gagal menghentikan unggahan multipart yang selesai. Namun, menghentikan unggahan yang sudah berhenti akan berhasil, untuk waktu yang singkat.

Operasi ini bersifat idempoten.

Untuk informasi tentang unggahan multipart, lihat Mengunggah Arsip Besar dalam Beberapa Bagian (Unggahan Multipart).

Permintaan

Untuk menghentikan unggahan multipart, kirim permintaan `DELETE` HTTP ke URI subsumber daya `multipart-uploads` vault dan mengidentifikasi ID unggahan multipart tertentu sebagai bagian dari URI.

Sintaksis

```
DELETE /AccountId/vaults/VaultName/multipart-uploads/uploadID HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: Date
Authorization: SignatureValue
x-amz-glacier-version: 2012-06-01
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID akun yang memiliki brankas. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan

untuk menandatangani permintaan. Jika Anda menggunakan ID akun, jangan sertakan tanda hubung ('-') di ID.

Parameter Permintaan

Operasi ini tidak menggunakan parameter permintaan.

Header Permintaan

Operasi ini hanya menggunakan header permintaan yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header permintaan umum, lihat Header Permintaan Umum.

Isi Permintaan

Operasi ini tidak memiliki isi permintaan.

Respons

Sintaksis

```
HTTP/1.1 204 No Content
x-amzn-RequestId: x-amzn-RequestId
Date: Date
```

Header Respons

Operasi ini hanya menggunakan header respons yang umum untuk sebagian besar respons. Untuk informasi tentang header permintaan umum, lihat Header Respons Umum.

Isi Respons

Operasi ini tidak mengembalikan isi respons.

Kesalahan

Untuk informasi tentang pengecualian Amazon Glacier dan pesan kesalahan, lihat. Respons Kesalahan

Contoh

Contoh Permintaan

Dalam contoh berikut, permintaan `DELETE` dikirim ke URI dari sumber daya ID unggahan multipart.

```
DELETE /-/vaults/examplevault/multipart-uploads/
OW2fM5iVylEpFEMM9_HpKowRapC3vn5sSL39_396UW9zLFUWVrnRHaPjUJddQ5OxSHVXjYtrN47NBZ-
khxOjyEXAMPLE  HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
x-amz-glacier-version: 2012-06-01
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
```

Contoh Respons

```
HTTP/1.1 204 No Content
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:00:00 GMT
```

Bagian Terkait

- Memulai Unggahan Multipart (POST multipart-uploads)
- Mengunggah Bagian (PUT UploadID)
- Menyelesaikan Unggahan Multipart (POST uploadID)
- Mencantumkan Unggahan Multipart (GET multipart-uploads)
- Mencantumkan Bagian (GET uploadID)
- Mengunggah Arsip Besar dalam Beberapa Bagian (Unggahan Multipart)
- Identity and Access Management untuk Amazon Glacier

Menyelesaikan Unggahan Multipart (POST uploadID)

Deskripsi

Anda memanggil operasi pengunggahan multibagian ini untuk memberi tahu Amazon Glacier (Amazon Glacier) bahwa semua bagian arsip telah diunggah dan Amazon Glacier sekarang dapat merakit arsip dari bagian yang diunggah.

Untuk informasi tentang unggahan multipart, lihat Mengunggah Arsip Besar dalam Beberapa Bagian (Unggahan Multipart).

Setelah merakit dan menyimpan arsip ke lemari besi, Amazon Glacier mengembalikan ID arsip sumber daya arsip yang baru dibuat. Setelah mengunggah arsip, Anda harus menyimpan ID arsip yang dikembalikan untuk mengambil arsip di titik berikutnya.

Dalam permintaan, Anda harus menyertakan hash SHA256 pohon yang dihitung dari seluruh arsip yang telah Anda unggah. Untuk informasi tentang menghitung hash SHA256 pohon, lihatChecksum Komputasi. Di sisi server, Amazon Glacier juga membangun hash pohon SHA256 dari arsip yang dirakit. Jika nilainya cocok, Amazon Glacier menyimpan arsip ke brankas; jika tidak, ia mengembalikan kesalahan, dan operasi gagal. Operasi Mencantumkan Bagian (GET uploadID) mengembalikan daftar bagian yang diunggah untuk unggahan multipart tertentu. Ini termasuk informasi checksum untuk setiap bagian yang diunggah yang dapat digunakan untuk melakukan debug masalah checksum yang buruk.

Selain itu, Amazon Glacier juga memeriksa rentang konten yang hilang. Ketika mengunggah bagian, Anda menentukan nilai kisaran yang mengidentifikasi tempat setiap bagian cocok dalam penyusunan final arsip. Saat merakit arsip akhir Amazon Glacier memeriksa rentang konten yang hilang dan jika ada rentang konten yang hilang, Amazon Glacier mengembalikan kesalahan dan operasi Unggah Multipart Lengkap gagal.

Selesaikan Unggahan Multipart adalah operasi idempoten. Setelah selesaikan unggahan multipart pertama Anda berhasil, jika Anda memanggil operasi lagi dalam waktu singkat, operasi akan berhasil dan mengembalikan ID arsip yang sama. Hal ini berguna jika Anda mengalami masalah jaringan atau menerima kesalahan server 500, dalam hal ini, Anda dapat mengulangi permintaan Selesaikan Unggahan Multipart dan mendapatkan ID arsip yang sama tanpa membuat arsip duplikat. Namun, perhatikan bahwa setelah unggahan multipart selesai, Anda tidak dapat memanggil operasi Cantumkan Bagian dan unggahan multipart tidak akan muncul dalam respond Cantumkan Unggahan Multipart, bahkan jika idempoten selesai dimungkinkan.

Permintaan

Untuk menyelesaikan unggahan multibagian, Anda mengirim permintaan HTTP POST ke URI ID unggahan yang dibuat Amazon Glacier sebagai tanggapan atas permintaan Inisiate Multipart Upload Anda. Ini adalah URI yang sama dengan yang Anda gunakan saat mengunggah bagian. Selain header umum yang diperlukan, Anda harus menyertakan hasil hash SHA256 pohon dari seluruh arsip dan ukuran total arsip dalam byte.

Sintaksis

```
POST /AccountId/vaults/VaultName/multipart-uploads/uploadID
```

```
Host: glacier.Region.amazonaws.com
Date: date
Authorization: SignatureValue
x-amz-sha256-tree-hash: SHA256 tree hash of the archive
x-amz-archive-size: ArchiveSize in bytes
x-amz-glacier-version: 2012-06-01
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID akun yang memiliki brankas. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredensil yang digunakan untuk menandatangani permintaan. Jika Anda menggunakan ID akun, jangan sertakan tanda hubung ('-') di ID.

Parameter Permintaan

Operasi ini tidak menggunakan parameter permintaan.

Header Permintaan

Operasi ini menggunakan header permintaan berikut, selain header permintaan yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header permintaan umum, lihat Header Permintaan Umum.

Nama	Deskripsi	Wajib
`x-amz-archive-size`	Ukuran total, dalam byte, dari seluruh arsip. Nilai ini harus jumlah dari semua ukuran masing-masing bagian yang Anda unggah. Tipe: String Default: Tidak ada Batasan: Tidak Ada	Ya

Nama	Deskripsi	Wajib
`x-amz-sha256-tree-hash`	Hash SHA256 pohon dari seluruh arsip. Ini adalah hash pohon hash SHA256 pohon dari masing-masing bagian. Jika nilai yang Anda tentukan dalam permintaan tidak cocok dengan hash SHA256 pohon dari arsip rakitan akhir seperti yang dihitung oleh Amazon Glacier, Amazon Glacier mengembalikan kesalahan dan permintaan gagal. Tipe: String Default: Tidak ada Batasan: Tidak Ada	Ya

Elemen permintaan

Operasi ini tidak menggunakan elemen permintaan.

Respons

Amazon Glacier (Amazon Glacier) membuat SHA256 hash pohon dari seluruh arsip. Jika nilainya cocok dengan hash SHA256 pohon dari seluruh arsip yang Anda tentukan dalam permintaan, Amazon Glacier menambahkan arsip ke vault. Sebagai respons, ini mengembalikan header `Location` HTTP dengan jalur URL dari sumber daya arsip yang baru ditambahkan. Jika ukuran arsip atau SHA256 yang Anda kirim dalam permintaan tidak cocok, Amazon Glacier akan mengembalikan kesalahan dan unggahan tetap dalam keadaan tidak lengkap. Dimungkinkan untuk mencoba lagi operasi Selesaikan Unggahan Multipart nanti dengan nilai yang benar, pada saat mana Anda dapat berhasil membuat arsip. Jika unggahan multipart tidak selesai, maka pada akhirnya Amazon Glacier akan merebut kembali ID unggahan tersebut.

Sintaksis

```
HTTP/1.1 201 Created
x-amzn-RequestId: x-amzn-RequestId
Date: Date
Location: Location
x-amz-archive-id: ArchiveId
```

Header Respons

Respons yang sukses termasuk header respon berikut, selain header respons yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header respons umum, lihat Header Respons Umum.

Nama	Deskripsi
`Location`	Jalur URI relatif dari arsip yang baru dibuat. URL ini menyertakan ID arsip yang dihasilkan oleh Amazon Glacier. Tipe: String
`x-amz-archive-id`	ID arsip. Nilai ini juga disertakan sebagai bagian dari header `Location`. Tipe: String

Bidang Respons

Operasi ini tidak mengembalikan isi respons.

Contoh

Contoh Permintaan

Dalam contoh ini, permintaan HTTP POST dikirim ke URI yang dikembalikan oleh permintaan Mulai Unggahan Multipart. Permintaan menentukan hash SHA256 pohon dari seluruh arsip dan ukuran arsip total.

```
POST /-/vaults/examplevault/multipart-uploads/
OW2fM5iVylEpFEMM9_HpKowRapC3vn5sSL39_396UW9zLFUWVrnRHaPjUJddQ5OxSHVXjYtrN47NBZ-
khxOjyEXAMPLE HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
z-amz-Date: 20170210T120000Z
x-amz-sha256-tree-hash:1ffc0f54dd5fdd66b62da70d25edacd0
x-amz-archive-size:8388608
x-amz-glacier-version: 2012-06-01
```

```
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
```

Contoh Respons

Contoh respons berikut menunjukkan bahwa Amazon Glacier berhasil membuat arsip dari bagian yang Anda unggah. Respons mencakup ID arsip dengan jalur lengkap.

```
HTTP/1.1 201 Created
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:00:00 GMT
Location: /111122223333/vaults/examplevault/archives/
NkbByEejwEggmBz2fTHgJrg0XBoDfjP4q6iu87-
TjhqG6eGoOY9Z8i1_AUyUsuhPAdTqLHy8pTl5nfCFJmDl2yEZONi5L26Omw12vcs01MNGntHEQL8MBfGlqrEXAMPLEArchi
x-amz-archive-id: NkbByEejwEggmBz2fTHgJrg0XBoDfjP4q6iu87-
TjhqG6eGoOY9Z8i1_AUyUsuhPAdTqLHy8pTl5nfCFJmDl2yEZONi5L26Omw12vcs01MNGntHEQL8MBfGlqrEXAMPLEArchi
```

Anda sekarang dapat mengirim permintaan HTTP ke URI dari sumber/arsip yang baru ditambahkan. Misalnya, Anda dapat mengirim permintaan GET untuk mengambil arsip.

Bagian Terkait

- Memulai Unggahan Multipart (POST multipart-uploads)
- Mengunggah Bagian (PUT UploadID)
- Membatalkan Unggahan Multipart (DELETE uploadID)
- Mencantumkan Unggahan Multipart (GET multipart-uploads)
- Mencantumkan Bagian (GET uploadID)
- Mengunggah Arsip Besar dalam Beberapa Bagian (Unggahan Multipart)
- Menghapus Arsip (DELETE archive)
- Identity and Access Management untuk Amazon Glacier

Memulai Unggahan Multipart (POST multipart-uploads)

Deskripsi

Operasi ini memulai unggahan multipart (lihat Mengunggah Arsip Besar dalam Beberapa Bagian (Unggahan Multipart)). Amazon Glacier (Amazon Glacier) membuat sumber daya unggahan

multibagian dan mengembalikan ID-nya sebagai respons. Anda menggunakan ID unggahan dalam operasi unggahan multipart selanjutnya.

Ketika memulai unggahan multipart, Anda menentukan ukuran bagian dalam jumlah byte. Ukuran bagian harus mebibyte (MiB) (1024 kibibytes [KiB]) dikalikan dengan kekuatan 2—misalnya, 1048576 (1 MiB), 2097152 (2 MiB), 4194304 (4 MiB), 8388608 (8 MiB), dan seterusnya. Ukuran bagian minimum yang diijinkan adalah 1 MiB, dan maksimum adalah 4 gibibytes (GiB).

Setiap bagian yang Anda unggah menggunakan ID unggah ini, kecuali yang terakhir, harus memiliki ukuran yang sama. Yang terakhir bisa berukuran sama atau lebih kecil. Misalnya, Anda ingin mengunggah file 16,2 MiB. Jika Anda memulai unggahan multipart dengan ukuran bagian 4 MiB, Anda akan mengunggah empat bagian masing-masing 4 MiB dan satu bagian 0,2 MiB.

> **Note**
>
> Anda tidak perlu mengetahui ukuran arsip saat memulai unggahan multipart karena Amazon Glacier tidak mengharuskan Anda menentukan ukuran arsip keseluruhan.

Setelah Anda menyelesaikan unggahan multibagian, Amazon Glacier menghapus sumber daya unggahan multibagian yang direferensikan oleh ID. Amazon Glacier juga akan menghapus sumber daya unggahan multibagian jika Anda membatalkan unggahan multibagian atau mungkin dihapus jika tidak ada aktivitas selama 24 jam. ID mungkin masih tersedia setelah 24 jam, tetapi aplikasi seharusnya tidak menduga perilaku ini.

Permintaan

Untuk memulai unggahan multipart, Anda mengirim permintaan POST HTTP ke URI subsumber daya `multipart-uploads` dari vault tempat Anda ingin menyimpan arsip. Permintaan harus mencakup ukuran bagian dan secara opsional dapat menyertakan deskripsi arsip.

Sintaksis

```
POST /AccountId/vaults/VaultName/multipart-uploads
Host: glacier.us-west-2.amazonaws.com
Date: Date
Authorization: SignatureValue
x-amz-glacier-version: 2012-06-01
x-amz-archive-description: ArchiveDescription
```

```
x-amz-part-size: PartSize
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID akun yang memiliki brankas. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menggunakan ID akun, jangan sertakan tanda hubung ('-') di ID.

Parameter Permintaan

Operasi ini tidak menggunakan parameter permintaan.

Header Permintaan

Operasi ini menggunakan header permintaan berikut, selain header permintaan yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header permintaan umum, lihat Header Permintaan Umum.

Nama	Deskripsi	Wajib
`x-amz-part-size`	Ukuran setiap bagian kecuali yang terakhir, dalam byte. Bagian terakhir bisa lebih kecil dari ukuran bagian ini. Tipe: String Default: Tidak Ada Kendala: Ukuran bagian harus mebibyte (1024 KiB) dikalikan dengan kekuatan 2 — misalnya, 1048576 (1 MiB), 2097152 (2 MiB), 4194304 (4 MiB), 8388608 (8 MiB), dan seterusnya. Ukuran bagian minimum yang diijinkan adalah 1 MiB, dan maksimum adalah 4 GiB (4096 MiB).	Ya

Nama	Deskripsi	Wajib
`x-amz-archive-description`	Arsipkan deskripsi yang Anda unggah di bagian. Ini bisa menjadi deskripsi bahasa biasa atau beberapa pengidentifikasi unik yang Anda pilih untuk ditetapkan. Saat Anda mengambil inventaris vault (lihat Memulai Tugas (POST jobs)), inventaris mencakup deskripsi ini untuk masing-masing arsip yang dikembalikan sebagai respons. Spasi yang mendahului dalam deskripsi arsip akan dihapus. Tipe: String Default: Tidak ada Batasan: Deskripsi harus kurang dari atau sama dengan 1024 byte. Karakter yang diizinkan adalah ASCII 7 bit tanda kode kontrol, khususnya nilai ASCII 32-126 desimal atau 0x20-0x7E heksadesimal.	Tidak

Isi Permintaan

Operasi ini tidak memiliki isi permintaan.

Respons

Sebagai tanggapan, Amazon Glacier membuat sumber daya unggahan multibagian yang diidentifikasi oleh ID dan mengembalikan jalur URI relatif dari ID unggahan multibagian.

Sintaksis

```
HTTP/1.1 201 Created
x-amzn-RequestId: x-amzn-RequestId
Date: Date
Location: Location
```

```
x-amz-multipart-upload-id: multiPartUploadId
```

Header Respons

Respons yang sukses termasuk header respon berikut, selain header respons yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header respons umum, lihat Header Respons Umum.

Nama	Deskripsi
`Location`	Jalur URI relatif dari ID unggahan multibagian Amazon Glacier dibuat. Anda menggunakan jalur URI ini untuk mencakup permintaan Anda untuk mengunggah bagian, dan untuk menyelesaikan unggahan multipart. Tipe: String
`x-amz-multipart-upload-id`	ID unggahan multipart. Nilai ini juga disertakan sebagai bagian dari header `Location`. Tipe: String

Isi Respons

Operasi ini tidak mengembalikan isi respons.

Kesalahan

Untuk informasi tentang pengecualian Amazon Glacier dan pesan kesalahan, lihat. Respons Kesalahan

Contoh

Contoh Permintaan

Contoh berikut memulai unggahan multipart dengan mengirimkan permintaan `POST` HTTP ke URI dari subsumber daya `multipart-uploads` dari vault bernama `examplevault`. Permintaan termasuk header untuk menentukan ukuran bagian dari 4 MiB (4194304 byte) dan deskripsi arsip opsional.

```
POST /-/vaults/examplevault/multipart-uploads
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
x-amz-archive-description: MyArchive-101
x-amz-part-size: 4194304
x-amz-glacier-version: 2012-06-01
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
```

Contoh Respons

Amazon Glacier membuat sumber daya unggahan multibagian dan menambahkannya ke subsumber daya brankas`multipart-uploads`. Header respons `Location` termasuk jalur URI relatif ke ID unggahan multipart.

```
HTTP/1.1 201 Created
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:00:00 GMT
Location: /111122223333/vaults/examplevault/multipart-uploads/
OW2fM5iVylEpFEMM9_HpKowRapC3vn5sSL39_396UW9zLFUWVrnRHaPjUJddQ5OxSHVXjYtrN47NBZ-
khxOjyEXAMPLE
x-amz-multipart-upload-id:
 OW2fM5iVylEpFEMM9_HpKowRapC3vn5sSL39_396UW9zLFUWVrnRHaPjUJddQ5OxSHVXjYtrN47NBZ-
khxOjyEXAMPLE
```

Untuk informasi tentang mengunggah bagian-bagian individu, lihat Mengunggah Bagian (PUT UploadID).

Bagian Terkait

- Mengunggah Bagian (PUT UploadID)
- Menyelesaikan Unggahan Multipart (POST uploadID)
- Membatalkan Unggahan Multipart (DELETE uploadID)
- Mencantumkan Unggahan Multipart (GET multipart-uploads)
- Mencantumkan Bagian (GET uploadID)
- Menghapus Arsip (DELETE archive)
- Mengunggah Arsip Besar dalam Beberapa Bagian (Unggahan Multipart)

- Identity and Access Management untuk Amazon Glacier

Mencantumkan Bagian (GET uploadID)

Deskripsi

Operasi unggahan multipart ini mencantumkan bagian arsip yang telah diunggah dalam unggahan multipart tertentu yang diidentifikasi oleh ID unggah. Untuk informasi tentang unggahan multipart, lihat Mengunggah Arsip Besar dalam Beberapa Bagian (Unggahan Multipart).

Anda dapat membuat permintaan ini kapan saja selama unggahan multipart yang sedang berlangsung sebelum menyelesaikan unggahan multipart. Amazon Glacier mengembalikan daftar bagian yang diurutkan berdasarkan rentang yang Anda tentukan di setiap unggahan bagian. Jika Anda mengirim permintaan Suku Cadang Daftar setelah menyelesaikan unggahan multibagian, Amazon Glacier (Amazon Glacier) mengembalikan kesalahan.

Operasi Cantumkan Bagian mendukung pemberian nomor halaman. Anda harus selalu memeriksa bidang `Marker` dalam isi respons untuk penanda tempat untuk melanjutkan daftar; jika tidak ada lagi item, bidang `marker` adalah `null`. Jika `marker` tidak null, untuk mengambil kumpulan bagian berikutnya, Anda mengirim permintaan Bagian Daftar lain dengan parameter `marker` permintaan yang disetel ke nilai penanda Amazon Glacier yang dikembalikan sebagai tanggapan atas permintaan Bagian Daftar Anda sebelumnya.

Anda juga dapat membatasi jumlah bagian yang dikembalikan dalam respons dengan menentukan parameter `limit` dalam permintaan.

Permintaan

Sintaksis

Untuk mencantumkan bagian dari unggahan multipart yang sedang berlangsung, Anda mengirim permintaan `GET` ke URI dari sumber daya ID unggahan multipart. ID unggahan multipart dikembalikan ketika Anda memulai unggahan multipart (Memulai Unggahan Multipart (POST multipart-uploads)). Anda dapat secara opsional menentukan parameter `marker` dan `limit`.

```
GET /AccountId/vaults/VaultName/multipart-uploads/uploadID HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: Date
Authorization: SignatureValue
```

```
x-amz-glacier-version: 2012-06-01
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID akun yang memiliki brankas. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menggunakan ID akun, jangan sertakan tanda hubung ('-') di ID.

Parameter Permintaan

Nama	Deskripsi	Wajib
`limit`	Jumlah maksimum bagian yang dikembalikan. Batas default adalah 50. Jumlah bagian yang dikembalikan mungkin kurang dari batas yang ditentukan, tetapi jumlah bagian yang dikembali kan tidak pernah melebihi batas. Tipe: String Batasan: Nilai bilangan bulat minimum 1. Nilai bilangan bulat maksimum `50`.	Tidak
`marker`	String buram yang digunakan untuk pemberian nomor halaman. `marker` menentukan bagian tempat cantuman bagian harus dimulai. Dapatkan nilai `marker` dari respons dari respons Cantumkan Bagian sebelumnya. Anda hanya perlu menyertak an `marker` jika Anda melanjutkan pemberian nomor halaman dari hasil yang dimulai dalam permintaan Cantumkan Bagian sebelumnya. Tipe: String Batasan: Tidak Ada	Tidak

Header Permintaan

Operasi ini hanya menggunakan header respons yang umum untuk sebagian besar respons. Untuk informasi tentang header permintaan umum, lihat Header Respons Umum.

Isi Permintaan

Operasi ini tidak memiliki isi permintaan.

Respons

Sintaksis

```
HTTP/1.1 200 OK
x-amzn-RequestId: x-amzn-RequestId
Date: Date
Content-Type: application/json
Content-Length: Length

{
    "ArchiveDescription" : String,
    "CreationDate" : String,
    "Marker": String,
    "MultipartUploadId" : String,
    "PartSizeInBytes" : Number,
    "Parts" :
    [ {
      "RangeInBytes" : String,
      "SHA256TreeHash" : String
      },
      ...
     ],
    "VaultARN" : String
}
```

Header Respons

Operasi ini hanya menggunakan header respons yang umum untuk sebagian besar respons. Untuk informasi tentang header permintaan umum, lihat Header Respons Umum.

Isi Respons

Isi respons berisi bidang JSON berikut.

ArchiveDescription

Deskripsi arsip yang ditentukan dalam permintaan Mulai Unggahan Multipart. Bidang ini adalah `null` jika tidak ada deskripsi arsip yang ditentukan dalam operasi Mulai Unggahan Multipart.

Tipe: String

CreationDate

Waktu UTC unggahan multipart dimulai.

Tipe: String. Representasi string dalam format tanggal ISO 8601, misalnya `2013-03-20T17:03:43.221Z`.

Penanda

String buram yang mewakili tempat untuk melanjutkan pemberian nomor halaman hasil. Anda menggunakan `marker` dalam permintaan Cantumkan Bagian baru untuk mendapatkan lebih banyak tugas dalam daftar. Jika tidak ada lagi bagian, nilai ini adalah `null`.

Tipe: String

MultipartUploadId

ID upload yang bagian-bagiannya terkait.

Tipe: String

PartSizeInBytes

Ukuran bagian dalam byte. Ini adalah nilai yang sama dengan yang Anda tentukan dalam permintaan Mulai Unggahan Multipart.

Tipe: Angka

Bagian

Daftar ukuran bagian dari unggahan multipart. Setiap objek dalam array berisi pasangan nama/nilai `RangeBytes` dan `sha256-tree-hash`.

Tipe: Array

RangeInBytes

Rentang byte bagian, termasuk nilai atas rentang.

Tipe: String

SHA256TreeHash

Nilai hash SHA256 pohon yang dihitung Amazon Glacier untuk bagian tersebut. Bidang ini tidak pernah ada `null`.

Tipe: String

Vaultarn

Amazon Resource Name (ARN) dari vault tempat unggahan multipart dimulai.

Tipe: String

Kesalahan

Untuk informasi tentang pengecualian Amazon Glacier dan pesan kesalahan, lihat. Respons Kesalahan

Contoh

Contoh: Mencantumkan Bagian dari Unggahan Multipart

Contoh berikut mencantumkan semua bagian dari unggah. Contoh mengirimkan permintaan `GET` HTTP ke URI ID unggahan multipart spesifik dari unggahan multipart yang sedang berlangsung dan mengembalikan hingga 1.000 bagian.

Contoh Permintaan

```
GET /-/vaults/examplevault/multipart-uploads/
OW2fM5iVylEpFEMM9_HpKowRapC3vn5sSL39_396UW9zLFUWVrnRHaPjUJddQ5OxSHVXjYtrN47NBZ-
khxOjyEXAMPLE HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
x-amz-glacier-version: 2012-06-01
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
```

Contoh Respons

Sebagai tanggapan, Amazon Glacier mengembalikan daftar bagian yang diunggah yang terkait dengan ID unggahan multibagian yang ditentukan. Dalam contoh ini, hanya ada dua bagian. Bidang

`Marker` yang dikembalikan adalah `null` yang menunjukkan tidak ada lagi bagian dari unggahan multipart.

```
HTTP/1.1 200 OK
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:00:00 GMT
Content-Type: application/json
Content-Length: 412

{
    "ArchiveDescription" : "archive description",
    "CreationDate" : "2012-03-20T17:03:43.221Z",
    "Marker": null,
    "MultipartUploadId" :
 "OW2fM5iVylEpFEMM9_HpKowRapC3vn5sSL39_396UW9zLFUWVrnRHaPjUJddQ5OxSHVXjYtrN47NBZ-
khxOjyEXAMPLE",
    "PartSizeInBytes" : 4194304,
    "Parts" :
    [ {
      "RangeInBytes" : "0-4194303",
      "SHA256TreeHash" : "01d34dabf7be316472c93b1ef80721f5d4"
      },
      {
      "RangeInBytes" : "4194304-8388607",
      "SHA256TreeHash" : "0195875365afda349fc21c84c099987164"
      }],
    "VaultARN" : "arn:aws:glacier:us-west-2:012345678901:vaults/demo1-vault"
}
```

Contoh: Mencantumkan Bagian dari Unggahan Multipart (Menentukan Penanda dan Parameter Permintaan Batas)

Contoh berikut menunjukkan cara menggunakan pemberian nomor halaman untuk mendapatkan sejumlah hasil terbatas. Contoh mengirimkan permintaan `GET` HTTP ke URI ID unggahan multipart spesifik dari unggahan multipart yang sedang berlangsung untuk mengembalikan satu bagian. Parameter `marker` yang memulai menentukan di bagian mana untuk memulai daftar bagian. Anda bisa mendapatkan nilai `marker` dari respons permintaan sebelumnya untuk daftar bagian. Selain itu, dalam contoh ini, parameter `limit` diatur ke 1 dan mengembalikan satu bagian. Perhatikan bahwa bidang `Marker` bukan `null`, yang menunjukkan setidaknya ada satu bagian lagi yang harus didapatkan.

Contoh Permintaan

```
GET /-/vaults/examplevault/multipart-uploads/
OW2fM5iVylEpFEMM9_HpKowRapC3vn5sSL39_396UW9zLFUWVrnRHaPjUJddQ5OxSHVXjYtrN47NBZ-
khxOjyEXAMPLE?marker=1001&limit=1 HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
x-amz-glacier-version: 2012-06-01
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
```

Contoh Respons

Sebagai tanggapan, Amazon Glacier menampilkan daftar bagian yang diunggah yang terkait dengan ID unggahan multibagian yang sedang berlangsung yang ditentukan.

```
HTTP/1.1 200 OK
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:00:00 GMT
Content-Type: text/json
Content-Length: 412

{
    "ArchiveDescription" : "archive description 1",
    "CreationDate" : "2012-03-20T17:03:43.221Z",
    "Marker": "MfgsKHVjbQ6EldVl72bn3_n5h2TaGZQUO-Qb3B9j3TITf7WajQ",
    "MultipartUploadId" :
 "OW2fM5iVylEpFEMM9_HpKowRapC3vn5sSL39_396UW9zLFUWVrnRHaPjUJddQ5OxSHVXjYtrN47NBZ-
khxOjyEXAMPLE",
    "PartSizeInBytes" : 4194304,
    "Parts" :
    [ {
      "RangeInBytes" : "4194304-8388607",
      "SHA256TreeHash" : "01d34dabf7be316472c93b1ef80721f5d4"
      }],
    "VaultARN" : "arn:aws:glacier:us-west-2:012345678901:vaults/demo1-vault"
}
```

Bagian Terkait

- Memulai Unggahan Multipart (POST multipart-uploads)

- Mengunggah Bagian (PUT UploadID)
- Menyelesaikan Unggahan Multipart (POST uploadID)
- Membatalkan Unggahan Multipart (DELETE uploadID)
- Mencantumkan Unggahan Multipart (GET multipart-uploads)
- Mengunggah Arsip Besar dalam Beberapa Bagian (Unggahan Multipart)
- Identity and Access Management untuk Amazon Glacier

Mencantumkan Unggahan Multipart (GET multipart-uploads)

Deskripsi

Operasi unggahan multipart ini mencantumkan unggahan multiparti yang sedang berlangsung untuk vault yang ditentukan. Unggahan multipart yang sedang berlangsung adalah unggahan multipart yang telah dimulai menggunakan memulai permintaan Memulai Unggahan Multipart (POST multipart-uploads), tetapi belum diselesaikan atau dihentikan. Daftar yang dikembalikan dalam respons Cantumkan Unggahan Multipart memiliki urutan tidak terjamin.

Operasi Cantumkan Unggahan Multipart mendukung pemberian nomor halaman. Secara default, operasi ini mengembalikan hingga 50 unggahan multipart dalam respons. Anda harus selalu memeriksa bidang `marker` dalam isi respons untuk penanda tempat untuk melanjutkan daftar; jika tidak ada lagi item, bidang `marker` adalah `null`.

Jika tidak null, untuk mengambil kumpulan unggahan multipart berikutnya, Anda mengirim permintaan Unggahan Multipart Daftar lainnya dengan parameter `marker` permintaan yang disetel ke nilai penanda Amazon Glacier (Amazon Glacier) yang dikembalikan sebagai tanggapan atas permintaan Unggahan Multipart Daftar sebelumnya. `marker`

Perhatikan perbedaan antara operasi ini dan operasi Mencantumkan Bagian (GET uploadID)). Operasi Cantumkan Unggahan Multipart mencantumkan semua unggahan multipart untuk vault. Operasi Cantumkan Bagian mengembalikan bagian unggahan multiparti spesifik yang diidentifikasi oleh ID Unggah.

Untuk informasi tentang unggahan multipart, lihat Mengunggah Arsip Besar dalam Beberapa Bagian (Unggahan Multipart).

Permintaan

Sintaksis

Untuk mencantumkan unggahan multipart, kirim permintaan `GET` ke URI dari subsumber daya `multipart-uploads` vault. Anda dapat secara opsional menentukan parameter `marker` dan `limit`.

```
GET /AccountId/vaults/VaultName/multipart-uploads HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: Date
Authorization: SignatureValue
x-amz-glacier-version: 2012-06-01
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID akun yang memiliki brankas. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menggunakan ID akun, jangan sertakan tanda hubung ('-') di ID.

Parameter Permintaan

Nama	Deskripsi	Wajib
`limit`	Menentukan jumlah maksimum unggahan yang dikembalikan dalam isi respons. Jika tidak ditentukan, operasi Cantumkan Daftar akan mengembalikan hingga 50 unggahan. Tipe: String Batasan: Nilai bilangan bulat minimum 1. Nilai bilangan bulat maksimum `50`.	Tidak
`marker`		Tidak

Nama	Deskripsi	Wajib
	String buram yang digunakan untuk pemberian nomor halaman. `marker` menentukan unggahan tempat cantuman unggahan harus dimulai. Dapatkan nilai `marker` dari respons Cantumkan Unggahan sebelumnya. Anda hanya perlu menyertakan `marker` jika Anda melanjutkan pemberian nomor halaman dari hasil yang dimulai dalam permintaan Cantumkan Unggahan sebelumnya. Tipe: String Batasan: Tidak Ada	

Header Permintaan

Operasi ini hanya menggunakan header respons yang umum untuk sebagian besar respons. Untuk informasi tentang header permintaan umum, lihat Header Respons Umum.

Isi Permintaan

Operasi ini tidak memiliki isi permintaan.

Respons

Sintaksis

```
HTTP/1.1 200 OK
x-amzn-RequestId: x-amzn-RequestId
Date: Date
Content-Type: application/json
Content-Length: Length

{
  "Marker": String,
  "UploadsList" : [
    {
      "ArchiveDescription": String,
      "CreationDate": String,
      "MultipartUploadId": String,
      "PartSizeInBytes": Number,
```

```
      "VaultARN": String
    },
   ...
  ]
}
```

Header Respons

Operasi ini hanya menggunakan header respons yang umum untuk sebagian besar respons. Untuk informasi tentang header permintaan umum, lihat Header Respons Umum.

Isi Respons

Isi respons berisi bidang JSON berikut.

ArchiveDescription

Deskripsi arsip yang ditentukan dalam permintaan Mulai Unggahan Multipart. Bidang ini adalah `null` jika tidak ada deskripsi arsip yang ditentukan dalam operasi Mulai Unggahan Multipart.

Tipe: String

CreationDate

Waktu UTC unggahan multipart dimulai.

Tipe: String. Representasi string dalam format tanggal ISO 8601, misalnya `2013-03-20T17:03:43.221Z`.

Penanda

String buram yang mewakili tempat untuk melanjutkan pemberian nomor halaman hasil. Anda menggunakan `marker` dalam permintaan Cantumkan Unggahan Multipart baru untuk mendapatkan lebih banyak unggahan dalam daftar. Jika tidak ada lagi unggahan, nilai ini adalah `null`.

Tipe: String

PartSizeInBytes

Ukuran bagian yang ditentukan dalam permintaan Memulai Unggahan Multipart (POST multipart-uploads). Ini adalah ukuran semua bagian dalam unggahan kecuali bagian terakhir, yang mungkin lebih kecil dari ukuran ini.

Tipe: Angka

MultipartUploadId

ID unggahan multipart.

Tipe: String

UploadsList

Daftar metadata tentang objek unggahan multipart. Setiap item dalam daftar berisi serangkaian pasangan nama nilai untuk unggahan yang sesuai, termasuk `ArchiveDescription`, `CreationDate`, `MultipartUploadId`, `PartSizeInBytes`, dan `VaultARN`.

Tipe: Array

Vaultarn

Amazon Resource Name (ARN) dari vault yang berisi arsip.

Tipe: String

Kesalahan

Untuk informasi tentang pengecualian Amazon Glacier dan pesan kesalahan, lihat. Respons Kesalahan

Contoh

Contoh: Mencantumkan Semua Unggahan Multipart

Contoh berikut mencantumkan semua unggahan multipart yang sedang berlangsung untuk vault. Contoh ini menunjukkan permintaan `GET` HTTP ke URI dari subsumber daya `multipart-uploads` vault tertentu. Karena parameter `marker` dan `limit` tidak ditentukan dalam permintaan, hingga 1.000 unggahan multipart yang sedang berlangsung dikembalikan.

Contoh Permintaan

```
GET /-/vaults/examplevault/multipart-uploads HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
x-amz-glacier-version: 2012-06-01
```

```
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
```

Contoh Respons

Sebagai tanggapan Amazon Glacier mengembalikan daftar semua unggahan multipart yang sedang berlangsung untuk vault yang ditentukan. Bidang `marker` adalah `null`, yang menunjukkan tidak ada lagi unggahan ke daftar.

```
HTTP/1.1 200 OK
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:00:00 GMT
Content-Type: application/json
Content-Length: 1054

{
  "Marker": null,
  "UploadsList": [
    {
      "ArchiveDescription": "archive 1",
      "CreationDate": "2012-03-19T23:20:59.130Z",
      "MultipartUploadId":
 "xsQdFIRsfJr20CW2AbZBKpRZAFTZSJIMtL2hYf8mvp8dM0m4RUzlaqoEye6g3h3ecqB_zqwB7zLDMeSWhwo65re4C4Ev"
      "PartSizeInBytes": 4194304,
      "VaultARN": "arn:aws:glacier:us-west-2:012345678901:vaults/examplevault"
    },
    {
      "ArchiveDescription": "archive 2",
      "CreationDate": "2012-04-01T15:00:00.000Z",
      "MultipartUploadId": "nPyGOnyFcx67qqX7E-0tSGiRi88hHMOwOxR-
_jNyM6RjVMFfV29lFqZ3rNsSaWBugg6OP92pRtufeHdQH7ClIpSF6uJc",
      "PartSizeInBytes": 4194304,
      "VaultARN": "arn:aws:glacier:us-west-2:012345678901:vaults/examplevault"
    },
    {
      "ArchiveDescription": "archive 3",
      "CreationDate": "2012-03-20T17:03:43.221Z",
      "MultipartUploadId": "qt-RBst_7yO8gVIonIBsAxr2t-db0pE4s8MNeGjKjGdNpuU-
cdSAcqG62guwV9r5jh5mLyFPzFEitTpNE7iQfHiu1XoV",
      "PartSizeInBytes": 4194304,
      "VaultARN": "arn:aws:glacier:us-west-2:012345678901:vaults/examplevault"
    }
```

```
  ]
}
```

Contoh: Mencantumkan Bagian dari Unggahan Multipart

Contoh berikut menunjukkan cara menggunakan pemberian nomor halaman untuk mendapatkan sejumlah hasil terbatas. Contoh ini menunjukkan permintaan GET HTTP ke URI dari subsumber daya `multipart-uploads` untuk vault tertentu. Dalam contoh ini, parameter `limit` diatur ke 1, yang berarti hanya satu unggahan yang dikembalikan dalam daftar, dan parameter `marker` menunjukkan ID unggahan multipart tempat daftar yang dikembalikan dimulai.

Contoh Permintaan

```
GET /-/vaults/examplevault/multipart-uploads?
limit=1&marker=xsQdFIRsfJr20CW2AbZBKpRZAFTZSJIMtL2hYf8mvp8dM0m4RUzlaqoEye6g3h3ecqB_zqwB7zLDMeSW
 HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
x-amz-glacier-version: 2012-06-01
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
```

Contoh Respons

Sebagai tanggapan, Amazon Glacier (Amazon Glacier) mengembalikan daftar tidak lebih dari dua unggahan multipart yang sedang berlangsung untuk vault yang ditentukan, dimulai dari penanda yang ditentukan dan menampilkan dua hasil.

```
HTTP/1.1 200 OK
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:00:00 GMT
Content-Type: application/json
Content-Length: 470

{
  "Marker": "qt-RBst_7yO8gVIonIBsAxr2t-db0pE4s8MNeGjKjGdNpuU-
cdSAcqG62guwV9r5jh5mLyFPzFEitTpNE7iQfHiu1XoV",
  "UploadsList" : [
    {
      "ArchiveDescription": "archive 2",
      "CreationDate": "2012-04-01T15:00:00.000Z",
```

```
      "MultipartUploadId": "nPyGOnyFcx67qqX7E-0tSGiRi88hHMOwOxR-
_jNyM6RjVMFfV29lFqZ3rNsSaWBugg6OP92pRtufeHdQH7ClIpSF6uJc",
      "PartSizeInBytes": 4194304,
      "VaultARN": "arn:aws:glacier:us-west-2:012345678901:vaults/examplevault"
    }
  ]
}
```

Bagian Terkait

- Memulai Unggahan Multipart (POST multipart-uploads)
- Mengunggah Bagian (PUT UploadID)
- Menyelesaikan Unggahan Multipart (POST uploadID)
- Membatalkan Unggahan Multipart (DELETE uploadID)
- Mencantumkan Bagian (GET uploadID)
- Mengunggah Arsip Besar dalam Beberapa Bagian (Unggahan Multipart)
- Identity and Access Management untuk Amazon Glacier

Mengunggah Bagian (PUT UploadID)

Deskripsi

Operasi unggahan multipart ini mengunggah bagian dari arsip. Anda dapat mengunggah bagian arsip dalam urutan apa pun karena dalam permintaan Unggah Bagian, Anda menentukan rentang byte dalam arsip yang disusun yang akan diunggah di bagian ini. Anda juga dapat mengunggah bagian ini secara paralel. Anda dapat mengunggah hingga 10.000 bagian untuk unggahan multipart.

Untuk informasi tentang unggahan multipart, lihat Mengunggah Arsip Besar dalam Beberapa Bagian (Unggahan Multipart).

Amazon Glacier (Amazon Glacier) menolak permintaan bagian unggah Anda jika salah satu dari kondisi berikut ini benar:

- SHA256 hash pohon tidak cocok— Untuk memastikan bahwa data bagian tidak rusak dalam transmisi, Anda menghitung hash SHA256 pohon dari bagian tersebut dan memasukkannya

ke dalam permintaan Anda. Setelah menerima data bagian, Amazon Glacier juga menghitung SHA256 hash pohon. Jika dua nilai hash tidak cocok, operasi gagal. Untuk informasi tentang menghitung hash SHA256 pohon, lihatChecksum Komputasi.

- SHA256 hash linier tidak cocok— Diperlukan untuk otorisasi, Anda menghitung hash SHA256 linier dari seluruh muatan yang diunggah dan memasukkannya ke dalam permintaan Anda. Untuk informasi tentang menghitung hash SHA256 linier, lihatChecksum Komputasi.
- Ukuran bagian tidak cocok—Ukuran setiap bagian kecuali yang terakhir harus sesuai dengan ukuran yang ditentukan dalam permintaan Memulai Unggahan Multipart (POST multipart-uploads) yang sesuai. Ukuran bagian terakhir harus berukuran sama dengan, atau lebih kecil dari, ukuran yang ditentukan.

> **Note**
>
> Jika Anda mengunggah bagian yang ukurannya lebih kecil dari ukuran bagian yang Anda tentukan dalam permintaan mulai unggahan multipart Anda dan bagian tersebut bukan bagian terakhir, permintaan unggah bagian akan berhasil. Namun, permintaan Selesaikan Unggahan Multipart berikutnya akan gagal.

- Rentang tidak selaras—Nilai rentang byte dalam permintaan tidak selaras dengan ukuran bagian yang ditentukan dalam permintaan mulai yang sesuai. Sebagai contoh, jika Anda menentukan ukuran bagian sebesar 4194304 byte (4 MB), 0 hingga 4194303 byte (4 MB —1) dan 4194304 (4 MB) hingga 8388607 (8 MB —1) adala rentang bagian yang valid. Namun, jika Anda menetapkan nilai rentang sebesar 2 MB hingga 6 MB, rentang tidak selaras dengan ukuran bagian dan unggahan akan gagal.

Operasi ini bersifat idempoten. Jika Anda mengunggah bagian yang sama beberapa kali, data yang disertakan dalam permintaan terbaru akan menimpa data yang diunggah sebelumnya.

Permintaan

Anda mengirim permintaan PUT HTTP ini ke URI ID unggah yang dikembalikan oleh permintaan Mulai Unggahan Multipart Anda. Amazon Glacier menggunakan ID unggahan untuk mengaitkan unggahan bagian dengan unggahan multibagian tertentu. Permintaan harus menyertakan hash SHA256 pohon dari data bagian (`x-amz-SHA256-tree-hash`header), hash SHA256 linier dari seluruh payload (`x-amz-content-sha256`header), rentang byte (`Content-Range`header), dan panjang bagian dalam byte (header). `Content-Length`

Sintaksis

```
PUT /AccountId/vaults/VaultName/multipart-uploads/uploadID HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: Date
Authorization: SignatureValue
Content-Range: ContentRange
Content-Length: PayloadSize
Content-Type: application/octet-stream
x-amz-sha256-tree-hash: Checksum of the part
x-amz-content-sha256: Checksum of the entire payload
x-amz-glacier-version: 2012-06-01
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID akun yang memiliki brankas. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menggunakan ID akun, jangan sertakan tanda hubung ('-') di ID.

Parameter Permintaan

Operasi ini tidak menggunakan parameter permintaan.

Header Permintaan

Operasi ini menggunakan header permintaan berikut, selain header permintaan yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header permintaan umum, lihat Header Permintaan Umum.

Nama	Deskripsi	Wajib
`Content-Length`	Mengidentifikasi panjang bagian dalam byte. Tipe: String Default: Tidak ada	Tidak

Nama	Deskripsi	Wajib
	Batasan: Tidak Ada	
`Content-Range`	Mengidentifikasi rentang byte dalam arsip yang disusun yang akan diunggah di bagian ini. Amazon Glacier menggunakan informasi ini untuk mengumpulkan arsip dalam urutan yang tepat. Format header ini mengikuti RFC 2616. Contoh header adalah `Content-Range:bytes 0-4194303/*` . Tipe: String Default: Tidak ada Batasan: Rentang tidak dapat lebih besar dari ukuran bagian yang Anda tentukan ketika Anda memulai unggahan multipart.	Ya
`x-amz-content-sha256`	SHA256 Checksum (hash linier) dari payload yang diunggah. Ini bukan nilai yang sama seperti yang Anda tentukan di header `x-amz-sha256-tree-hash` . Tipe: String Default: Tidak ada Batasan: Tidak Ada	Ya

Nama	Deskripsi	Wajib
`x-amz-sha256-tree-hash`	Menentukan hash SHA256 pohon dari data yang diunggah. Untuk informasi tentang menghitung hash SHA256 pohon, lihatChecksum Komputasi. Tipe: String Default: Tidak ada Batasan: Tidak Ada	Ya

Isi Permintaan

Isi permintaan berisi data untuk diunggah.

Respons

Setelah pengunggahan bagian yang berhasil, Amazon Glacier mengembalikan respons. `204 No Content`

Sintaksis

```
HTTP/1.1 204 No Content
x-amzn-RequestId: x-amzn-RequestId
Date: Date
x-amz-sha256-tree-hash: ChecksumComputedByAmazonGlacier
```

Header Respons

Respons yang sukses termasuk header respon berikut, selain header respons yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header respons umum, lihat Header Respons Umum.

Nama	Penjelasan
`x-amz-sha256-tree-hash`	Hash SHA256 pohon yang dihitung Amazon Glacier untuk bagian yang diunggah.

Nama	Penjelasan
	Tipe: String

Isi Respons

Operasi ini tidak mengembalikan isi respons.

Contoh

Permintaan berikut mengunggah bagian sebesar 4 MB. Permintaan menetapkan rentang byte untuk membuat ini sebagai bagian pertama dalam arsip.

Contoh Permintaan

Contoh mengirimkan permintaan PUT HTTP untuk mengunggah bagian sebesar 4 MB. Permintaan dikirim ke URI ID Unggah yang dikembalikan oleh permintaan Mulai Unggahan Multipart Anda. Header `Content-Range` mengidentifikasi bagian sebagai bagian data 4 MB pertama dari arsip.

```
PUT /-/vaults/examplevault/multipart-uploads/
OW2fM5iVylEpFEMM9_HpKowRapC3vn5sSL39_396UW9zLFUWVrnRHaPjUJddQ5OxSHVXjYtrN47NBZ-
khxOjyEXAMPLE HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
Date: Wed, 10 Feb 2017 12:00:00 GMT
Content-Range:bytes 0-4194303/*
x-amz-sha256-tree-hash:c06f7cd4baacb087002a99a5f48bf953
x-amz-content-sha256:726e392cb4d09924dbad1cc0ba3b00c3643d03d14cb4b823e2f041cff612a628
Content-Length: 4194304
Authorization: Authorization=AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-content-sha256;x-amz-date;x-
amz-glacier-
version,Signature=16b9a9e220a37e32f2e7be196b4ebb87120ca7974038210199ac5982e792cace
```

Untuk mengunggah bagian selanjutnya, prosedurnya sama; Namun, Anda harus menghitung hash SHA256 pohon baru dari bagian yang Anda unggah dan juga menentukan rentang byte baru untuk menunjukkan ke mana bagian tersebut akan pergi dalam perakitan akhir. Permintaan berikut mengunggah bagian lain menggunakan ID unggah yang sama. Permintaan menentukan 4 MB berikutnya dari arsip setelah permintaan sebelumnya dan ukuran bagian sebesar 4 MB.

```
PUT /-/vaults/examplevault/multipart-uploads/
OW2fM5iVylEpFEMM9_HpKowRapC3vn5sSL39_396UW9zLFUWVrnRHaPjUJddQ5OxSHVXjYtrN47NBZ-
khxOjyEXAMPLE HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
Date: Wed, 10 Feb 2017 12:00:00 GMT
Content-Range:bytes 4194304-8388607/*
Content-Length: 4194304
x-amz-sha256-tree-hash:f10e02544d651e2c3ce90a4307427493
x-amz-content-sha256:726e392cb4d09924dbad1cc0ba3b00c3643d03d14cb4b823e2f041cff612a628
x-amz-glacier-version: 2012-06-01
Authorization: Authorization=AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20120525/
us-west-2/glacier/aws4_request, SignedHeaders=host;x-amz-content-sha256;x-amz-date;x-
amz-glacier-version,
 Signature=16b9a9e220a37e32f2e7be196b4ebb87120ca7974038210199ac5982e792cace
```

Suku cadang dapat diunggah dalam urutan apa pun; Amazon Glacier menggunakan spesifikasi rentang untuk setiap bagian untuk menentukan urutan untuk merakitnya.

Contoh Respons

```
HTTP/1.1 204 No Content
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
x-amz-sha256-tree-hash: c06f7cd4baacb087002a99a5f48bf953
Date: Wed, 10 Feb 2017 12:00:00 GMT
```

Bagian Terkait

- Memulai Unggahan Multipart (POST multipart-uploads)
- Mengunggah Bagian (PUT UploadID)
- Menyelesaikan Unggahan Multipart (POST uploadID)
- Membatalkan Unggahan Multipart (DELETE uploadID)
- Mencantumkan Unggahan Multipart (GET multipart-uploads)
- Mencantumkan Bagian (GET uploadID)
- Mengunggah Arsip Besar dalam Beberapa Bagian (Unggahan Multipart)
- Identity and Access Management untuk Amazon Glacier

Operasi Tugas

Berikut ini adalah operasi pekerjaan yang tersedia di Amazon Glacier.

Topik

- Mendeskripsikan Tugas (GET JobID)
- Mendapatkan Output Tugas (GET output)
- Memulai Tugas (POST jobs)
- Mencantumkan Tugas (GET jobs)

Mendeskripsikan Tugas (GET JobID)

Deskripsi

Operasi ini menampilkan informasi tentang pekerjaan yang Anda mulai sebelumnya, termasuk tanggal inisiasi pekerjaan, pengguna yang memulai pekerjaan, kode/pesan status pekerjaan, dan topik Simple Notification Service Amazon (Amazon SNS) untuk memberi tahu setelah Amazon Glacier (Amazon Glacier) menyelesaikan pekerjaan. Untuk informasi selengkapnya tentang memulai tugas, lihat Memulai Tugas (POST jobs).

> **Note**
>
> Operasi ini memungkinkan Anda memeriksa status tugas Anda. Namun, kami sangat menyarankan agar Anda menyiapkan topik Amazon SNS dan menentukannya dalam permintaan pekerjaan awal Anda sehingga Amazon Glacier dapat memberi tahu topik tersebut setelah menyelesaikan pekerjaan.

ID pekerjaan tidak akan kedaluwarsa setidaknya 24 jam setelah Amazon Glacier menyelesaikan pekerjaan.

Permintaan

Sintaksis

Untuk mendapatkan informasi tentang tugas, Anda menggunakan metode GET HTTP dan cakupan permintaan untuk tugas tertentu. Perhatikan bahwa jalur URI relatif sama dengan yang dikembalikan Amazon Glacier kepada Anda saat Anda memulai pekerjaan.

```
GET /AccountID/vaults/VaultName/jobs/JobID HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: date
Authorization: signatureValue
x-amz-glacier-version: 2012-06-01
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID akun yang memiliki brankas. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menggunakan ID akun, jangan sertakan tanda hubung ('-') di ID.

> **Note**
>
> Dalam permintaan, jika Anda menghilangkan `JobID`, respons mengembalikan daftar semua tugas aktif pada vault yang ditentukan. Untuk informasi selengkapnya tentang mencantumkan tugas, lihat Mencantumkan Tugas (GET jobs).

Parameter Permintaan

Operasi ini tidak menggunakan parameter permintaan.

Header Permintaan

Operasi ini hanya menggunakan header permintaan yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header permintaan umum, lihat Header Permintaan Umum.

Isi Permintaan

Operasi ini tidak memiliki isi permintaan.

Respons

Sintaksis

```
HTTP/1.1 201 Created
x-amzn-RequestId: x-amzn-RequestId
Date: Date
Content-Type: application/json
Content-Length: Length

{
    "Action": "string",
    "ArchiveId": "string",
    "ArchiveSHA256TreeHash": "string",
    "ArchiveSizeInBytes": number,
    "Completed": boolean,
    "CompletionDate": "string",
    "CreationDate": "string",
    "InventoryRetrievalParameters": {
        "EndDate": "string",
        "Format": "string",
        "Limit": "string",
        "Marker": "string",
        "StartDate": "string"
    },
    "InventorySizeInBytes": number,
    "JobDescription": "string",
    "JobId": "string",
    "JobOutputPath": "string",
    "OutputLocation": {
        "S3": {
            "AccessControlList": [
                {
                    "Grantee": {
                        "DisplayName": "string",
                        "EmailAddress": "string",
                        "ID": "string",
                        "Type": "string",
                        "URI": "string"
                    },
```

```
                    "Permission": "string"
                }
            ],
            "BucketName": "string",
            "CannedACL": "string",
            "Encryption": {
                "EncryptionType": "string",
                "KMSContext": "string",
                "KMSKeyId": "string"
            },
            "Prefix": "string",
            "StorageClass": "string",
            "Tagging": {
                "string": "string"
            },
            "UserMetadata": {
                "string": "string"
            }
        }
    },
    "RetrievalByteRange": "string",
    "SelectParameters": {
        "Expression": "string",
        "ExpressionType": "string",
        "InputSerialization": {
            "csv": {
                "Comments": "string",
                "FieldDelimiter": "string",
                "FileHeaderInfo": "string",
                "QuoteCharacter": "string",
                "QuoteEscapeCharacter": "string",
                "RecordDelimiter": "string"
            }
        },
        "OutputSerialization": {
            "csv": {
                "FieldDelimiter": "string",
                "QuoteCharacter": "string",
                "QuoteEscapeCharacter": "string",
                "QuoteFields": "string",
                "RecordDelimiter": "string"
            }
        }
    },
```

```
    "SHA256TreeHash": "string",
    "SNSTopic": "string",
    "StatusCode": "string",
    "StatusMessage": "string",
    "Tier": "string",
    "VaultARN": "string"
}
```

Header Respons

Operasi ini hanya menggunakan header respons yang umum untuk sebagian besar respons. Untuk informasi tentang header permintaan umum, lihat Header Respons Umum.

Isi Respons

Isi respons berisi bidang JSON berikut.

Tindakan

Jenis tugas. Baik `ArchiveRetrieval`, `InventoryRetrieval`, maupun `Select`.

Tipe: String

ArchiveId

ID arsipyang diminta untuk memilih atau mengarsipkan tugas pengambilan. Jika tidak, bidang ini adalah `null`.

Tipe: String

Arsip SHA256 TreeHash

Hash SHA256 pohon dari seluruh arsip untuk pekerjaan pengambilan arsip. Untuk tugas pengambilan inventaris, bidang ini adalah `null`.

Tipe: String

ArchiveSizeInBytes

Untuk tugas `ArchiveRetrieval`, ini berukuran dalam byte dari arsip yang diminta untuk diunduh. Untuk tugas `InventoryRetrieval`, nilai adalah `null`.

Tipe: Angka

Selesai

Status tugas. Ketika arsip atau tugas pengambilan inventaris selesai, Anda mendapatkan output tugas menggunakan Mendapatkan Output Tugas (GET output).

Tipe: Boolean

CompletionDate

Waktu Universal Terkoordinasi (UTC) saat permintaan tugas selesai. Sewaktu tugas sedang berlangsung, nilai adalah nol.

Tipe: String

CreationDate

Waktu UTC saat tugas dibuat.

Tipe: Representasi string dalam format tanggal ISO 8601, misalnya `2013-03-20T17:03:43.221Z`.

InventoryRetrievalParameters

Parameter input yang digunakan untuk pengambilan berbagai inventaris.

Tipe: Objek InventoryRetrievalJobInput

InventorySizeInBytes

Untuk tugas `InventoryRetrieval`, ini berukuran dalam byte dari inventaris yang diminta untuk diunduh. Untuk tugas `ArchiveRetrieval` atau `Select`, nilai adalah `null`.

Tipe: Angka

JobDescription

Deskripsi tugas yang Anda berikan ketika Anda memulai tugas.

Tipe: String

JobId

ID yang mengidentifikasi pekerjaan di Amazon Glacier.

Tipe: String

JobOutputPath

Berisi lokasi output tugas.

Tipe: String

OutputLocation

Objek yang berisi informasi tentang lokasi tempat hasil pilih tugas dan kesalahan disimpan.

Tipe: Objek OutputLocation

RetrievalByteRange

Rentang byte yang diambil untuk pekerjaan pengambilan arsip dalam bentuk "*StartByteValue*-*EndByteValue*." Jika Anda tidak menentukan rentang dalam pengambilan arsip, maka seluruh arsip diambil; juga StartByteValuesama dengan 0, dan EndByteValuesama dengan ukuran arsip dikurangi 1. Untuk tugas pengambilan atau pemilihan inventaris, bidang ini adalah `null`.

Tipe: String

SelectParameters

Objek yang berisi informasi tentang parameter yang digunakan untuk select.

Tipe: Objek SelectParameters

SHA256TreeHash

Nilai hash SHA256 pohon untuk rentang arsip yang diminta. Jika permintaan Memulai Tugas (POST jobs) untuk arsip menentukan rentang tree-hash selaras, bidang ini mengembalikan nilai. Untuk informasi selengkapnya tentang penyelarasan tree-hash untuk pengambilan rentang arsip, lihat Menerima Checksum Saat Mengunduh Data.

Untuk kasus tertentu ketika seluruh arsip diambil, nilai ini sama dengan nilai `ArchiveSHA256TreeHash`.

Bidang ini adalah `null` dalam situasi berikut:

- Tugas pengambilan arsip yang menentukan rentang yang bukan tree-hash selaras.
- Tugas arsip yang menentukan rentang yang sama dengan status seluruh arsip dan tugas adalah `InProgress`.
- Tugas inventaris.
- Pilih tugas.

Tipe: String

SNSTopic

Topik Amazon SNS yang menerima notifikasi.

Tipe: String

StatusCode

Kode yang menunjukkan status tugas.

Nilai yang Valid: `InProgress` | `Succeeded` | `Failed`

Tipe: String

StatusMessage

Pesan mudah diingar yang menjelaskan status tugas.

Tipe: String

Tingkat

Tingkat akses data yang digunakan untuk pengambilan pilih atau arsip.

Nilai yang Valid: `Bulk` | `Expedited` | `Standard`

Tipe: String

Vaultarn

Amazon Resource Name (ARN) dari vault yang tugasnya adalah subsumber daya.

Tipe: String

Kesalahan

Untuk informasi tentang pengecualian Amazon Glacier dan pesan kesalahan, lihat. Respons Kesalahan

Contoh

Contoh berikut menunjukkan permintaan untuk tugas yang mengambil arsip.

Permintaan Contoh: Mendapatkan deskripsi tugas

```
GET /-/vaults/examplevault/jobs/HkF9p6o7yjhFx-
K3CGl6fuSm6VzW9T7esGQfco8nUXVYwS0jlb5gq1JZ55yHgt5vP54ZShjoQzQVVh7vEXAMPLEjobID HTTP/1.1
```

```
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
x-amz-glacier-version: 2012-06-01
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
```

Contoh Respons

Isi respon termasuk JSON yang menggambarkan tugas yang ditentukan. Perhatikan bahwa untuk tugas pengambilan inventaris dan pengambilan arsip, bidang JSON adalah sama. Namun, ketika bidang tidak berlaku untuk tipe tugas, nilainya adalah `null`. Berikut ini adalah contoh respons untuk tugas pengambilan arsip. Perhatikan hal-hal berikut:

- Nilai bidang `Action` adalah `ArchiveRetrieval`.
- Bidang `ArchiveSizeInBytes` menunjukkan ukuran arsip yang diminta dalam tugas pengambilan arsip.
- `ArchiveSHA256TreeHash`Bidang menunjukkan hash SHA256 pohon untuk seluruh arsip.
- Bidang `RetrievalByteRange` menunjukkan rentang yang diminta dalam permintaan Mulai Tugas. Dalam contoh ini, seluruh arsip diminta.
- `SHA256TreeHash`Bidang menunjukkan hash SHA256 pohon untuk rentang yang diminta dalam permintaan Initiate Job. Dalam contoh ini, ini sama dengan bidang `ArchiveSHA256TreeHash`, yang berarti seluruh arsip diminta.
- Nilai bidang `InventorySizeInBytes` adalah `null` karena tidak berlaku untuk tugas pengambilan arsip.

```
HTTP/1.1 200 OK
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:00:00 GMT
Content-Type: application/json
Content-Length: 419
{
  "Action": "ArchiveRetrieval",
  "ArchiveId": "NkbByEejwEggmBz2fTHgJrg0XBoDfjP4q6iu87-
TjhqG6eGoOY9Z8i1_AUyUsuhPAdTqLHy8pTl5nfCFJmDl2yEZONi5L26Omw12vcs01MNGntHEQL8MBfGlqrEXAMPLEArchi
  "ArchiveSizeInBytes": 16777216,
  "ArchiveSHA256TreeHash":
 "beb0fe31a1c7ca8c6c04d574ea906e3f97b31fdca7571defb5b44dca89b5af60",
```

```
  "Completed": false,
  "CompletionDate": null,
  "CreationDate": "2012-05-15T17:21:39.339Z",
  "InventorySizeInBytes": null,
  "JobDescription": "My ArchiveRetrieval Job",
  "JobId": "HkF9p6o7yjhFx-
K3CGl6fuSm6VzW9T7esGQfco8nUXVYwS0jlb5gq1JZ55yHgt5vP54ZShjoQzQVVh7vEXAMPLEjobID",
  "RetrievalByteRange": "0-16777215",
  "SHA256TreeHash": "beb0fe31a1c7ca8c6c04d574ea906e3f97b31fdca7571defb5b44dca89b5af60",
  "SNSTopic": "arn:aws:sns:us-west-2:012345678901:mytopic",
  "StatusCode": "InProgress",
  "StatusMessage": "Operation in progress.",
  "Tier": "Bulk",
  "VaultARN": "arn:aws:glacier:us-west-2:012345678901:vaults/examplevault"
}
```

Berikut adalah contoh respons untuk tugas pengambilan inventaris. Perhatikan hal-hal berikut:

- Nilai bidang `Action` adalah `InventoryRetrieval`.
- Nilai bidang `ArchiveSizeInBytes`, `ArchiveSHA256TreeHash`, dan `RetrievalByteRange` adalah nol karena bidang ini tidak berlaku untuk tugas pengambilan inventaris.
- Nilai bidang `InventorySizeInBytes` adalah `null` karena tugas masih sedang berlangsung, dan belum sepenuhnya mempersiapkan persediaan untuk diunduh. Jika tugas selesai sebelum Anda menjelaskan permintaan tugas, bidang ini akan memberikan ukuran output.

```
{
   "Action": "InventoryRetrieval",
   "ArchiveId": null,
   "ArchiveSizeInBytes": null,
   "ArchiveSHA256TreeHash": null,
   "Completed": false,
   "CompletionDate": null,
   "CreationDate": "2012-05-15T23:18:13.224Z",
   "InventorySizeInBytes": null,
   "JobDescription": "Inventory Description",
   "JobId": "HkF9p6o7yjhFx-
K3CGl6fuSm6VzW9T7esGQfco8nUXVYwS0jlb5gq1JZ55yHgt5vP54ZShjoQzQVVh7vEXAMPLEjobID",
   "RetrievalByteRange": null,
   "SHA256TreeHash": null,
   "SNSTopic": "arn:aws:sns:us-west-2:012345678901:mytopic",
```

```
   "StatusCode": "InProgress",
   "StatusMessage": "Operation in progress.",
   "VaultARN": "arn:aws:glacier:us-west-2:012345678901:vaults/examplevault"
}
```

Berikut adalah contoh respons untuk tugas pengambilan inventaris yang selesai yang berisi penanda yang digunakan untuk melanjutkan pemberian nomor halaman dari pengambilan inventaris vault.

```
{
    "Action": "InventoryRetrieval",
    "ArchiveId": null,
    "ArchiveSHA256TreeHash": null,
    "ArchiveSizeInBytes": null,
    "Completed": true,
    "CompletionDate": "2013-12-05T21:51:13.591Z",
    "CreationDate": "2013-12-05T21:51:12.281Z",
    "InventorySizeInBytes": 777062,
    "JobDescription": null,
    "JobId": "sCC2RZNBF2nildYD_roe0J9bHRdPQUbDRkmTdg-mXi2u3lc49uW6TcEhDF2D9pB2phx-
BN30JaBru7PMyOlfXHdStzu8",
    "NextInventoryRetrievalMarker": null,
    "RetrievalByteRange": null,
    "SHA256TreeHash": null,
    "SNSTopic": null,
    "StatusCode": "Succeeded",
    "StatusMessage": "Succeeded",
    "Tier": "Bulk",
    "VaultARN": "arn:aws:glacier-devo:us-west-2:836579025725:vaults/inventory-
icecube-2",
    "InventoryRetrievalParameters": {
        "StartDate": "2013-11-12T13:43:12Z",
        "EndDate": "2013-11-20T08:12:45Z",
        "Limit": "120000",
        "Format": "JSON",
        "Marker":
 "vyS0t2jHQe5qbcDggIeD50chS1SXwYMrkVKo0KHiTUjEYxBGCqRLKaiySzdN7QXGVVV5XZpNVG67pCZ_uykQXFMLaxOSu
    },
}
```

Bagian Terkait

- Mendapatkan Output Tugas (GET output)

- Identity and Access Management untuk Amazon Glacier

Mendapatkan Output Tugas (GET output)

Deskripsi

Operas ini mengunduh output tugas yang Anda mulai menggunakan Memulai Tugas (POST jobs). Bergantung pada tipe tugas yang Anda tentukan ketika memulai tugas, output akan berupa isi arsip atau inventaris vault.

Anda dapat mengunduh semua output tugas atau mengunduh bagian output dengan menentukan rentang byte. Untuk tugas pengambilan arsip dan tugas, Anda harus memverifikasi ukuran yang diunguh terhadap ukuran yang dikembalikan dalam header dari respons Dapatkan Output Tugas.

Untuk tugas pengambilan arsip, Anda juga harus memastikan bahwa ukurannya sesuai dengan harapan Anda. Jika Anda mengunduh bagian output, ukuran yang diharapkan didasarkan pada rentang byte yang Anda tentukan. Sebagai contoh, jika Anda menentukan berbagai `bytes=0-1048575`, Anda harus memastikan ukuran unduhan Anda adalah 1.048,576 byte. Jika Anda mengunduh seluruh arsip, ukuran yang diharapkan adalah ukuran arsip saat Anda mengunggahnya ke Amazon Glacier (Amazon Glacier). Ukuran yang diharapkan juga dikembalikan dalam header dari respons Dapatkan Output Tugas.

Dalam kasus pekerjaan pengambilan arsip, tergantung pada rentang byte yang Anda tentukan, Amazon Glacier mengembalikan checksum untuk bagian data. Untuk memastikan bagian yang Anda unduh adalah data yang benar, hitung checksum pada klien, pastikan nilai-nilainya cocok, dan pastikan ukuran sesuai dengan harapan Anda.

ID pekerjaan tidak kedaluwarsa setidaknya 24 jam setelah Amazon Glacier menyelesaikan pekerjaan. Artinya, Anda dapat mengunduh hasil pekerjaan dalam periode 24 jam setelah Amazon Glacier menyelesaikan pekerjaan.

Permintaan

Sintaksis

Untuk mengambil output tugas, Anda mengirim permintaan `GET` HTTP ke URI `output` dari tugas tertentu.

```
GET /AccountId/vaults/VaultName/jobs/JobID/output HTTP/1.1
Host: glacier.Region.amazonaws.com
```

```
Date: Date
Authorization: SignatureValue
Range: ByteRangeToRetrieve
x-amz-glacier-version: 2012-06-01
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID akun yang memiliki brankas. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menggunakan ID akun, jangan sertakan tanda hubung ('-') di ID.

Parameter Permintaan

Operasi ini tidak menggunakan parameter permintaan.

Header Permintaan

Operasi ini menggunakan header permintaan berikut, selain header permintaan yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header permintaan umum, lihat Header Permintaan Umum.

Nama	Deskripsi	Wajib
`Range`	Rentang byte yang diambil dari output. Sebagai contoh, jika Anda ingin mengunduh byte 1.048,576 pertama, tentukan rentang sebagai `bytes=0-1048575` . Untuk informasi selengkapnya, buka Definisi Bidang Header Rentang. Rentang ini terkait dengan rentang yang ditentukan dalam permintaa n Mulai Tugas. Secara default, operasi ini akan mengunduh seluruh output. Jika output tugasnya besar, Anda dapat menggunakan header permintaan `Range` untuk mengambil sebagian dari output. Anda dapat mengunduh seluruh output dalam potongan byte yang lebih kecil. Misalnya, Anda memiliki output tugas sebesar	Tidak

Nama	Deskripsi	Wajib
	1 GB yang ingin Anda unduh dan Anda memutuskan untuk mengunduh potongan data sebesar 128 MB pada suatu waktu, total delapan permintaan Dapatkan Output Tugas. Anda akan menggunakan proses berikut untuk mengunduh output tugas: 1. Unduh potongan output sebesar 128 MB dengan menentukan rentang byte yang sesuai menggunakan header `Range`. Verifikasi semua 128 MB data diterima. 2. Bersama dengan data, respons akan mencakup checksum dari muatan. Anda menghitung checksum muatan pada klien dan membandingkannya dengan checksum yang Anda terima sebagai respons untuk memastikan Anda menerima semua data yang diharapkan. 3. Ulangi langkah 1 dan 2 untuk kedelapan potongan data output sebesar 128 MB, setiap kali menentukan rentang byte yang sesuai. 4. Setelah mengunduh semua bagian dari output tugas, Anda memiliki daftar delapan nilai checksum. Hitung hash pohon dari nilai-nilai ini untuk menemukan checksum dari seluruh output. Menggunakan operasi Mendeskripsikan Tugas (GET JobID), dapatkan informasi tugas dari tugas yang memberi Anda output. Tanggapan tersebut mencakup checksum dari seluruh arsip yang disimpan di Amazon Glacier. Anda membandingkan nilai ini dengan checksum yang Anda hitung untuk memastikan Anda sudah mengunduh seluruh konten arsip tanpa kesalahan. Tipe: String Default: Tidak ada Batasan: Tidak Ada	

Isi Permintaan

Operasi ini tidak memiliki isi permintaan.

Respons

Sintaksis

Untuk permintaan pengambilan yang mengembalikan semua data tugas, respons output tugas mengembalikan kode respons `200 OK`. Ketika konten parsial diminta, misalnya, jika Anda menentukan header `Range` dalam permintaan, kode respons`206 Partial Content` dikembalikan.

```
HTTP/1.1 200 OK
x-amzn-RequestId: x-amzn-RequestId
Date: Date
Content-Type: ContentType
Content-Length: Length
x-amz-sha256-tree-hash: ChecksumComputedByAmazonGlacier

[Body containing job output.]
```

Header Respons

Header	Deskripsi
`Content-Range`	Kisaran byte dikembalikan oleh Amazon Glacier. Jika hanya sebagian output yang diunduh, responsnya memberikan kisaran byte Amazon Glacier yang dikembalikan. Misalnya, `bytes 0-1048575/8388608` mengembalikan 1 MB pertama dari 8 MB. Untuk informasi selengkapnya tentang header `Content-Range` , buka Definisi Bidang Header Content-Range. Tipe: String
`Content-Type`	Content-Type bergantung pada apakah output tugas adalah arsip atau inventaris vault.

Header	Deskripsi
	• Untuk data arsip, Content-Type adalah `application/octet-stream` . • Untuk inventaris vault, jika Anda meminta format CSV saat memulai tugas, Content-Type adalah `text/csv`. Jika tidak, secara default, inventaris vault dikembalikan sebagai JSON, dan Content-Type adalah `application/json` . Tipe: String

Header	Deskripsi
`x-amz-sha256-tree-hash`	Checksum data dalam respons. Header ini dikembalikan hanya ketika mengambil output untuk tugas pengambilan arsip. Selanjutnya, header ini muncul ketika berbagai data yang diambil yang diminta dalam permintaan Mulai Tugas adalah hash pohon selaras dan rentang yang diunduh di Dapatkan Output Tugas juga merupakan hash pohon selaras. Untuk informasi selengkapnya tentang rentang hash pohon selaras, lihat Menerima Checksum Saat Mengunduh Data. Misalnya, jika dalam permintaan Mulai Tugas, Anda menentukan rentang hash pohon selaras yang diambil (yang mencakup seluruh arsip), Anda akan menerima checksum dari data yang Anda unduh berdasarkan kondisi berikut: • Anda mendapatkan seluruh rentang data yang diambil. • Anda meminta rentang byte dari data yang diambil yang berukuran megabyte (1024 KB) yang dikalikan dengan pangkat 2 serta yang dimulai dan berakhir di beberapa ukuran rentang yang diminta. Sebagai contoh, jika Anda memiliki 3,1 MB data yang diambil dan Anda menentukan rentang untuk dikembalikan yang dimulai pada 1 MB dan berakhir pada 2 MB, `x-amz-sha256-tree-hash` dikembalikan sebagai header respons. • Anda meminta rentang untuk mengembalikan data yang diambil yang mengarah ke bagian akhir data, dan awal rentang adalah kelipatan dari ukuran rentang yang diambil yang dibulatkan ke pangkat dua berikutnya, tetapi tidak lebih kecil dari satu megabyte (1024 KB). Sebagai contoh, jika Anda memiliki 3,1 MB data yang diambil dan Anda menentukan rentang yang dimulai pada 2 MB dan berakhir pada 3,1 MB (akhir data), `x-amz-sha256-tree-hash` dikembalikan sebagai header respons.

Header	Deskripsi
	Tipe: String

Isi Respons

Amazon Glacier mengembalikan output pekerjaan di badan respons. Bergantung pada tipe tugas, output dapat berupa isi arsip atau inventaris vault. Dalam kasus inventaris vault, secara default daftar inventaris dikembalikan sebagai isi JSON berikut.

```
{
 "VaultARN": String,
 "InventoryDate": String,
 "ArchiveList": [
      {"ArchiveId": String,
       "ArchiveDescription": String,
       "CreationDate": String,
       "Size": Number,
       "SHA256TreeHash": String
      },
      ...
    ]
}
```

Jika Anda meminta format output nilai yang dipisahkan koma (CSV) saat Anda memulai tugas inventaris vault, inventaris vault dikembalikan dalam format CSV di isinya. Format CSV memiliki lima kolom "ArchiveId“," ArchiveDescription “," CreationDate “, “Ukuran”, dan "SHA256TreeHash" dengan definisi yang sama dengan bidang JSON yang sesuai.

> **Note**
>
> Dalam format CSV yang dikembalikan, bidang dapat dikembalikan dengan seluruh bidang tertutup dalam tanda kutip ganda. Bidang yang berisi koma atau tanda kutip ganda selalu dikembalikan tertutup dalam tanda kutip ganda. Misalnya, `my archive description,1` dikembalikan sebagai `"my archive description,1"`. Karakter kutip ganda yang berada dalam bidang tertutup kutip ganda yang dikembalikan adalah escaped dengan mendahuluinya dengan karakter garis miring terbalik. Misalnya, `my archive description,1"2` dikembalikan sebagai `"my archive description,1\"2"` dan `my`

`archive description,1\"2` dikembalikan sebagai `"my archive description,1\ \"2"`. Karakter garis miring terbalik bukan karakter escaped.

Isi respons JSON berisi bidang JSON berikut.

ArchiveDescription

Deskripsi arsip.

Tipe: String

ArchiveId

ID arsip.

Tipe: String

ArchiveList

Array metadata arsip. Setiap objek dalam array mewakili metadata untuk satu arsip yang terkandung dalam vault.

Tipe: Array

CreationDate

Tanggal dan waktu UTC saat arsip dibuat.

Tipe: Representasi string dalam format tanggal ISO 8601, misalnya `2013-03-20T17:03:43.221Z`.

InventoryDate

Tanggal dan waktu UTC dari inventaris terakhir untuk vault yang diselesaikan setelah perubahan ke vault. Meskipun Amazon Glacier menyiapkan inventaris brankas sekali sehari, tanggal inventaris hanya diperbarui jika ada penambahan arsip atau penghapusan ke brankas sejak inventaris terakhir.

Tipe: Representasi string dalam format tanggal ISO 8601, misalnya `2013-03-20T17:03:43.221Z`.

SHA256TreeHash

Hash pohon arsip.

Tipe: String

Ukuran

Ukuran dalam byte dari arsip.

Tipe: Angka

Vaultarn

Sumber daya Amazon Resource Name (ARN) asal pengambilan arsip diminta.

Tipe: String

Kesalahan

Untuk informasi tentang pengecualian Amazon Glacier dan pesan kesalahan, lihat. Respons Kesalahan

Contoh

Contoh berikut menunjukkan permintaan untuk tugas yang mengambil arsip.

Contoh 1: Mengunduh output

Contoh ini mengambil data yang disiapkan oleh Amazon Glacier sebagai tanggapan atas permintaan pekerjaan pengambilan arsip inisiat Anda.

Contoh Permintaan

```
GET /-/vaults/examplevault/jobs/HkF9p6o7yjhFx-
K3CGl6fuSm6VzW9T7esGQfco8nUXVYwS0jlb5gq1JZ55yHgt5vP54ZShjoQzQVVh7vEXAMPLEjobID/output
 HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
x-amz-glacier-version: 2012-06-01
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
```

Contoh Respons

Berikut adalah contoh respons dari tugas pengambilan arsip. Perhatikan bahwa header `Content-Type` adalah `application/octet-stream` dan header `x-amz-sha256-tree-hash` disertakan dalam respons, yang berarti semua data tugas dikembalikan.

```
HTTP/1.1 200 OK
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
x-amz-sha256-tree-hash:
 beb0fe31a1c7ca8c6c04d574ea906e3f97b31fdca7571defb5b44dca89b5af60
Date: Wed, 10 Feb 2017 12:00:00 GMT
Content-Type: application/octet-stream
Content-Length: 1048576

[Archive data.]
```

Berikut adalah contoh respons dari tugas pengambilan inventaris. Perhatikan bahwa header `Content-Type` adalah `application/json`. Perhatikan juga bahwa respons tidak termasuk header `x-amz-sha256-tree-hash`.

```
HTTP/1.1 200 OK
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:00:00 GMT
Content-Type: application/json
Content-Length: 906

{
 "VaultARN": "arn:aws:glacier:us-west-2:012345678901:vaults/examplevault",
 "InventoryDate": "2011-12-12T14:19:01Z",
 "ArchiveList": [
   {
     "ArchiveId": "DMTmICA2n5Tdqq5BV2z7og-
A20xnpAPKt3UXwWxdWsn_D6auTUrW6kwy5Qyj9xd1MCE1mBYvMQ63LWaT8yTMzMaCxB_9VBWrW4Jw4zsvg5kehAPDVKcppU
oA",
     "ArchiveDescription": "my archive1",
     "CreationDate": "2012-05-15T17:19:46.700Z",
     "Size": 2140123,
     "SHA256TreeHash":
 "6b9d4cf8697bd3af6aa1b590a0b27b337da5b18988dbcc619a3e608a554a1e62"
   },
   {
```

```
     "ArchiveId": "2lHzwhKhgF2JHyvCS-
ZRuF08IQLuyB4265Hs3AXj9MoAIhz7tbXAvcFeHusgU_hViO1WeCBe0N5lsYYHRyZ7rrmRkNRuYrXUs_sjl2K8ume_7mKO_
uHE1oHqaW9d37pabXrSA",
     "ArchiveDescription": "my archive2",
     "CreationDate": "2012-05-15T17:21:39.339Z",
     "Size": 2140123,
     "SHA256TreeHash":
 "7f2fe580edb35154041fa3d4b41dd6d3adaef0c85d2ff6309f1d4b520eeecda3"
   }
  ]
}
```

Contoh 2: Menguduh hanya sebagian output

Contoh ini hanya mengambil sebagian dari arsip yang disiapkan oleh Amazon Glacier sebagai tanggapan atas permintaan pekerjaan pengambilan arsip inisiat Anda. Permintaan menggunakan header `Range` opsional untuk hanya mengambil 1.024 byte pertama.

Contoh Permintaan

```
GET /-/vaults/examplevault/jobs/HkF9p6o7yjhFx-
K3CGl6fuSm6VzW9T7esGQfco8nUXVYwS0jlb5gq1JZ55yHgt5vP54ZShjoQzQVVh7vEXAMPLEjobID/output
 HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
Range: bytes=0-1023
x-amz-glacier-version: 2012-06-01
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
```

Contoh Respons

Respons yang berhasil berikut menunjukkan respons `206 Partial Content`. Dalam hal ini, respon juga menyertakan `Content-Range` header yang menentukan kisaran byte Amazon Glacier kembali.

```
HTTP/1.1 206 Partial Content
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:00:00 GMT
Content-Range: bytes 0-1023/8388608
Content-Type: application/octet-stream
Content-Length: 1024
```

```
[Archive data.]
```

Bagian Terkait

- Mendeskripsikan Tugas (GET JobID)
- Memulai Tugas (POST jobs)
- Identity and Access Management untuk Amazon Glacier

Memulai Tugas (POST jobs)

Operasi ini memulai jenis pekerjaan Amazon Glacier (Amazon Glacier) berikut:

- `archive-retrieval`— Mengambil arsip.
- `inventory-retrieval`— Menyediakan vault.

Topik

- Memulai Tugas Pengambilan Arsip atau Inventaris Vault
- Permintaan
- Respons
- Contoh
- Bagian Terkait

Memulai Tugas Pengambilan Arsip atau Inventaris Vault

Mengambil arsip atau inventaris vault adalah operasi asinkron yang mengharuskan Anda memulai tugas. Setelah dimulai, tugas tidak dapat dibatalkan. Pengambilan adalah proses dua langkah:

1. Mulai tugas pengambilan menggunakan operasi Memulai Tugas (POST jobs).

> **Important**
>
> Kebijakan pengambilan data dapat menyebabkan permintaan mulai tugas pengambilan Anda gagal dengan `PolicyEnforcedException`. Untuk informasi selengkapnya tentang kebijakan pengambilan data, lihat Kebijakan Pengambilan Data Amazon Glacier.

Untuk informasi selengkapnya tentang pengecualian `PolicyEnforcedException`, lihat Respons Kesalahan.

2. Setelah tugas selesai, unduh byte menggunakan operasi Mendapatkan Output Tugas (GET output).

Permintaan pengambilan dijalankan secara asinkron. Ketika Anda memulai tugas pengambilan, Amazon Glacier membuat tugas dan menghasilkan ID tugas dalam respons. Saat Amazon Glacier menyelesaikan pekerjaan, Anda bisa mendapatkan output pekerjaan (data arsip atau inventaris). Untuk informasi tentang mendapatkan output tugas, lihat operasi Mendapatkan Output Tugas (GET output).

Tugas harus selesai sebelum Anda bisa mendapatkan outputnya. Untuk menentukan kapan tugas selesai, Anda memiliki opsi berikut:

- Gunakan pemberitahuan Amazon SNS— Anda dapat menentukan topik Amazon SNS yang Amazon Glacier dapat memposting pemberitahuan setelah pekerjaan selesai. Anda dapat menentukan topik SNS per permintaan tugas. Pemberitahuan dikirim hanya setelah Amazon Glacier menyelesaikan pekerjaan. Selain menentukan topik SNS per permintaan tugas, Anda dapat mengonfigurasi notifikasi vault untuk vault sehingga notifikasi tugas dikirim untuk semua pengambilan. Untuk informasi selengkapnya, lihat Menetapkan Konfigurasi Notifikasi Vault (PUT notification-configuration).
- Dapatkan detail tugas— Anda dapat membuat permintaan Mendeskripsikan Tugas (GET JobID) untuk mendapatkan informasi status tugas saat tugas sedang berlangsung. Namun, penggunaan notifikasi Amazon SNS untuk menentukan kapan tugas selesai lebih efisien.

> **Note**
>
> Informasi yang Anda dapatkan melalui notifikasi sama dengan yang Anda dapatkan dengan memanggil Mendeskripsikan Tugas (GET JobID).

Jika untuk acara tertentu, Anda menambahkan konfigurasi notifikasi pada vault dan juga menentukan topik SNS dalam permintaan pekerjaan inisiat Anda, Amazon Glacier mengirimkan kedua notifikasi. Untuk informasi selengkapnya, lihat Menetapkan Konfigurasi Notifikasi Vault (PUT notification-configuration).

Inventaris Vault

Amazon Glacier memperbarui inventaris brankas kira-kira sekali sehari, dimulai pada hari Anda pertama kali mengunggah arsip ke lemari besi. Jika tidak ada penambahan arsip atau penghapusan vault sejak inventaris terakhir, tanggal inventaris tidak diperbarui. Saat Anda memulai pekerjaan untuk inventaris vault, Amazon Glacier mengembalikan inventaris terakhir yang dihasilkannya, yang merupakan point-in-time snapshot dan bukan data real-time.

Setelah Amazon Glacier membuat inventaris pertama untuk lemari besi, biasanya dibutuhkan setengah hari dan hingga sehari sebelum inventaris itu tersedia untuk diambil.

Anda mungkin merasa mengambil inventaris vault untuk setiap unggahan arsip tidak berguna. Namun, misalkan Anda memelihara database di sisi klien terkait metadata tentang arsip yang Anda unggah ke Amazon Glacier. Selanjutnya, Anda mungkin merasa inventaris vault berguna untuk mencocokkan informasi, jika perlu, dalam basis data Anda dengan inventaris vault aktual. Untuk informasi selengkapnya tentang bidang data yang dikembalikan dalam output tugas, lihat Isi Respons.

Pengambilan Inventaris Rentang

Anda dapat membatasi jumlah item inventaris yang diambil dengan memfilter tanggal pembuatan arsip atau dengan menetapkan batas.

Memfilter berdasarkan Tanggal Pembuatan Arsip

Anda dapat mengambil item inventaris untuk arsip yang dibuat antara `StartDate` dan `EndDate` dengan menentukan nilai untuk parameter ini dalam permintaan Mulai Tugas. Arsip yang dibuat di atau setelah `StartDate` dan sebelum `EndDate` dikembalikan. Jika Anda hanya memberikan `StartDate` tanpa `EndDate`, Anda mengambil inventaris untuk semua arsip yang dibuat pada atau setelah `StartDate`. Jika Anda hanya memberikan `EndDate` tanpa `StartDate`, Anda mengambil inventaris untuk semua arsip yang dibuat sebelum `EndDate`.

Membatasi Item Inventaris per Pengambilan

Anda dapat membatasi jumlah item inventaris yang dikembalikan dengan mengatur parameter `Limit` dalam permintaan Mulai Tugas. Output tugas inventaris berisi item inventaris sampai dengan `Limit` yang ditentukan. Jika ada lebih banyak item inventarus yang tersedia, hasilnya adalah pemberian nomor halaman. Setelah tugas selesai, Anda dapat menggunakan operasi Mendeskripsikan Tugas (GET JobID) untuk mendapatkan penanda yang Anda gunakan dalam permintaan Mulai Tugas

berikut. Penanda menunjukkan titik awal untuk mengambil sekumpulan item inventaris berikutnya. Anda dapat menelusuri seluruh inventaris dengan berulang kali membuat permintaan Mulai Tugas dengan penanda dari output Jelaskan Tugas sebelumnya. Anda melakukannya sampai Anda mendapatkan penanda dari Jelaskan Tugas yang mengembalikan nol, menunjukkan tidak ada item lainnya yang tersedia.

Anda dapat menggunakan parameter `Limit` bersama-sama dengan parameter rentang tanggal.

Pengambilan Arsip Berkisar

Anda dapat memulai pengambilan arsip untuk seluruh arsip atau rentang arsip. Dalam hal pengambilan arsip berkisar, Anda menentukan rentang byte untuk dikembalikan atau seluruh arsip. Rentang yang ditentukan harus megabyte (MB) selaras. Dengan kata lain, rentang awal nilai harus habis dibagi 1 MB dan rentang nilai akhir ditambah 1 harus habis dibagi 1 MB atau sama dengan akhir arsip. Jika pengambilan arsip berkisar bukan megabyte selaras, operasi ini mengembalikan respons `400`. Selain itu, untuk memastikan Anda mendapatkan nilai checksum untuk data yang Anda unduh menggunakan Dapatkan Output Tugas (Mendapatkan Output Tugas (GET output)), rentang harus berupa tree-hash selaras. Untuk informasi selengkapnya tentang rentang tree-hash selaras, lihat Menerima Checksum Saat Mengunduh Data.

Tingkat Dipercepat, Standar, dan Massal.

Saat memulai pekerjaan pengambilan arsip, Anda dapat menentukan salah satu opsi berikut di `Tier` bidang badan permintaan:

- **`Expedited`**— Dipercepat memungkinkan Anda untuk dengan cepat mengakses data Anda ketika permintaan mendesak sesekali untuk memulihkan arsip diperlukan. Untuk semua, kecuali arsip terbesar (250 MB+), data yang diakses menggunakan tingkat Dipercepat biasanya disediakan dalam waktu 1–5 menit.
- **`Standard`** – Standar memungkinkan Anda mengakses salah satu arsip Anda dalam beberapa jam. Data yang diakses menggunakan tingkat Standar biasanya tersedia dalam 3–5 jam. Ini adalah opsi default untuk permintaan tugas yang tidak menentukan opsi tingkat.
- **`Bulk`**- Massal adalah tingkat biaya terendah untuk Amazon Glacier, memungkinkan Anda untuk mengambil sejumlah besar, bahkan petabyte, data dengan harga murah dalam sehari. Data yang diakses menggunakan tingkat Massal biasanya tersedia dalam 5–12 jam.

Untuk informasi selengkapnya tentang pengambilan yang dipercepat dan massal, lihat Mengambil Arsip Amazon Glacier.

Permintaan

Untuk memulai tugas, Anda menggunakan metode `POST` HTTP dan mencakup permintaan untuk subsumber daya `jobs` vault. Anda menentukan detail permintaan tugas dalam dokumen JSON dari permintaan Anda. Tipe tugas ditentukan dengan bidang `Type`. Secara opsional, Anda dapat menentukan `SNSTopic` bidang untuk menunjukkan topik Amazon SNS tempat Amazon Glacier dapat memposting pemberitahuan setelah menyelesaikan pekerjaan.

> **Note**
>
> Untuk mengirim notifikasi ke Amazon SNS, Anda harus membuat topik sendiri jika belum ada. Amazon Glacier tidak membuat topik untuk Anda. Topik harus memiliki izin untuk menerima publikasi dari lemari besi Amazon Glacier. Amazon Glacier tidak memverifikasi apakah brankas memiliki izin untuk mempublikasikan ke topik. Jika izin tidak dikonfigurasi dengan tepat, Anda mungkin tidak menerima notifikasi bahkan setelah tugas selesai.

Sintaksis

Berikut ini adalah sintaks permintaan untuk memulai tugas.

```
POST /AccountId/vaults/VaultName/jobs HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: Date
Authorization: SignatureValue
x-amz-glacier-version: 2012-06-01

{
   "jobParameters": {
      "ArchiveId": "string",
      "Description": "string",
      "Format": "string",
      "InventoryRetrievalParameters": {
         "EndDate": "string",
         "Limit": "string",
         "Marker": "string",
         "StartDate": "string"
      },
      "OutputLocation": {
         "S3": {
            "AccessControlList": [
```

```
            {
               "Grantee": {
                  "DisplayName": "string",
                  "EmailAddress": "string",
                  "ID": "string",
                  "Type": "string",
                  "URI": "string"
               },
               "Permission": "string"
            }
         ],
         "BucketName": "string",
         "CannedACL": "string",
         "Encryption": {
            "EncryptionType": "string",
            "KMSContext": "string",
            "KMSKeyId": "string"
         },
         "Prefix": "string",
         "StorageClass": "string",
         "Tagging": {
            "string" : "string"
         },
         "UserMetadata": {
            "string" : "string"
         }
      }
   },
   "RetrievalByteRange": "string",
   "SelectParameters": {
      "Expression": "string",
      "ExpressionType": "string",
      "InputSerialization": {
         "csv": {
            "Comments": "string",
            "FieldDelimiter": "string",
            "FileHeaderInfo": "string",
            "QuoteCharacter": "string",
            "QuoteEscapeCharacter": "string",
            "RecordDelimiter": "string"
         }
      },
      "OutputSerialization": {
         "csv": {
```

```
                "FieldDelimiter": "string",
                "QuoteCharacter": "string",
                "QuoteEscapeCharacter": "string",
                "QuoteFields": "string",
                "RecordDelimiter": "string"
            }
         }
      },
      "SNSTopic": "string",
      "Tier": "string",
      "Type": "string"
   }
}
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID akun yang memiliki brankas. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menggunakan ID akun, jangan sertakan tanda hubung ('-') di ID.

Isi Permintaan

Permintaan menerima data berikut dalam format JSON dalam isi permintaan.

jobParameters

Menyediakan opsi untuk menentukan informasi tugas.

Tipe: Objek jobParameters

Wajib: Ya

Respons

Amazon Glacier menciptakan pekerjaan itu. Sebagai respons, S3 Glacier mengembalikan URI tugas.

Sintaksis

```
HTTP/1.1 202 Accepted
x-amzn-RequestId: x-amzn-RequestId
Date: Date
Location: location
x-amz-job-id: jobId
x-amz-job-output-path: jobOutputPath
```

Header Respons

Header	Deskripsi
`Location`	Jalur URI relatif dari tugas. Anda dapat menggunakan jalur URI ini untuk menemukan status tugas. Untuk informasi selengkapnya, lihat Mendeskripsikan Tugas (GET JobID). Tipe: String Default: Tidak ada
`x-amz-job-id`	ID tugas. Nilai ini juga disertakan sebagai bagian dari header `Location`. Tipe: String Default: Tidak Ada
`x-amz-job-output-path`	Jalur ke lokasi tempat hasil pilih disimpan. Tipe: String Default: Tidak ada

Isi Respons

Operasi ini tidak mengembalikan isi respons.

Kesalahan

Operasi ini mencakup kesalahan atau kesalahan berikut, di samping kemungkinan kesalahan yang umum terjadi pada semua operasi Amazon Glacier. Untuk informasi tentang kesalahan Amazon Glacier dan daftar kode kesalahan, lihat. Respons Kesalahan

Kode	Deskripsi	Kode Status HTTP	Tipe
`InsufficientCapacityException`	Dikembalikan jika kapasitas tidak memadai untuk memproses permintaan yang dipercepat ini. Kesalahan ini hanya berlaku untuk pengambilan yang dipercepat dan bukan untuk pengambilan standar atau massal.	`503 Service Unavailable`	Server

Contoh

Contoh Permintaan: Memulai tugas pengambilan arsip

```
POST /-/vaults/examplevault/jobs HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
x-amz-glacier-version: 2012-06-01
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2

{
  "Type": "archive-retrieval",
  "ArchiveId": "NkbByEejwEggmBz2fTHgJrg0XBoDfjP4q6iu87-
TjhqG6eGoOY9Z8i1_AUyUsuhPAdTqLHy8pTl5nfCFJmDl2yEZONi5L26Omw12vcs01MNGntHEQL8MBfGlqrEXAMPLEArchi
  "Description": "My archive description",
  "SNSTopic": "arn:aws:sns:us-west-2:111111111111:Glacier-ArchiveRetrieval-topic-
Example",
  "Tier" : "Bulk"
}
```

Berikut adalah contoh isi permintaan yang menentukan rentang arsip yang diambil menggunakan bidang `RetrievalByteRange`.

```
{
  "Type": "archive-retrieval",
  "ArchiveId": "NkbByEejwEggmBz2fTHgJrg0XBoDfjP4q6iu87-
TjhqG6eGoOY9Z8i1_AUyUsuhPAdTqLHy8pTl5nfCFJmDl2yEZONi5L26Omw12vcs01MNGntHEQL8MBfGlqrEXAMPLEArchi
  "Description": "My archive description",
  "RetrievalByteRange": "2097152-4194303",
  "SNSTopic": "arn:aws:sns:us-west-2:111111111111:Glacier-ArchiveRetrieval-topic-
Example",
  "Tier" : "Bulk"
}
```

Contoh Respons

```
HTTP/1.1 202 Accepted
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:00:00 GMT
Location: /111122223333/vaults/examplevault/jobs/HkF9p6o7yjhFx-
K3CGl6fuSm6VzW9T7esGQfco8nUXVYwS0jlb5gq1JZ55yHgt5vP54ZShjoQzQVVh7vEXAMPLEjobID
x-amz-job-id: HkF9p6o7yjhFx-
K3CGl6fuSm6VzW9T7esGQfco8nUXVYwS0jlb5gq1JZ55yHgt5vP54ZShjoQzQVVh7vEXAMPLEjobID
```

Contoh Permintaan: Memulai tugas pengambilan inventaris

Permintaan berikut memulai tugas pengambilan inventaris untuk mendapatkan daftar arsip dari vault `examplevault`. `Format` yang ditetapkan ke `CSV` dalam isi permintaan menunjukkan inventaris dikembalikan dalam format CSV.

```
POST /-/vaults/examplevault/jobs HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
Content-Type: application/x-www-form-urlencoded
x-amz-glacier-version: 2012-06-01
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2

{
  "Type": "inventory-retrieval",
  "Description": "My inventory job",
```

```
  "Format": "CSV",
  "SNSTopic": "arn:aws:sns:us-west-2:111111111111:Glacier-InventoryRetrieval-topic-
Example"
}
```

Contoh Respons

```
HTTP/1.1 202 Accepted
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:00:00 GMT
Location: /111122223333/vaults/examplevault/jobs/HkF9p6o7yjhFx-
K3CGl6fuSm6VzW9T7esGQfco8nUXVYwS0jlb5gq1JZ55yHgt5vP54ZShjoQzQVVh7vEXAMPLEjobID
x-amz-job-id: HkF9p6o7yjhFx-
K3CGl6fuSm6VzW9T7esGQfco8nUXVYwS0jlb5gq1JZ55yHgt5vP54ZShjoQzQVVh7vEXAMPLEjobID
```

Contoh Permintaan: Memulai tugas pengambilan inventaris menggunakan penyaringan tanggal dengan batas yang ditetapkan, dan permintaan berikutnya untuk mengambil halaman item inventaris berikutnya.

Permintaan berikut memulai tugas pengambilan inventaris vault menggunakan penyaringan tanggal dan menetapkan batas.

```
{
    "ArchiveId": null,
    "Description": null,
    "Format": "CSV",
    "RetrievalByteRange": null,
    "SNSTopic": null,
    "Type": "inventory-retrieval",
    "InventoryRetrievalParameters": {
        "StartDate": "2013-12-04T21:25:42Z",
        "EndDate": "2013-12-05T21:25:42Z",
        "Limit" : "10000"
    },
}
```

Permintaan berikut adalah contoh dari permintaan berikutnya untuk mengambil halaman item inventaris berikutnya menggunakan penanda yang diperoleh dari Mendeskripsikan Tugas (GET JobID).

```
{
```

```
    "ArchiveId": null,
    "Description": null,
    "Format": "CSV",
    "RetrievalByteRange": null,
    "SNSTopic": null,
    "Type": "inventory-retrieval",
    "InventoryRetrievalParameters": {
        "StartDate": "2013-12-04T21:25:42Z",
        "EndDate": "2013-12-05T21:25:42Z",
        "Limit": "10000",
        "Marker":
 "vyS0t2jHQe5qbcDggIeD50chS1SXwYMrkVKo0KHiTUjEYxBGCqRLKaiySzdN7QXGVVV5XZpNVG67pCZ_uykQXFMLaxOSu
    },
}
```

Contoh Respons

```
HTTP/1.1 202 Accepted
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:00:00 GMT
Location: /111122223333/vaults/examplevault/jobs/HkF9p6o7yjhFx-
K3CGl6fuSm6VzW9T7esGQfco8nUXVYwS0jlb5gq1JZ55yHgt5vP54ZShjoQzQVVh7vEXAMPLEjobID
x-amz-job-id: HkF9p6o7yjhFx-
K3CGl6fuSm6VzW9T7esGQfco8nUXVYwS0jlb5gq1JZ55yHgt5vP54ZShjoQzQVVh7vEXAMPLEjobID
x-amz-job-output-path: test/HkF9p6o7yjhFx-
K3CGl6fuSm6VzW9T7esGQfco8nUXVYwS0jlb5gq1JZ55yHgt5vP54ZShjoQzQVVh7vEXAMPLEjobID/
```

Bagian Terkait

- Mendeskripsikan Tugas (GET JobID)
- Mendapatkan Output Tugas (GET output)
- Identity and Access Management untuk Amazon Glacier

Mencantumkan Tugas (GET jobs)

Deskripsi

Operasi ini mencantumkan daftar tugas untuk vault, termasuk tugas yang sedang berlangsung dan tugas yang baru saja selesai.

> **Note**
>
> Amazon Glacier (Amazon Glacier) mempertahankan pekerjaan yang baru saja selesai untuk jangka waktu sebelum menghapusnya; Namun, akhirnya menghapus pekerjaan yang sudah selesai. Output dari tugas yang diselesaikan dapat diambil. Mempertahankan tugas yang selesai untuk jangka waktu tertentu setelah selesai memungkinkan Anda mendapatkan output tugas jika Anda melewatkan notifikasi penyelesaian tugas, atau upaya pertama Anda untuk mengunduhnya gagal. Misalnya, Anda memulai tugas pengambilan arsip untuk mengunduh arsip. Setelah tugas selesai, Anda mulai mengunduh arsip, tetapi mengalami kesalahan jaringan. Dalam skenario ini, Anda dapat mencoba lagi dan mengunduh arsip sewaktu tugas ada.

Operasi `List Jobs` mendukung pemberian nomor halaman. Anda harus selalu memeriksa bidang `Marker` respons. Jika tidak ada tugas lainnya yang dicantumkan, bidang `Marker` diatur ke `null`. Jika ada tugas lainnya yang dicantumkan, bidang `Marker` diatur ke nilai non-nol, yang dapat Anda gunakan untuk melanjutkan pemberian nomor halaman daftar. Untuk mengembalikan daftar tugas yang dimulai pada tugas tertentu, atur parameter permintaan `marker` ke nilai `Marker` untuk tugas yang Anda peroleh dari permintaan `List Jobs` sebelumnya.

Anda juga dapat menetapkan batas maksimum untuk jumlah tugas yang dikembalikan dalam respons dengan menentukan parameter `limit` dalam permintaan. Batas default adalah 50. Jumlah tugas yang dikembalikan mungkin kurang dari batas, tetapi jumlah tugas yang dikembalikan tidak pernah melebihi batas.

Selain itu, Anda dapat memfilter daftar tugas yang dikembalikan dengan menentukan parameter `statuscode` opsional atau parameter `completed`, atau pun keduanya. Dengan menggunakan parameter `statuscode`, Anda dapat menentukan untuk hanya mengembalikan tugas yang cocok dengan status `InProgress`, `Succeeded`, atau `Failed`. Dengan menggunakan parameter `completed`, Anda dapat menentukan untuk hanya mengembalikan tugas yang sudah selesai (`true`) atau tugas yang belum selesai (`false`).

Permintaan

Sintaksis

Untuk mengembalikan daftar tugas dari semua tipe, kirim permintaan `GET` ke URI dari subsumber daya `jobs` vault.

```
GET /AccountId/vaults/VaultName/jobs HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: Date
Authorization: SignatureValue
x-amz-glacier-version: 2012-06-01
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID akun yang memiliki brankas. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menggunakan ID akun, jangan sertakan tanda hubung ('-') di ID.

Parameter Permintaan

Nama	Deskripsi	Wajib
`completed`	Status tugas yang dikembalikan. Anda dapat menentukan `true` atau `false`. Tipe: Boolean Batasan: Tidak Ada	Tidak
`limit`	Jumlah maksimum tugas yang dikembalikan. Batas default adalah 50. Jumlah tugas yang dikembalikan mungkin kurang dari batas yang ditentukan, tetapi jumlah tugas yang dikembali kan tidak pernah melebihi batas. Tipe: String Batasan: Nilai bilangan bulat minimum 1. Nilai bilangan bulat maksimum sebesar 50.	Tidak
`marker`		Tidak

Nama	Deskripsi	Wajib
	String buram yang digunakan untuk pemberian nomor halaman yang menentukan tugas tempat cantuman bagian harus dimulai. Anda mendapatkan nilai `marker` dari respons `List Jobs` sebelumnya. Anda harus menyertakan `marker` jika Anda melanjutkan pemberian nomor halaman dari hasil yang dimulai dalam permintaan `List Jobs` sebelumnya. Tipe: String Batasan: Tidak Ada	
`statuscode`	Tipe status tugas yang dikembalikan. Tipe: String Batasan: Salah satu nilai `InProgress` , `Succeeded` , atau `Failed`.	Tidak

Header Permintaan

Operasi ini hanya menggunakan header respons yang umum untuk sebagian besar respons. Untuk informasi tentang header permintaan umum, lihat Header Respons Umum.

Isi Permintaan

Operasi ini tidak memiliki isi permintaan.

Respons

Sintaksis

```
HTTP/1.1 200 OK
x-amzn-RequestId: x-amzn-RequestId
Date: Date
Location: Location
Content-Type: application/json
Content-Length: Length

{
```

```
    "JobList": [
        {
            "Action": "string",
            "ArchiveId": "string",
            "ArchiveSHA256TreeHash": "string",
            "ArchiveSizeInBytes": number,
            "Completed": boolean,
            "CompletionDate": "string",
            "CreationDate": "string",
            "InventoryRetrievalParameters": {
                "EndDate": "string",
                "Format": "string",
                "Limit": "string",
                "Marker": "string",
                "StartDate": "string"
            },
            "InventorySizeInBytes": number,
            "JobDescription": "string",
            "JobId": "string",
            "JobOutputPath": "string",
            "OutputLocation": {
                "S3": {
                    "AccessControlList": [
                        {
                            "Grantee": {
                                "DisplayName": "string",
                                "EmailAddress": "string",
                                "ID": "string",
                                "Type": "string",
                                "URI": "string"
                            },
                            "Permission": "string"
                        }
                    ],
                    "BucketName": "string",
                    "CannedACL": "string",
                    "Encryption": {
                        "EncryptionType": "string",
                        "KMSContext": "string",
                        "KMSKeyId": "string"
                    },
                    "Prefix": "string",
                    "StorageClass": "string",
                    "Tagging": {
```

```
                        "string": "string"
                    },
                    "UserMetadata": {
                        "string": "string"
                    }
                }
            },
            "RetrievalByteRange": "string",
            "SelectParameters": {
                "Expression": "string",
                "ExpressionType": "string",
                "InputSerialization": {
                    "csv": {
                        "Comments": "string",
                        "FieldDelimiter": "string",
                        "FileHeaderInfo": "string",
                        "QuoteCharacter": "string",
                        "QuoteEscapeCharacter": "string",
                        "RecordDelimiter": "string"
                    }
                },
                "OutputSerialization": {
                    "csv": {
                        "FieldDelimiter": "string",
                        "QuoteCharacter": "string",
                        "QuoteEscapeCharacter": "string",
                        "QuoteFields": "string",
                        "RecordDelimiter": "string"
                    }
                }
            },
            "SHA256TreeHash": "string",
            "SNSTopic": "string",
            "StatusCode": "string",
            "StatusMessage": "string",
            "Tier": "string",
            "VaultARN": "string"
        }
    ],
    "Marker": "string"
}
```

Header Respons

Operasi ini hanya menggunakan header respons yang umum untuk sebagian besar respons. Untuk informasi tentang header permintaan umum, lihat Header Respons Umum.

Isi Respons

Isi respons berisi bidang JSON berikut.

JobList

Daftar objek tugas. Setiap objek tugas berisi metadata yang menggambarkan tugas.

Tipe: Array objek GlacierJobDescription

Penanda

String buram yang mewakili tempat untuk melanjutkan pemberian nomor halaman hasil. Anda menggunakan nilai `marker` dalam permintaan `List Jobs` baru untuk mendapatkan lebih banyak tugas dalam daftar. Jika tidak ada tugas lainnya, nilai ini adalah `null`.

Tipe: String

Kesalahan

Untuk informasi tentang pengecualian Amazon Glacier dan pesan kesalahan, lihat. Respons Kesalahan

Contoh

Contoh berikut menunjukkan cara mengembalikan informasi tentang tugas vault. Contoh pertama mengembalikan daftar dua tugas, dan contoh kedua mengembalikan subset tugas.

Contoh: Mengembalikan Semua Tugas

Contoh Permintaan

Permintaan GET berikut mengembalikan tugas untuk vault.

```
GET /-/vaults/examplevault/jobs  HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
```

```
x-amz-Date: 20170210T120000Z
x-amz-glacier-version: 2012-06-01
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
```

Contoh Respons

Respons berikut menyertakan tugas pengambilan arsip dan tugas pengambilan inventaris yang berisi penanda yang digunakan untuk melanjutkan pemberian nomor halaman dari pengambilan inventaris vault. Respons juga menunjukkan bidang `Marker` yang diatur ke`null`, yang menunjukkan tidak ada lagi tugasyang dicantumkan.

```
HTTP/1.1 200 OK
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:00:00 GMT
Content-Type: application/json
Content-Length: 1444

{
  "JobList": [
    {
      "Action": "ArchiveRetrieval",
      "ArchiveId": "BDfaUQul0dVzYwAMr8YSa_6_8abbhZq-
i1oT69g8ByClfJyBgAGBkWl2QbF5os851P7Y7KdZDOHWJIn4rh1ZHaOYD3MgFhK_g0oDPesW34uHQoVGwoIqubf6BgUEfQm
      "ArchiveSizeInBytes": 1048576,
      "ArchiveSHA256TreeHash":
 "25499381569ab2f85e1fd0eb93c5406a178ab77c5933056eb5d6e7d4adda609b",
      "Completed": true,
      "CompletionDate": "2012-05-01T00:00:09.304Z",
      "CreationDate": "2012-05-01T00:00:06.663Z",
      "InventorySizeInBytes": null,
      "JobDescription": null,
      "JobId": "hDe9t9DTHXqFw8sBGpLQQOmIM0-
JrGtu1O_YFKLnzQ64548qJc667BRWTwBLZC76Ygy1jHYruqXkdcAhRsh0hYv4eVRU",
      "RetrievalByteRange": "0-1048575",
      "SHA256TreeHash":
 "25499381569ab2f85e1fd0eb93c5406a178ab77c5933056eb5d6e7d4adda609b",
      "SNSTopic": null,
      "StatusCode": "Succeeded",
      "StatusMessage": "Succeeded",
      "Tier": "Bulk",
      "VaultARN": "arn:aws:glacier:us-west-2:012345678901:vaults/examplevault"
```

```
    },
    {
      "Action": "InventoryRetrieval",
      "ArchiveId": null,
      "ArchiveSizeInBytes": null,
      "ArchiveSHA256TreeHash": null,
      "Completed": true,
      "CompletionDate": "2013-05-11T00:25:18.831Z",
      "CreationDate": "2013-05-11T00:25:14.981Z",
      "InventorySizeInBytes": 1988,
      "JobDescription": null,
      "JobId":
 "2cvVOnBL36btzyP3pobwIceiaJebM1bx9vZOOUtmNAr0KaVZ4WkWgVjiPldJ73VU7imlm0pnZriBVBebnqaAcirZq_C5"
      "RetrievalByteRange": null,
      "SHA256TreeHash": null,
      "SNSTopic": null,
      "StatusCode": "Succeeded",
      "StatusMessage": "Succeeded",
      "VaultARN": "arn:aws:glacier:us-west-2:012345678901:vaults/examplevault"
      "InventoryRetrievalParameters": {
          "StartDate": "2013-11-12T13:43:12Z",
          "EndDate": "2013-11-20T08:12:45Z",
          "Limit": "120000",
          "Format": "JSON",
          "Marker":
 "vyS0t2jHQe5qbcDggIeD50chS1SXwYMrkVKo0KHiTUjEYxBGCqRLKaiySzdN7QXGVVV5XZpNVG67pCZ_uykQXFMLaxOSu
    }
  ],
  "Marker": null
}
```

Contoh: Mengembalikan Daftar Sebagian Tugas

Contoh Permintaan

Permintaan GET berikut mengembalikan tugas yang ditentukan oleh parameter `marker`. Mengatur parameter `limit` sebagai 2 menetapkan bahwa hingga dua tugas dikembalikan.

```
GET /-/vaults/examplevault/jobs?marker=HkF9p6o7yjhFx-
K3CGl6fuSm6VzW9T7esGQfco8nUXVYwS0jlb5gq1JZ55yHgt5vP54ZShjoQzQVVh7vEXAMPLEjobID&limit=2
 HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
```

```
x-amz-glacier-version: 2012-06-01
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
```

Contoh Respons

Respons berikut menunjukkan dua tugas yang dikembalikan dan bidang `Marker` yang diatur ke nilai non-nol yang dapat digunakan untuk melanjutkan pemberian nomor halaman daftar tugas.

```
HTTP/1.1 200 OK
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:00:00 GMT
Content-Type: application/json
Content-Length: 1744

{
  "JobList": [
    {
      "Action": "ArchiveRetrieval",
      "ArchiveId": "58-3KpZfcMPUznvMZNPaKyJx9wODCsWTnqcjtx2CjKZ6b-
XgxEuA8yvZOYTPQfd7gWR4GRm2XR08gcnWbLV4VPV_kDWtZJKi0TFhKKVPzwrZnA4-
FXuIBfViYUIVveeiBE51FO4bvg",
      "ArchiveSizeInBytes": 8388608,
      "ArchiveSHA256TreeHash":
 "106086b256ddf0fedf3d9e72f461d5983a2566247ebe7e1949246bc61359b4f4",
      "Completed": true,
      "CompletionDate": "2012-05-01T00:25:20.043Z",
      "CreationDate": "2012-05-01T00:25:16.344Z",
      "InventorySizeInBytes": null,
      "JobDescription": "aaabbbccc",
      "JobId": "s4MvaNHIh6mOa1f8iY4ioG2921SDPihXxh3Kv0FBX-
JbNPctpRvE4c2_BifuhdGLqEhGBNGeB6Ub-JMunR9JoVa8y1hQ",
      "RetrievalByteRange": "0-8388607",
      "SHA256TreeHash":
 "106086b256ddf0fedf3d9e72f461d5983a2566247ebe7e1949246bc61359b4f4",
      "SNSTopic": null,
      "StatusCode": "Succeeded",
      "StatusMessage": "Succeeded",
      "Tier": "Bulk",
      "VaultARN": "arn:aws:glacier:us-west-2:012345678901:vaults/examplevault"
    },
    {
```

```
      "Action": "ArchiveRetrieval",
      "ArchiveId": "2NVGpf83U6qB9M2u-
Ihh61yoFLRDEoh7YLZWKBn80A2i1xG8uieBwGjAr4RkzOHA0E07ZjtI267R03Z-6Hxd8pyGQkBdciCSH1-
Lw63Kx9qKpZbPCdU0uTW_WAdwF6lR6w8iSyKdvw",
      "ArchiveSizeInBytes": 1048576,
      "ArchiveSHA256TreeHash":
 "3d2ae052b2978727e0c51c0a5e32961c6a56650d1f2e4ceccab6472a5ed4a0",
      "Completed": true,
      "CompletionDate": "2012-05-01T16:59:48.444Z",
      "CreationDate": "2012-05-01T16:59:42.977Z",
      "InventorySizeInBytes": null,
      "JobDescription": "aaabbbccc",
      "JobId":
 "CQ_tf6fOR4jrJCL61Mfk6VM03oY8lmnWK93KK4gLig1UPAbZiN3UV4G_5nq4AfmJHQ_dOMLOX5k8ItFv0wCPN0oaz5dG"
      "RetrievalByteRange": "0-1048575",
      "SHA256TreeHash":
 "3d2ae052b2978727e0c51c0a5e32961c6a56650d1f2e4ceccab6472a5ed4a0",
      "SNSTopic": null,
      "StatusCode": "Succeeded",
      "StatusMessage": "Succeeded",
      "Tier": "Standard",
      "VaultARN": "arn:aws:glacier:us-west-2:012345678901:vaults/examplevault"
    }
  ],
  "Marker":
 "CQ_tf6fOR4jrJCL61Mfk6VM03oY8lmnWK93KK4gLig1UPAbZiN3UV4G_5nq4AfmJHQ_dOMLOX5k8ItFv0wCPN0oaz5dG"
}
```

Bagian Terkait

- Mendeskripsikan Tugas (GET JobID)
- Identity and Access Management untuk Amazon Glacier

Tipe Data yang Digunakan dalam Operasi Tugas

Berikut ini adalah tipe data yang digunakan dengan operasi pekerjaan di Amazon Glacier.

Topik

- CSVInput
- CSVOutput

- Enkripsi
- GlacierJobDescription
- Pemberian Izin
- Penerima izin
- InputSerialization
- InventoryRetrievalJobInput
- jobParameters
- OutputLocation
- OutputSerialization
- S3Location
- SelectParameters

CSVInput

Berisi informasi tentang file nilai yang dipisahkan koma (CSV).

Daftar Isi

Komentar

Karakter tunggal yang digunakan untuk menunjukkan baris harus diabaikan ketika karakter muncul di awal baris tersebut.

Tipe: String

Wajib: tidak

FieldDelimiter

Karakter tunggal yang digunakan untuk memisahkan bidang individu satu sama lain dalam catatan. Karakternya harus berupa `\n`, `\r`, atau karakter ASCII dalam rentang 32–126. Default adalah koma (`,`).

Tipe: String

Default: ,

Wajib: tidak

FileHeaderInfo

Nilai yang menggambarkan hal yang harus dilakukan dengan baris pertama input.

Tipe: String

Nilai yang Valid: `Use` | `Ignore` | `None`

Wajib: tidak

QuoteCharacter

Karakter tunggal yang digunakan sebagai karakter escape dengan pembatas bidang adalah bagian dari nilai.

Tipe: String

Wajib: tidak

QuoteEscapeCharacter

Karakter tunggal yang digunakan untuk membebaskan karakter tanda kutip dalam nilai yang sudah dibebaskan.

Tipe: String

Wajib: tidak

RecordDelimiter

Karakter tunggal yang digunakan untuk memisahkan catatan individu satu sama lain.

Tipe: String

Wajib: tidak

Info Selengkapnya

- Memulai Tugas (POST jobs)

CSVOutput

Berisi informasi tentang format nilai yang dipisahkan koma (CSV) yang disimpan dalam hasil tugas.

Daftar Isi

FieldDelimiter

Karakter tunggal yang digunakan untuk memisahkan bidang individu satu sama lain dalam catatan.

Tipe: String

Wajib: tidak

QuoteCharacter

Karakter tunggal yang digunakan sebagai karakter escape dengan pembatas bidang adalah bagian dari nilai.

Tipe: String

Wajib: tidak

QuoteEscapeCharacter

Karakter tunggal yang digunakan untuk membebaskan karakter tanda kutip dalam nilai yang sudah dibebaskan.

Tipe: String

Wajib: tidak

QuoteFields

Nilai yang menunjukkan apakah semua bidang output harus tercantum dalam tanda kutip.

Nilai yang Valid: `ALWAYS` | `ASNEEDED`

Tipe: String

Wajib: tidak

RecordDelimiter

Karakter tunggal yang digunakan untuk memisahkan catatan individu satu sama lain.

Tipe: String

Wajib: tidak

Info Selengkapnya

- Memulai Tugas (POST jobs)

Enkripsi

Berisi informasi tentang enkripsi yang digunakan untuk menyimpan hasil tugas di Amazon S3.

Daftar Isi

Enkripsi

Algoritme enkripsi sisi server yang digunakan saat menyimpan hasil tugas di Amazon S3. Defaultnya adalah tidak ada enkripsi.

Tipe: String

Nilai yang Valid: `aws:kms` | `AES256`

Wajib: tidak

KMSContext

Tidak wajib. Jika tipe enkripsi adalah `aws:kms`, Anda dapat menggunakan nilai ini untuk menentukan konteks enkripsi untuk hasil tugas.

Tipe: String

Wajib: tidak

KMSKeyId

ID kunci AWS Key Management Service (AWS KMS) yang digunakan untuk enkripsi objek.

Tipe: String

Wajib: tidak

Info Selengkapnya

- Memulai Tugas (POST jobs)

GlacierJobDescription

Berisi deskripsi pekerjaan Amazon Glacier (Amazon Glacier).

Daftar Isi

Tindakan

Jenis tugas. Baik `ArchiveRetrieval`, `InventoryRetrieval`, maupun `Select`.

Tipe: String

ArchiveId

ID arsipyang diminta untuk memilih atau mengarsipkan tugas pengambilan. Jika tidak, bidang ini adalah `null`.

Tipe: String

Arsip SHA256 TreeHash

Hash SHA256 pohon dari seluruh arsip untuk pengambilan arsip. Untuk tugas pengambilan inventaris, bidang ini adalah `null`.

Tipe: String

ArchiveSizeInBytes

Untuk tugas `ArchiveRetrieval`, ini berukuran dalam byte dari arsip yang diminta untuk diunduh. Untuk tugas `InventoryRetrieval`, nilai adalah `null`.

Tipe: Angka

Selesai

`true` jika tugas selesai; `false` jika tidak.

Tipe: Boolean

CompletionDate

Tanggal ketika tugas selesai.

Waktu Universal Terkoordinasi (UTC) saat permintaan tugas selesai. Sewaktu tugas sedang berlangsung, nilai adalah nol.

Tipe: Representasi string dalam format tanggal ISO 8601, misalnya `2013-03-20T17:03:43.221Z`.

CreationDate

Tanggal Waktu Universal Terkoordinasi (UTC) saat tugas dimulai.

Tipe: Representasi string dalam format tanggal ISO 8601, misalnya `2013-03-20T17:03:43.221Z`.

InventoryRetrievalParameters

Parameter input yang digunakan untuk pengambilan berbagai inventaris.

Tipe: Objek InventoryRetrievalJobInput

InventorySizeInBytes

Untuk tugas `InventoryRetrieval`, ini berukuran dalam byte dari inventaris yang diminta untuk diunduh. Untuk tugas `ArchiveRetrieval` atau `Select`, nilai adalah `null`.

Tipe: Angka

JobDescription

Deskripsi tugas yang Anda berikan ketika Anda memulai tugas.

Tipe: String

JobId

ID yang mengidentifikasi pekerjaan di Amazon Glacier.

Tipe: String

JobOutputPath

Berisi lokasi output tugas.

Tipe: String

OutputLocation

Objek yang berisi informasi tentang lokasi tempat hasil pilih tugas dan kesalahan disimpan.

Tipe: Objek OutputLocation

RetrievalByteRange

Rentang byte yang diambil untuk pekerjaan pengambilan arsip dalam bentuk "*StartByteValue*-*EndByteValue*." Jika tidak ada rentang yang ditentukan dalam pengambilan arsip, maka seluruh arsip diambil dan StartByteValuesama dengan 0 dan EndByteValuesama dengan ukuran arsip dikurangi 1. Untuk tugas pengambilan inventaris, bidang ini adalah `null`.

Tipe: String

SelectParameters

Objek yang berisi informasi tentang parameter yang digunakan untuk select.

Tipe: Objek SelectParameters

SHA256TreeHash

Nilai hash SHA256 pohon untuk rentang arsip yang diminta. Jika permintaan Memulai Tugas (POST jobs) untuk arsip menentukan rentang tree-hash selaras, bidang ini mengembalikan nilai. Untuk informasi selengkapnya tentang penyelarasan tree-hash untuk pengambilan rentang arsip, lihat Menerima Checksum Saat Mengunduh Data.

Untuk kasus tertentu ketika seluruh arsip diambil, nilai ini sama dengan nilai `ArchiveSHA256TreeHash`.

Bidang ini adalah `null` dalam situasi berikut:

- Tugas pengambilan arsip yang menentukan rentang yang bukan tree-hash selaras.
- Tugas arsip yang menentukan rentang yang sama dengan status seluruh arsip dan tugas adalah `InProgress`.
- Tugas inventaris.
- Pilih tugas.

Tipe: String

SNSTopic

Amazon Resource Name (ARN) yang mewakili topik Amazon SNS tempat notifikasi penyelesaian atau kegagalan tugas dikirim, jika notifikasi dikonfigurasi dalam inisiasi tugas (Memulai Tugas (POST jobs)).

Tipe: String

StatusCode

Kode yang menunjukkan status tugas.

Nilai yang Valid: `InProgress` | `Succeeded` | `Failed`

Tipe: String

StatusMessage

Pesan status tugas.

Tipe: String

Tingkat

Tingkat akses data yang digunakan untuk pengambilan pilih atau arsip.

Nilai yang Valid: `Expedited` | `Standard` | `Bulk`

Tipe: String

Vaultarn

ARN dari vault yang tugasnya adalah subsumber daya.

Tipe: String

Info Selengkapnya

- Memulai Tugas (POST jobs)

Pemberian Izin

Berisi informasi tentang pemberian izin.

Daftar Isi

Penerima hibah

Penerima izin.

Tipe: Objek Penerima izin

Wajib: tidak

Izin

Izin yang diberikan kepada penerima izin.

Tipe: String

Nilai yang Valid: `FULL_CONTROL` | `WRITE` | `WRITE_ACP` | `READ` | `READ_ACP`

Wajib: tidak

Info Selengkapnya

- Memulai Tugas (POST jobs)

Penerima izin

Berisi informasi tentang penerima izin.

Daftar Isi

DisplayName

Nama layar penerima izin.

Tipe: String

Wajib: tidak

EmailAddress

Alamat email penerima izin.

Tipe: String

Wajib: tidak

ID

ID pengguna kanonik penerima izin.

Tipe: String

Wajib: tidak

Jenis

Tipe penerima izin.

Tipe: String

Nilai yang Valid: `AmazonCustomerByEmail` | `CanonicalUser` | `Group`

Wajib: tidak

URI

URI dari grup penerima izin.

Tipe: String

Wajib: tidak

Info Selengkapnya

- Memulai Tugas (POST jobs)

InputSerialization

Menjelaskan cara arsip diserialkan.

Daftar Isi

CSV

Objek yang menggambarkan serialisasi objek yang dikodekan CSV.

Tipe: Objek CSVInput

Wajib: tidak

Info Selengkapnya

- Memulai Tugas (POST jobs)

InventoryRetrievalJobInput

Menyediakan opsi untuk menentukan berbagai tugas pengambilan inventaris.

Daftar Isi

EndDate

Akhir rentang tanggal, dalam waktu UTC, untuk pengambilan inventaris vault yang mencakup arsip yang dibuat sebelum tanggal ini.

Nilai String: Representasi string dalam format tanggal ISO 8601 (`YYYY-MM-DDThh:mm:ssTZD`) dalam detik, misalnya `2013-03-20T17:03:43Z`.

Tipe: String. Representasi string dalam format tanggal ISO 8601 (`YYYY-MM-DDThh:mm:ssTZD`) dalam detik, misalnya `2013-03-20T17:03:43Z`.

Wajib: tidak

Format

Format output untuk daftar inventaris vault, yang diatur oleh permintaan Memulai Tugas (POST jobs) saat memulai tugas untuk mengambil inventaris vault.

Nilai yang Valid: `CSV | JSON`

Wajib: tidak

Tipe: String

Batasi

Jumlah maksimum item inventaris yang dapat dikembalikan untuk setiap permintaan pengambilan inventaris vault.

Nilai yang valid: Nilai bilangan bulat lebih besar dari atau sama dengan 1.

Tipe: String

Wajib: tidak

Penanda

String buram yang mewakili tempat untuk melanjutkan pemberian nomor halaman hasil pengambilan inventaris vault. Anda menggunakan penanda ini dalam permintaan `Initiate Job` untuk mendapatkan item inventaris tambahan. Jika tidak ada item inventaris lainnya, nilai ini nol.

Tipe: String

Wajib: tidak

StartDate

Awal rentang tanggal, dalam waktu UTC, untuk pengambilan inventaris vault yang mencakup arsip yang dibuat pada atau setelah tanggal ini.

Nilai String: Representasi string dalam format tanggal ISO 8601 (`YYYY-MM-DDThh:mm:ssTZD`) dalam detik, misalnya `2013-03-20T17:03:43Z`.

Tipe: String. Representasi string dalam format tanggal ISO 8601 (`YYYY-MM-DDThh:mm:ssTZD`) dalam detik, misalnya `2013-03-20T17:03:43Z`.

Wajib: tidak

Info Selengkapnya

- Memulai Tugas (POST jobs)

jobParameters

Menyediakan opsi untuk mendefinisikan tugas.

Daftar Isi

ArchiveId

ID arsip yang Anda inginkan. Kolom ini wajib diisi jika bidang `Type` diatur ke `select` atau `archive-retrieval`. Terjadi kesalahan jika Anda menentukan bidang ini untuk permintaan tugas pengambilan inventaris.

Nilai yang Valid: Harus berupa ID arsip valid yang Anda peroleh dari permintaan sebelumnya ke Amazon Glacier (Amazon Glacier).

Tipe: String

Wajib: Ya ketika `Type` diatur ke `select` atau `archive-retrieval`.

Deskripsi

Deskripsi opsional untuk tugas.

Nilai yang Valid: Deskripsi harus kurang dari atau sama dengan 1.024 byte. Karakter yang diizinkan adalah ASCII 7 bit tanpa kode kontrol—khususnya, nilai ASCII 32–126 desimal atau 0x20–0x7E heksadesimal.

Tipe: String

Wajib: tidak

Format

(Opsional) Format output, saat memulai tugas untuk mengambil inventaris vault. Jika Anda memulai tugas inventaris dan tidak menentukan bidang `Format`, JSON adalah format default.

Nilai yang Valid: `CSV` | `JSON`

Tipe: String

Diperlukan: tidak

InventoryRetrievalParameters

Parameter input yang digunakan untuk pengambilan berbagai inventaris.

Tipe: Objek InventoryRetrievalJobInput

Diperlukan: tidak

OutputLocation

Objek yang berisi informasi tentang lokasi tempat hasil tugas pilih disimpan.

Tipe: Objek OutputLocation

Wajib: Ya, untuk tugas `select`.

RetrievalByteRange

Rentang byte untuk mengambil untuk`archive-retrieval`, dalam bentuk "*`StartByteValue`*-*`EndByteValue`*". Jika bidang ini tidak ditentukan, seluruh arsip akan

diambil. Jika bidang ini ditentukan, rentang byte harus megabyte (1024*1024) selaras. Megabyte-aligned berarti yang StartByteValueharus habis dibagi 1 MB, dan EndByteValueditambah 1 harus habis dibagi 1 MB atau menjadi akhir dari arsip yang ditentukan sebagai nilai ukuran byte arsip dikurangi 1. Jika RetrievalByteRangetidak selaras dengan megabyte, operasi ini mengembalikan respons. `400`

Terjadi kesalahan jika Anda menentukan bidang ini untuk permintaan tugas `inventory-retrieval` atau `select`.

Tipe: String

Diperlukan: tidak

SelectParameters

Objek yang berisi informasi tentang parameter yang digunakan untuk select.

Tipe: Objek SelectParameters

Diperlukan: tidak

SNSTopic

Nama Sumber Daya Amazon (ARN) dari topik Amazon SNS tempat Amazon Glacier mengirimkan pemberitahuan saat pekerjaan selesai dan output siap untuk Anda unduh. Topik yang ditentukan menerbitkan notifikasi kepada pelanggannya.

Topik SNS harus ada. Jika tidak, Amazon Glacier tidak membuatnya untuk Anda. Selain itu, topik SNS harus memiliki kebijakan yang memungkinkan akun yang membuat tugas menerbitkan pesan ke topik. Untuk informasi tentang nama topik SNS, lihat CreateTopicdi Referensi API Layanan Pemberitahuan Sederhana Amazon.

Tipe: String

Wajib: tidak

Tingkat

Tingkat yang digunakan untuk tugas pilih atau pengambilan arsip. `Standard` adalah nilai default yang digunakan.

Nilai yang Valid: `Expedited` | `Standard` | `Bulk`

Tipe: String

Wajib: tidak

Jenis

Jenis tugas. Anda dapat memulai tugas untuk melakukan kueri pilih di arsip, mengambil arsip, atau mendapatkan inventaris vault.

Nilai yang Valid: `select` | `archive-retrieval` | `inventory-retrieval`

Tipe: String

Wajib: ya

Info Selengkapnya

- Memulai Tugas (POST jobs)

OutputLocation

Berisi informasi tentang lokasi tempat hasil dan kesalahan tugas disimpan.

Daftar Isi

S3

Objek yang menggambarkan lokasi Amazon S3 untuk menerima hasil permintaan pemulihan.

Jenis: S3Location

Wajib: ya

Info Selengkapnya

- Memulai Tugas (POST jobs)

OutputSerialization

Menjelaskan cara output diserialkan.

Daftar Isi

CSV

Objek yang menjelaskan serialisasi hasil kueri yang dikodekan nilai yang dipisahkan koma (CSV).

Tipe: Objek CSVOutput

Wajib: tidak

Info Selengkapnya

- Memulai Tugas (POST jobs)

S3Location

Berisi informasi tentang lokasi Amazon S3 tempat hasil tugas disimpan.

Daftar Isi

AccessControlList

Daftar pemberian izin yang mengontrol akses ke hasil yang disimpan.

Tipe: Array objek Pemberian Izin

Wajib: tidak

BucketName

Nama bucket Amazon S3 tempat hasil tugas disimpan. Bucket harus berada di AWS Region yang sama dengan vault yang berisi objek arsip input.

Tipe: String

Wajib: ya

CannedACL

Daftar kontrol akses (ACL) terekam untuk mencantumkan ke hasil tugas.

Tipe: String

Nilai yang Valid: `private` | `public-read` | `public-read-write` | `aws-exec-read` | `authenticated-read` | `bucket-owner-read` | `bucket-owner-full-control`

Wajib: tidak

Enkripsi

Objek yang berisi informasi tentang enkripsi yang digunakan untuk menyimpan hasil tugas di Amazon S3.

Tipe: Objek Enkripsi

Wajib: tidak

Awalan

Prefiks yang diawali dengan hasil untuk permintaan ini. Panjang maksimum prefiks adalah 512 byte.

Tipe: String

Wajib: ya

StorageClass

Kelas penyimpanan yang digunakan untuk menyimpan hasil tugas.

Tipe: String

Nilai yang Valid: `STANDARD` | `REDUCED_REDUNDANCY` | `STANDARD_IA`

Wajib: tidak

Menandai

Serangkaian tanda yang diterapkan ke hasil tugas.

Tipe: Peta string ke string

Wajib: tidak

UserMetadata

Peta metadata untuk menyimpan dengan hasil tugas di Amazon S3.

Tipe: Peta string ke string

Wajib: tidak

Info Selengkapnya

- Memulai Tugas (POST jobs)

SelectParameters

Berisi informasi tentang parameter yang digunakan untuk select.

Daftar Isi

Ekspresi

Ekspresi yang digunakan untuk memilih objek. Ekspresi tidak boleh melebihi kuota 128.000 karakter.

Tipe: String

Wajib: ya

ExpressionType

Tipe ekspresi yang disediakan, misalnya SQL.

Nilai yang Valid: SQL

Tipe: String

Wajib: ya

InputSerialization

Menjelaskan format serialisasi objek dalam select.

Tipe: Objek InputSerialization

Wajib: tidak

OutputSerialization

Menjelaskan bagaimana hasil dari tugas pilih diserialkan.

Wajib: tidak

Tipe: Objek OutputSerialization

Info Selengkapnya

- Memulai Tugas (POST jobs)

Operasi Pengambilan Data

Berikut ini adalah operasi terkait pengambilan data yang tersedia di Amazon Glacier.

Topik

- Mendapatkan Kebijakan Pengambilan Data (GET policy)
- Mencantumkan Kapasitas yang Disediakan (GET provisioned-capacity)
- Membeli Kapasitas yang Disediakan (GET provisioned-capacity)
- Mengatur Kebijakan Pengambilan Data (PUT policy)

Mendapatkan Kebijakan Pengambilan Data (GET policy)

Deskripsi

Operasi ini mengembalikan kebijakan pengambilan data saat ini untuk Akun AWS dan AWS Wilayah yang ditentukan dalam `GET` permintaan. Untuk informasi selengkapnya tentang kebijakan pengambilan data, lihat Kebijakan Pengambilan Data Amazon Glacier.

Permintaan

Untuk mengembalikan kebijakan pengambilan data saat ini, kirim permintaan `GET` HTTP ke URI kebijakan pengambilan data seperti yang ditunjukkan dalam contoh sintaks berikut.

Sintaksis

```
GET /AccountId/policies/data-retrieval HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: Date
```

```
Authorization: SignatureValue
x-amz-glacier-version: 2012-06-01
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID. Nilai ini harus cocok dengan Akun AWS ID yang terkait dengan kredensional yang digunakan untuk menandatangani permintaan. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menentukan ID akun, jangan sertakan tanda hubung ('-') di ID.

Parameter Permintaan

Operasi ini tidak menggunakan parameter permintaan.

Header Permintaan

Operasi ini hanya menggunakan header permintaan yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header permintaan umum, lihat Header Permintaan Umum.

Isi Permintaan

Operasi ini tidak memiliki isi permintaan.

Respons

Sintaksis

```
HTTP/1.1 200 OK
x-amzn-RequestId: x-amzn-RequestId
Date: Date
Content-Type: application/json
Content-Length: length
{
  "Policy":
    {
      "Rules":[
         {
```

```
            "BytesPerHour": Number,
            "Strategy": String
          }
        ]
    }
}
```

Header Respons

Operasi ini hanya menggunakan header respons yang umum untuk sebagian besar respons. Untuk informasi tentang header permintaan umum, lihat Header Respons Umum.

Isi Respons

Isi respons berisi bidang JSON berikut.

BytesPerHour

Jumlah maksimum byte yang dapat diambil dalam satu jam.

Bidang ini hanya akan muncul jika nilai bidang Strategy (Strategi) adalah `BytesPerHour`.

Tipe: Angka

Aturan

Aturan kebijakan. Meskipun ini adalah jenis daftar, saat ini hanya akan ada satu aturan, yang berisi bidang Strategi dan opsional BytesPerHour bidang.

Tipe: Array

Strategi

Tipe kebijakan pengambilan data.

Tipe: String

Nilai yang valid: `BytesPerHour`|`FreeTier`|None. `BytesPerHour` sama dengan memilih Max Retrieval Rate (Tingkat Pengambilan Maks.) di konsol. `FreeTier` sama dengan memilih Free Tier Only (Hanya Tingkat Gratis) di konsol. None sama dengan memilih No Retrieval Policy (Tidak Ada Kebijakan) di konsol. Untuk informasi selengkapnya tentang memilih kebijakan pengambilan data di konsol, lihat Kebijakan Pengambilan Data Amazon Glacier.

Kesalahan

Untuk informasi tentang pengecualian Amazon Glacier dan pesan kesalahan, lihat. Respons Kesalahan

Contoh

Contoh berikut menunjukkan cara mendapatkan kebijakan pengambilan data.

Contoh Permintaan

Dalam contoh ini, permintaan GET dikirim ke URI dari lokasi kebijakan.

```
GET /-/policies/data-retrieval HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
x-amz-glacier-version: 2012-06-01
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
```

Contoh Respons

Respons yang sukses menunjukkan kebijakan pengambilan data dalam isi respons dalam format JSON.

```
HTTP/1.1 200 OK
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:00:00 GMT
Content-Type: application/json
Content-Length: 85

{
  "Policy":
    {
      "Rules":[
         {
           "BytesPerHour":10737418240,
           "Strategy":"BytesPerHour"
          }
       ]
    }
}
```

Bagian Terkait

- Mengatur Kebijakan Pengambilan Data (PUT policy)
- Memulai Tugas (POST jobs)

Mencantumkan Kapasitas yang Disediakan (GET provisioned-capacity)

Operasi ini mencantumkan unit kapasitas yang disediakan untuk Akun AWS yang ditentukan. Untuk informasi selengkapnya tentang kapasitas yang disediakan, lihat Opsi Pengambilan Arsip.

Unit kapasitas yang disediakan berlangsung selama satu bulan mulai tanggal dan waktu pembelian, yang merupakan tanggal mulai. Unit ini berakhir pada tanggal kedaluwarsa, yang tepat satu bulan setelah tanggal mulai ke detik terdekat.

Jika tanggal mulai adalah tanggal 31, tanggal kedaluwarsa adalah hari terakhir bulan berikutnya. Misalnya, jika tanggal mulai adalah 31 Agustus, tanggal kedaluwarsa adalah 30 September. Jika tanggal mulai 31 Januari, tanggal kedaluwarsa adalah 28 Februari. Anda dapat melihat fungsi ini di Contoh Respons.

Sintaks Permintaan

Untuk mencantumkan kapasitas pengambilan yang disediakan untuk akun, kirim permintaan HTTP GET ke URI kapasitas yang disediakan seperti yang ditunjukkan dalam contoh sintaks berikut.

```
GET /AccountId/provisioned-capacity HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: Date
Authorization: SignatureValue
x-amz-glacier-version: 2012-06-01
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID. Nilai ini harus cocok dengan Akun AWS ID yang terkait dengan kredensional yang digunakan untuk menandatangani permintaan. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini

Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menentukan ID akun, jangan sertakan tanda hubung ('-') di ID.

Parameter Permintaan

Operasi ini tidak menggunakan parameter permintaan.

Header Permintaan

Operasi ini hanya menggunakan header permintaan yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header permintaan umum, lihat Header Permintaan Umum.

Isi Permintaan

Operasi ini tidak memiliki isi permintaan.

Respons

Jika operasi berhasil, layanan mengirimkan kembali respon `200 OK` HTTP.

Sintaks Respons

```
HTTP/1.1 200 OK
x-amzn-RequestId: x-amzn-RequestId
Date: Date
Content-Type: application/json
Content-Length: Length
{
   "ProvisionedCapacityList":
      {
         "CapacityId" : "string",
         "StartDate" : "string"
         "ExpirationDate" : "string"
      }
}
```

Header Respons

Operasi ini hanya menggunakan header respons yang umum untuk sebagian besar respons. Untuk informasi tentang header permintaan umum, lihat Header Respons Umum.

Isi Respons

Isi respons berisi bidang JSON berikut.

CapacityId

ID yang mengidentifikasi unit kapasitas yang disediakan.

Tipe: String.

StartDate

Tanggal pembelian unit kapasitas yang disediakan, dalam Waktu Universal Terkoordinasi (UTC).

Tipe: String. Representasi string dalam format tanggal ISO 8601, misalnya `2013-03-20T17:03:43.221Z`.

ExpirationDate

Tanggal berakhirnya unit kapasitas yang disediakan, dalam Waktu Universal Terkoordinasi (UTC).

Tipe: String. Representasi string dalam format tanggal ISO 8601, misalnya `2013-03-20T17:03:43.221Z`.

Kesalahan

Untuk informasi tentang pengecualian Amazon Glacier dan pesan kesalahan, lihat. Respons Kesalahan

Contoh

Contoh berikut mencantumkan unit kapasitas yang disediakan untuk akun.

Contoh Permintaan

Dalam contoh ini, permintaan GET dikirim untuk mengambil daftar unit kapasitas yang disediakan untuk akun yang ditentukan.

```
GET /123456789012/priority-capacity HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
x-amz-glacier-version: 2012-06-01
```

```
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
```

Contoh Respons

Jika permintaan berhasil, Amazon Glacier (Amazon Glacier) mengembalikan dengan daftar unit kapasitas `HTTP 200 OK` yang disediakan untuk akun seperti yang ditunjukkan pada contoh berikut.

Unit kapasitas yang disediakan terdaftar pertama adalah contoh unit dengan tanggal mulai 31 Januari 2017 dan tanggal kedaluwarsa 28 Februari 2017. Seperti yang dijelaskan sebelumnya, jika tanggal mulai adalah tanggal 31, tanggal kedaluwarsa adalah hari terakhir bulan berikutnya.

```
HTTP/1.1 200 OK
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:02:00 GMT
Content-Type: application/json
Content-Length: length

{
   "ProvisionedCapacityList",
      {
         "CapacityId": "zSaq7NzHFQDANTfQkDen4V7z",
         "StartDate": "2017-01-31T14:26:33.031Z",
         "ExpirationDate": "2017-02-28T14:26:33.000Z",
      },
      {
         "CapacityId": "yXaq7NzHFQNADTfQkDen4V7z",
         "StartDate": "2016-12-13T20:11:51.095Z"",
         "ExpirationDate": "2017-01-13T20:11:51.000Z" ",
      },
      ...
}
```

Bagian Terkait

- Membeli Kapasitas yang Disediakan (GET provisioned-capacity)

Membeli Kapasitas yang Disediakan (GET provisioned-capacity)

Operasi ini membeli unit kapasitas yang disediakan untuk Akun AWS.

Unit kapasitas yang disediakan berlangsung selama satu bulan mulai tanggal dan waktu pembelian, yang merupakan tanggal mulai. Unit ini berakhir pada tanggal kedaluwarsa, yang tepat satu bulan setelah tanggal mulai ke detik terdekat.

Jika tanggal mulai adalah tanggal 31, tanggal kedaluwarsa adalah hari terakhir bulan berikutnya. Misalnya, jika tanggal mulai adalah 31 Agustus, tanggal kedaluwarsa adalah 30 September. Jika tanggal mulai 31 Januari, tanggal kedaluwarsa adalah 28 Februari.

Kapasitas yang disediakan membantu memastikan kapasitas pengambilan Anda untuk pengambilan yang dipercepat tersedia saat Anda membutuhkannya. Setiap unit kapasitas memastikan bahwa setidaknya tiga pengambilan yang dipercepat dapat dilakukan setiap lima menit dan menyediakan hingga 150 throughput pengambilan. MB/s Untuk informasi selengkapnya tentang kapasitas yang disediakan, lihat Opsi Pengambilan Arsip.

> **Note**
>
> Ada batas dua unit kapasitas yang disediakan per. Akun AWS

Permintaan

Untuk membeli unit kapasitas yang disediakan untuk Akun AWS mengirim `POST` permintaan HTTP ke URI kapasitas yang disediakan.

Sintaksis

```
POST /AccountId/provisioned-capacity HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: Date
Authorization: SignatureValue
Content-Length: Length
x-amz-glacier-version: 2012-06-01
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID. Nilai ini harus cocok dengan Akun AWS ID yang terkait dengan kredensional yang digunakan untuk menandatangani permintaan. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan

untuk menandatangani permintaan. Jika Anda menentukan ID akun, jangan sertakan tanda hubung ('-') di ID.

Parameter Permintaan

Header Permintaan

Operasi ini hanya menggunakan header permintaan yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header permintaan umum, lihat Header Permintaan Umum.

Isi Permintaan

Operasi ini tidak memiliki isi permintaan.

Respons

Jika permintaan operasi berhasil, layanan mengembalikan respons `201 Created` HTTP.

Sintaksis

```
HTTP/1.1 201 Created
x-amzn-RequestId: x-amzn-RequestId
Date: Date
x-amz-capacity-id: CapacityId
```

Header Respons

Respons yang sukses termasuk header respon berikut, selain header respons yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header respons umum, lihat Header Respons Umum.

Nama	Deskripsi
`x-amz-capacity-id`	ID yang mengidentifikasi unit kapasitas yang disediakan. Tipe: String

Isi Respons

Operasi ini tidak mengembalikan isi respons.

Kesalahan

Operasi ini mencakup kesalahan atau kesalahan berikut, di samping kemungkinan kesalahan yang umum terjadi pada semua operasi Amazon Glacier. Untuk informasi tentang kesalahan Amazon Glacier dan daftar kode kesalahan, lihat. Respons Kesalahan

Kode	Deskripsi	Kode Status HTTP	Tipe
`LimitExceededException`	Dikembalikan jika permintaan yang diberikan akan melebihi batas akun unit kapasitas yang disediakan.	`400 Bad Request`	Klien

Contoh

Contoh berikut membeli unit kapasitas yang disediakan untuk akun.

Contoh Permintaan

Contoh berikut mengirimkan permintaan HTTP POST untuk membeli unit kapasitas yang disediakan.

```
POST /123456789012/provisioned-capacity HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
Content-Length: length
x-amz-glacier-version: 2012-06-01
```

Contoh Respons

Jika permintaan berhasil, Amazon Glacier (Amazon Glacier) mengembalikan respons, seperti yang ditunjukkan pada contoh `HTTP 201 Created` berikut.

```
HTTP/1.1 201 Created
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:02:00 GMT
```

```
x-amz-capacity-id: zSaq7NzHFQDANTfQkDen4V7z
```

Bagian Terkait

- Mencantumkan Kapasitas yang Disediakan (GET provisioned-capacity)

Mengatur Kebijakan Pengambilan Data (PUT policy)

Deskripsi

Operasi ini menetapkan dan kemudian memberlakukan kebijakan pengambilan data di AWS Wilayah yang ditentukan dalam permintaan. PUT Anda dapat menetapkan satu kebijakan per AWS Wilayah untuk Akun AWS. Kebijakan ini diterapkan dalam beberapa menit setelah operasi PUT berhasil.

Operasi kebijakan yang ditetapkan tidak memengaruhi pengambilan tugas yang sedang berlangsung sebelum kebijakan diterapkan. Untuk informasi selengkapnya tentang kebijakan pengambilan data, lihat Kebijakan Pengambilan Data Amazon Glacier.

Permintaan

Sintaksis

Untuk menetapkan kebijakan pengambilan data, kirim permintaan HTTP PUT ke URI kebijakan pengambilan data seperti yang ditunjukkan dalam contoh sintaks berikut.

```
PUT /AccountId/policies/data-retrieval HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: Date
Authorization: SignatureValue
Content-Length: Length
x-amz-glacier-version: 2012-06-01

{
  "Policy":
    {
      "Rules":[
         {
             "Strategy": String,
             "BytesPerHour": Number
```

```
        }
      ]
    }
}
```

> **Note**
>
> `AccountId`Nilainya adalah Akun AWS ID. Nilai ini harus cocok dengan Akun AWS ID yang terkait dengan kredensional yang digunakan untuk menandatangani permintaan. Anda dapat menentukan Akun AWS ID atau secara opsional satu '-' (tanda hubung), dalam hal ini Amazon Glacier menggunakan Akun AWS ID yang terkait dengan kredenal yang digunakan untuk menandatangani permintaan. Jika Anda menentukan ID akun, jangan sertakan tanda hubung ('-') di ID.

Parameter Permintaan

Operasi ini tidak menggunakan parameter permintaan.

Header Permintaan

Operasi ini hanya menggunakan header permintaan yang umum untuk semua operasi. Untuk informasi selengkapnya tentang header permintaan umum, lihat Header Permintaan Umum.

Isi Permintaan

Isi permintaan berisi bidang JSON berikut.

BytesPerHour

Jumlah maksimum byte yang dapat diambil dalam satu jam.

Bidang ini hanya diperlukan jika nilai bidang Strategy (Strategi) adalah `BytesPerHour`. Operasi PUT Anda akan ditolak jika bidang Strategy (Strategi) tidak ditetapkan ke `BytesPerHour` dan Anda menetapkan kolom ini.

Tipe: Angka

Wajib: Ya, jika bidang Strategy (Strategi) diatur ke `BytesPerHour`. Jika tidak, tidak.

Nilai yang valid: Nilai bilangan bulat minimum sebesar 1. Nilai bilangan bulat maksimum 2^63 - 1 inklusif.

Aturan

Aturan kebijakan. Meskipun ini adalah jenis daftar, saat ini harus ada hanya satu aturan, yang berisi bidang Strategi dan opsional BytesPerHour bidang.

Tipe: Array

Wajib: Ya

Strategi

Tipe kebijakan pengambilan data yang ditetapkan.

Tipe: String

Wajib: Ya

Nilai yang valid: `BytesPerHour`|`FreeTier`|None. `BytesPerHour` sama dengan memilih Max Retrieval Rate (Tingkat Pengambilan Maks.) di konsol. `FreeTier` sama dengan memilih Free Tier Only (Hanya Tingkat Gratis) di konsol. None sama dengan memilih No Retrieval Policy (Tidak Ada Kebijakan) di konsol. Untuk informasi selengkapnya tentang memilih kebijakan pengambilan data di konsol, lihat Kebijakan Pengambilan Data Amazon Glacier.

Respons

Sintaksis

```
HTTP/1.1 204 No Content
x-amzn-RequestId: x-amzn-RequestId
Date: Date
```

Header Respons

Operasi ini hanya menggunakan header respons yang umum untuk sebagian besar respons. Untuk informasi tentang header permintaan umum, lihat Header Respons Umum.

Isi Respons

Operasi ini tidak mengembalikan isi respons.

Kesalahan

Untuk informasi tentang pengecualian Amazon Glacier dan pesan kesalahan, lihat. Respons Kesalahan

Contoh

Contoh Permintaan

Contoh berikut mengirimkan permintaan HTTP PUT dengan bidang Strategy (Strategi) diatur ke `BytesPerHour`.

```
PUT /-/policies/data-retrieval HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
x-amz-glacier-version: 2012-06-01
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2

{
  "Policy":
    {
      "Rules":[
         {
             "Strategy":"BytesPerHour",
             "BytesPerHour":10737418240
          }
       ]
    }
}
```

Contoh berikut mengirimkan permintaan HTTP PUT dengan bidang Strategy (Strategi) diatur ke `FreeTier`.

```
PUT /-/policies/data-retrieval HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
x-amz-glacier-version: 2012-06-01
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
```

```
{
  "Policy":
    {
      "Rules":[
         {
             "Strategy":"FreeTier"
          }
       ]
    }
}
```

Contoh berikut mengirimkan permintaan HTTP PUT dengan bidang Strategy (Strategi) diatur ke None.

```
PUT /-/policies/data-retrieval HTTP/1.1
Host: glacier.us-west-2.amazonaws.com
x-amz-Date: 20170210T120000Z
x-amz-glacier-version: 2012-06-01
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/
us-west-2/glacier/aws4_request,SignedHeaders=host;x-amz-date;x-amz-glacier-
version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2

{
  "Policy":
    {
      "Rules":[
         {
             "Strategy":"None"
          }
       ]
    }
}
```

Contoh Respons

Jika permintaan berhasil Amazon Glacier (Amazon Glacier) menetapkan kebijakan dan mengembalikan HTTP `204 No Content` seperti yang ditunjukkan pada contoh berikut.

```
HTTP/1.1 204 No Content
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:02:00 GMT
```

Bagian Terkait

- Mendapatkan Kebijakan Pengambilan Data (GET policy)

- Memulai Tugas (POST jobs)

Riwayat Dokumen

- Versi produk saat ini: 2012-06-01

Tabel berikut menjelaskan perubahan penting dalam setiap rilis Panduan Pengembang Amazon Glacier mulai 5 Juli 2018, dan seterusnya. Untuk notifikasi tentang pembaruan dokumentasi ini, Anda dapat berlangganan ke umpan RSS.

Perubahan	Deskripsi	Tanggal
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
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
Perubahan nama Amazon Glacier	Amazon Glacier sekarang menjadi Amazon Glacier untuk lebih mencerminkan integrasi Glacier dengan Amazon S3.	20 November 2018
Pembaruan kini tersedia melalui RSS	Anda sekarang dapat berlangganan umpan RSS untuk menerima pemberita huan tentang pembaruan	5 Juli 2018

panduan Pengembang
Amazon Glacier.

Pembaruan Sebelumnya

Tabel berikut menjelaskan perubahan penting dalam setiap rilis Panduan Pengembang Amazon Glacier sebelum 5 Juli 2018.

Ubah	Deskripsi	Tanggal Rilis
Pengambilan Data yang Dipercepat dan Massal	Amazon Glacier sekarang mendukung pengambil an data Dipercepat dan Massal selain pengambilan Standar. Untuk informasi selengkapnya, lihat Opsi Pengambilan Arsip.	21 November 2016
Kunci Vault	Amazon Glacier sekarang mendukung Vault Lock, yang memungkinkan Anda menerapkan dan menerapkan kontrol kepatuhan dengan mudah pada kubah Amazon Glacier individual dengan kebijakan Vault Lock. Untuk informasi selengkapnya, lihat Kunci Gletser Amazon Glacier dan Kebijakan Vault Lock.	8 Juli 2015
Penandaan Vault	Amazon Glacier sekarang memungkinkan Anda menandai brankas Amazon Glacier Anda untuk pengelolaan sumber daya dan biaya yang lebih mudah. Tag adalah label yang dapat Anda tentukan dan kaitkan dengan brankas Anda, dan menggunakan tag menambahkan kemampuan pemfilteran ke operasi seperti AWS laporan biaya. Untuk informasi selengkap nya, lihat Menandai Sumber Daya Amazon Glacier dan Menandai Brankas Amazon Glacier Anda.	22 Juni 2015
Kebijakan akses vault	Amazon Glacier sekarang mendukung pengelola an akses ke brankas Amazon Glacier pribadi Anda dengan menggunakan kebijakan akses vault. Sekarang Anda dapat menentukan kebijakan akses secara langsung di vault, sehingga lebih mudah untuk	27 April 2015

Ubah	Deskripsi	Tanggal Rilis
	memberikan akses ke pengguna dan grup bisnis internal untuk organisasi Anda, serta partner bisnis eksternal Anda. Untuk informasi selengkapnya, lihat Kebijakan Akses Vault.	
Kebijakan pengambilan data dan pencatatan audit	Amazon Glacier sekarang mendukung kebijakan pengambilan data dan pencatatan audit. Kebijakan pengambilan data memungkinkan Anda mengatur batas pengambilan data dengan mudah dan menyederhanakan manajemen biaya pengambilan data. Anda dapat menentukan batas pengambilan data Anda sendiri dengan beberapa klik di Konsol Manajemen AWS atau dengan menggunakan Amazon Glacier API. Untuk informasi selengkapnya, lihat Kebijakan Pengambilan Data Amazon Glacier. Selain itu, Amazon Glacier sekarang mendukung pencatatan audit, AWS CloudTrail yang merekam panggilan Amazon Glacier API untuk akun Anda dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Untuk informasi selengkapnya, lihat Mencatat Panggilan API Amazon Glacier dengan AWS CloudTrail.	11 Desember 2014
Pembaruan untuk sampel Java	Memperbarui contoh kode Java dalam panduan ini yang menggunakan file AWS SDK for Java.	27 Juni 2014
Membatasi pengambilan inventaris vault	Anda sekarang dapat membatasi jumlah item inventaris vault yang diambil dengan memfilter tanggal pembuatan arsip atau dengan menetapkan batas. Untuk informasi selengkapnya tentang membatasi pengambilan inventaris, lihat Pengambilan Inventaris Rentang di topik Memulai Tugas (POST jobs).	31 Desember 2013
Dihapus usang URLs	Menghapus URLs yang menunjuk ke halaman kredensi keamanan lama dari contoh kode.	26 Juli 2013

Ubah	Deskripsi	Tanggal Rilis
Dukungan untuk berbagai pengambilan	Amazon Glacier sekarang mendukung pengambilan rentang spesifik arsip Anda. Anda dapat memulai pekerjaan yang meminta Amazon Glacier untuk menyiapkan seluruh arsip atau sebagian arsip untuk unduhan berikutnya. Ketika ukuran arsip sangat besar, Anda mungkin merasakan penghematan biaya dengan memulai beberapa tugas berurutan untuk mempersiapkan arsip Anda. Untuk informasi selengkapnya, lihat Mengunduh Arsip di Amazon Glacier.	13 November 2012
Panduan Baru	Ini adalah rilis pertama dari Panduan Pengembang Amazon Glacier.	20 Agustus 2012

AWS Glosarium

Untuk AWS terminologi terbaru, lihat AWS glosarium di Referensi.Glosarium AWS