Guia do desenvolvedor

Amazon Glacier

Versão da API 2012-06-01

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

Amazon Glacier: Guia do desenvolvedor

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

As marcas comerciais e imagens de marcas da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestigie a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

..... x
O que é o Amazon Glacier? 1
Você usa atualmente o Amazon Glacier? 1
Modelo de dados 3
Cofre 3
Arquivo 4
Trabalho 5
Configuração de notificação 6
Operações com Suporte 6
Operações de cofre 7
Operações de arquivo 7
Tarefas 7
Acessar o Amazon Glacier 7
Regiões e endpoints 8
Conceitos básicos 9
Etapa 1: Antes de começar 10
Configurar um Conta da AWS 10
Baixe o AWS SDK apropriado 12
Etapa 2: Criar um cofre 13
Etapa 3: Fazer upload de um arquivo para um cofre 14
Fazer upload de um arquivo usando o Java 15
Fazer upload de um arquivo usando o .NET 21
Etapa 4: Fazer download de um arquivo de um cofre 23
Fazer download de um arquivo usando o Java 24
Fazer download de um arquivo usando o .NET 26
Etapa 5: Excluir um arquivo de um cofre 28
Seções relacionadas 28
Excluir um arquivo usando Java 28
Excluir um arquivo usando o .NET 30
Excluindo um arquivo usando o AWS CLI 31
Etapa 6: Excluir um cofre 34
Para onde ir agora? 35
Trabalhar com cofres 36
Operações de cofre no Amazon Glacier 37

Criar e excluir cofres ... 37
Recuperar metadados do cofre ... 37
Fazer download de um inventário de cofre ... 37
Configurar notificações de cofre ... 38
Criar de um cofre ... 39
Como criar um cofre usando o Java ... 39
Como criar um cofre usando o .NET ... 43
Criar um cofre usando REST ... 47
Como criar um cofre usando o console ... 47
Criar um cofre usando a AWS CLI ... 47
Recuperar metadados do cofre ... 49
Recuperar metadados do cofre usando o Java ... 49
Recuperar metadados do cofre usando o .NET ... 52
Recuperar metadados do cofre usando REST ... 55
Recuperar metadados do cofre no usando o AWS CLI ... 55
Fazer download de um inventário de cofre ... 56
Sobre o inventário ... 58
Fazer download de um inventário de cofre usando o Java ... 59
Fazer download de um inventário de cofre usando o .NET ... 66
Fazer download de um inventário de cofre usando REST ... 74
Baixando um inventário do Vault usando o AWS CLI ... 74
Configurar notificações de cofre ... 77
Conceitos gerais ... 78
Configurar notificações de cofre usando o Java ... 80
Configurar notificações de cofre usando o .NET ... 83
Configurar notificações de cofre usando a API REST ... 86
Configurar as notificações de cofre usando o console ... 86
Configurar notificações de cofre usando o CLI ... 88
Excluir um cofre ... 90
Excluir um cofre usando o Java ... 90
Excluir um cofre usando o .NET ... 92
Excluir um cofre usando REST ... 93
Excluir um cofre vazio usando o console ... 93
Excluindo um cofre usando o AWS CLI ... 94
Marcar cofres ... 98
Marcação de cofres usando o console do Amazon Glacier ... 98

Marcar cofres usando a API do AWS CLI 100
Marcação de cofres usando a API do Amazon Glacier 101
Seções relacionadas 101
Vault Lock 101
Visão geral do bloqueio de cofre 102
Bloquear um cofre usando a API 103
Bloquear um cofre usando a CLI 104
Bloqueio do cofre usando o console 106
Trabalhar com arquivos 109
Operações de arquivo 110
Upload de um arquivo 110
Localizar um arquivo 110
Download de um arquivo 110
Exclusão de um arquivo 111
Atualizar um arquivo 111
Manter metadados de arquivo no lado do cliente 111
Upload de um arquivo 112
Opções para fazer upload de um arquivo 112
Fazer upload de um arquivo em uma única operação 113
Upload de arquivos grandes em partes 124
Download de um arquivo 142
Recuperar arquivos 142
Fazer download de um arquivo usando o Java 147
Fazer download de um arquivo usando o .NET 164
Baixando um arquivo grande usando Python 181
Fazer download de um arquivo usando REST 189
Baixando um arquivo usando o AWS CLI 189
Exclusão de um arquivo 193
Excluir um arquivo usando Java 194
Excluir um arquivo usando o .NET 196
Excluir um arquivo usando REST 199
Excluindo um arquivamento usando o AWS CLI 199
Usando o AWS SDKs 203
AWS Bibliotecas do SDK para Java e.NET 203
O que é API de nível inferior? 203
O que é API de nível superior? 204

Quando usar as APIs de nível superior e de nível inferior 204
Trabalhando com AWS SDKs 205
Usando o AWS SDK para Java 206
Usar a API de nível inferior 206
Usar a API de nível superior 207
Execução de exemplos de Java usando o Eclipse 208
Definição do endpoint 209
Como usar o AWS SDK para .NET 209
Usar a API de nível inferior 210
Usar a API de nível superior 211
Executar exemplos do .NET 212
Definição do endpoint 212
Exemplos de código 214
Conceitos básicos 215
Olá, Amazon Glacier 216
Ações 217
Cenários 283
Arquivar um arquivo, obter notificações e iniciar um trabalho 284
Obter o conteúdo do arquivo e excluir o arquivo 290
Segurança 296
Proteção de dados 297
Criptografia de dados 297
Gerenciamento de chaves 298
Privacidade do tráfego entre redes 298
Gerenciamento de Identidade e Acesso 299
Público 299
Autenticação com identidades 300
Gerenciar o acesso usando políticas 301
Como funciona o Amazon Glacier com o IAM 303
Exemplos de políticas baseadas em identidade 310
Exemplos de políticas baseadas em atributos 316
Solução de problemas 319
Referência de permissões de API do Amazon Glacier 321
Registro e Monitoramento 329
Validação de Conformidade 330
Resiliência 332

Segurança da infraestrutura 333
Endpoints da VPC 333
Políticas de recuperação de dados 334
Escolher uma política de recuperação de dados do Amazon Glacier 334
Política Somente nível gratuito 335
Política Max Retrieval Rate 336
Política Sem limite de recuperação 336
Usar o console do Amazon Glacier para configurar uma política de recuperação de dados 337
Usar a API do Amazon Glacier para configurar uma política de recuperação de dados 337
Usar a API REST do Amazon Glacier para configurar uma política de recuperação de dados 338
Usando o AWS SDKs para configurar uma política de recuperação de dados 338
Marcação de recursos 339
Conceitos básicos da marcação 339
Restrições de tag 340
Monitoramento de custos com marcação 340
Gerenciar controle de acesso com marcação 341
Seções relacionadas 341
Registro de auditoria com AWS CloudTrail 342
Informações sobre o Amazon Glacier em CloudTrail 342
Noções básicas de entradas de arquivos de log do Amazon Glacier 343
Referência da API 347
Cabeçalhos de solicitação comuns 348
Cabeçalhos de resposta comuns 351
Solicitações de assinatura 352
Cálculo de assinatura de exemplo 353
Calcular assinaturas para as operações de streaming 355
Computar somas de verificação 357
Exemplo do hash de árvore 1: fazer upload de um arquivo em uma única solicitação 359
Exemplo do hash de árvore 2: fazer upload de um arquivo usando um multipart upload 360
Computar o hash de árvore de um arquivo 361
Receber somas de verificação durante o download de dados 370
Respostas de erro 373
Exemplo 1: descrever solicitação de trabalho com um ID de trabalho não existente 376
Exemplo 2: solicitação List Jobs com um valor inválido para o parâmetro da solicitação 377
Operações de cofre 378

Anular o Vault Lock ... 379
Adicionar tags ao cofre ... 382
Crie cofre ... 385
Concluir bloqueio de cofre ... 388
Excluir cofre ... 392
Excluir política de acesso do cofre ... 395
Excluir notificações do cofre ... 397
Descrever cofre ... 400
Obter a política de acesso do cofre ... 404
Obter bloqueio do cofre ... 408
Obter notificações do cofre ... 413
Iniciar bloqueio do cofre ... 416
Listar tags para cofre ... 421
Listar cofres ... 424
Remover tags do cofre ... 431
Definir política de acesso do cofre ... 435
Definir configuração de notificação do cofre ... 438
Operações de arquivo ... 442
Excluir arquivo ... 442
Fazer upload de arquivo ... 445
Operações de multipart upload ... 451
Anular carregamento fracionado ... 451
Concluir carregamento fracionado ... 454
Iniciar multipart upload ... 459
Listar partes ... 465
Listar carregamentos fracionados ... 472
Carregar parte ... 479
Operações de trabalho ... 486
Descrever trabalho ... 486
Obter saída de trabalho ... 497
Iniciar trabalho ... 508
Listar trabalhos ... 519
Tipos de dados usados em operações de trabalho ... 529
CSVInput ... 530
CSVOutput ... 531
Criptografia ... 533

GlacierJobDescription 533
Concessão 537
Grantee 538
InputSerialization 539
InventoryRetrievalJobInput 539
jobParameters 541
OutputLocation 544
OutputSerialization 544
S3Location 545
SelectParameters 547
Operações de recuperação de dados 548
Obter política de recuperação de dados 548
Listar a capacidade de provisão 552
Comprar capacidade provisionada 556
Definir política de recuperação de dados 559
Histórico do documento 565
Atualizações anteriores 566
Glossário da AWS 569

Esta página destina-se somente a clientes atuais do serviço Amazon Glacier que usam cofres e a API REST original de 2012.

Se você estiver procurando soluções de armazenamento de arquivos do Amazon Glacier, recomendamos usar as classes de armazenamento do Amazon S3, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive. Para saber mais sobre essas opções de armazenamento, consulte Classes de armazenamento do Amazon Glacier.

O Amazon Glacier (serviço autônomo original baseado em cofre) não está mais aceitando novos clientes. O Amazon Glacier é um serviço independente APIs que armazena dados em cofres e é diferente das classes de armazenamento Amazon S3 e Amazon S3 Glacier. Seus dados existentes permanecerão seguros e acessíveis no Amazon Glacier indefinidamente. Nenhuma migração é necessária. Para armazenamento de arquivamento de baixo custo e longo prazo, AWS recomenda as classes de armazenamento Amazon S3 Glacier, que oferecem uma experiência superior ao cliente com APIs base em buckets S3, disponibilidade Região da AWS total, custos mais baixos e integração de serviços. AWS Se você quiser recursos aprimorados, considere migrar para as classes de armazenamento do Amazon S3 Glacier usando nossas Orientações de soluções da AWS para transferir dados dos cofres do Amazon Glacier para as classes de armazenamento do Amazon S3 Glacier.

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.

O que é o Amazon Glacier?

Se você estiver usando atualmente o serviço do Amazon Glacier e quiser saber mais, encontrará as informações necessárias neste guia. Amazon Glacier é um serviço de armazenamento seguro, durável e de baixo custo para o arquivamento de dados e backup de longo prazo utilizando cofres. Para obter informações sobre a definição de preço do serviço, consulte Definição de preços do Amazon Glacier.

Tópicos

- Você usa atualmente o Amazon Glacier?
- Modelo de dados do Amazon Glacier
- Operações compatíveis no Amazon Glacier
- Acessar o Amazon Glacier

Você usa atualmente o Amazon Glacier?

> **Note**
>
> Esta seção é sobre o serviço do Amazon Glacier. Se você usa atualmente as classes de armazenamento do Amazon S3 Glacier (S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive), consulte Classes de armazenamento para arquivamento de objetos no Guia do usuário do Amazon S3.

Se você usa atualmente o serviço do Amazon Glacier e quiser saber mais, recomendamos que leia primeiro as seguintes seções:

- O que é o Amazon Glacier? O restante desta seção descreve o modelo de dados subjacente, as operações que ele suporta e as AWS SDKs que você pode usar para interagir com o serviço.
- Conceitos básicos: a seção Conceitos básicos do Amazon Glacier orienta em meio ao processo de criar um cofre, fazer upload de arquivos, criar trabalhos para fazer download de arquivos, recuperar a saída do trabalho e a excluir arquivos.

> **Important**
>
> O Amazon Glacier fornece um console. No entanto, qualquer operação de arquivamento, como upload, download ou exclusão, exige que você use o AWS Command Line Interface (AWS CLI) ou escreva o código. Não há suporte ao console para operações de arquivo. Por exemplo, para fazer upload de dados, como fotos, vídeos e outros documentos, você deve usar o código AWS CLI ou escrever o código para fazer solicitações, usando a API REST diretamente ou usando AWS SDKs o.
> Para instalar o AWS CLI, consulte AWS Command Line Interface. Para obter mais informações sobre o uso do Amazon Glacier com a AWS CLI, consulte a Referência da AWS CLI para o Amazon Glacier. Para exemplos de uso do AWS CLI para fazer upload de arquivos para o Amazon Glacier, consulte Usando o Amazon Glacier com o. AWS Command Line Interface

Além da seção de conceitos básicos, é desejável saber mais sobre as operações do Amazon Glacier. As seções a seguir fornecem informações detalhadas sobre como trabalhar com o Amazon Glacier usando a API REST e a API AWS SDKs para Java e Microsoft.NET:

- Usando o AWS SDKs com o Amazon Glacier

  Esta seção fornece uma visão geral do que AWS SDKs é usado em vários exemplos de código neste guia. Uma revisão desta seção ajudará durante a leitura das seções a seguir. Ele inclui uma visão geral do alto e do baixo nível APIs que eles SDKs oferecem, quando usá-los e etapas comuns para executar os exemplos de código fornecidos neste guia.

- Trabalhar com cofres no Amazon Glacier

  Esta seção apresenta detalhes de diversas operações de cofre, tais como criar um cofre, recuperar metadados, usar trabalhos para recuperar o inventário de cofre e configurar notificações de cofre. Além de usar o console do Amazon Glacier, você pode usá-lo AWS SDKs para várias operações de cofre. Esta seção descreve a API e fornece exemplos de trabalho usando o AWS SDK para Java e AWS SDK para .NET o.

- Trabalhar com arquivos no Amazon Glacier

  Esta seção apresenta detalhes das operações de arquivo, como fazer upload de um arquivo em uma única solicitação ou usar uma operação multipart upload para carregar arquivos grandes em partes. A seção também explica como criar trabalhos para fazer download de arquivos de maneira

assíncrona. Esta seção apresenta exemplos de como usar o AWS SDK para Java e o AWS SDK para .NET.

- Referência de API para Amazon Glacier

  O Amazon Glacier é um RESTful serviço. Esta seção descreve as operações REST, inclusive a sintaxe, e as solicitações de exemplo, além de respostas para todas as operações. As bibliotecas do AWS SDK envolvem essa API, simplificando suas tarefas de programação.

Modelo de dados do Amazon Glacier

Os principais componentes do modelo de dados do Amazon Glacier incluem cofres e arquivos. O Amazon Glacier é um serviço web baseado em REST. Em termos de REST, cofres e arquivos são os recursos. Além disso, o modelo de dados do Amazon Glacier inclui recursos de configuração da notificação. Esses recursos complementam os recursos básicos.

Tópicos

- Cofre
- Arquivo
- Trabalho
- Configuração de notificação

Cofre

No Amazon Glacier, um cofre é um contêiner para armazenar arquivos. Um cofre é semelhante a um bucket do Amazon S3. Ao criar um cofre, você especifica um nome e escolhe Região da AWS onde deseja criar o cofre.

Cada recurso do cofre tem um endereço exclusivo. A forma geral é:

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

Por exemplo, suponha que um cofre (`examplevault`) seja criado na região Oeste dos EUA (Oregon) em sua conta com o ID 111122223333. Esse cofre pode ser endereçado pelo seguinte URI:

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

Veja o que significam os vários componentes do URI:

- `glacier.us-west-2.amazonaws.com` identifica a região Oeste dos EUA (Oregon).
- `111122223333`é o Conta da AWS ID que possui o cofre.
- `vaults` se refere ao conjunto de cofres de propriedade da Conta da AWS.
- `examplevault` identifica um cofre específico no conjunto de cofres.

E Conta da AWS pode criar cofres em qualquer um compatível. Região da AWS Para obter uma lista dos compatíveis Regiões da AWS, consulteAcessar o Amazon Glacier. Em uma região, uma conta deve usar nomes de cofre exclusivos. E Conta da AWS pode criar cofres com o mesmo nome em diferentes regiões.

Você pode armazenar um número ilimitado de arquivos em um cofre. Dependendo das necessidades de negócios ou do aplicativo, você pode armazenar esses arquivos em um cofre ou em vários cofres.

O Amazon Glacier dá suporte a diversas operações de cofre. As operações de cofre são específicas da região. Por exemplo, ao criar um cofre, você faz isso em uma região específica. Quando você solicita uma lista de cofres, você a solicita de uma determinada Região da AWS região, e a lista resultante inclui somente cofres criados nessa região específica.

Arquivo

Um arquivo pode conter quaisquer dados, como uma foto, um vídeo ou um documento. Um arquivo é semelhante a um objeto do Amazon S3 e é a unidade básica de armazenamento no Amazon Glacier. Cada arquivo tem uma ID exclusiva e uma descrição opcional. Você pode especificar esta descrição opcional somente durante o upload de um arquivo. O Amazon Glacier atribui ao arquivo uma ID, que é exclusiva Região da AWS na qual o arquivo é armazenado.

Cada arquivo tem um endereço exclusivo. O formato geral é o seguinte:

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

Este é um URI de exemplo de um arquivo armazenado no cofre `examplevault` da região Oeste dos EUA (Oregon) na conta 111122223333:

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

Você pode armazenar um número ilimitado de arquivos em um cofre.

Trabalho

Um trabalho do Amazon Glacier pode ser iniciado para recuperar um arquivo ou obter um inventário de um cofre.

Recuperar um arquivo e um inventário de cofre (lista de arquivos) são operações assíncronas no Glacier nas quais deve-se primeiro iniciar um trabalho e, então, fazer download da saída do trabalho depois que Amazon Glacier concluir o trabalho.

> **Note**
>
> O Amazon Glacier oferece uma solução de arquivamento de dados de armazenamento a frio. Se o aplicativo precisar de uma solução de armazenamento que exija recuperação de dados em tempo real, você poderá considerar o uso do Amazon S3. Para obter mais informações, consulte Amazon Simple Storage Service (Amazon S3).

Para iniciar um trabalho de inventário de cofre, você fornece um nome de cofre. Os trabalhos de seleção e de recuperação de arquivo exigem o nome do cofre e o ID do arquivo. Você também pode fornecer uma descrição de trabalho opcional para ajudar a identificar os trabalhos.

Os trabalhos de recuperação de arquivo e inventário de cofre estão associados a um cofre. Um cofre pode ter vários trabalhos em andamento a qualquer momento. Quando envia-se uma solicitação de trabalho (iniciar um trabalho), o Amazon Glacier retorna um ID do trabalho para rastreá-lo. Cada trabalho é identificado com exclusividade por um URI da forma:

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

Veja a seguir um exemplo de um trabalho associado a um cofre `examplevault` na região Oeste dos EUA (Oregon) na conta 111122223333.

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

Para cada trabalho, o Amazon Glacier mantém informações como tipo de trabalho, descrição, data de criação, data de conclusão e status do trabalho. Você pode obter informações sobre um trabalho

específico ou uma lista de todos os trabalhos associados a um cofre. A lista de trabalhos retornados pelo Amazon Glacier inclui todos os trabalhos em andamento e concluídos recentemente.

Configuração de notificação

Como os trabalhos demoram para serem concluídos, o Amazon Glacier dá suporte a um mecanismo de notificação para notificá-lo quando um trabalho for concluído. Você pode configurar um cofre para enviar uma notificação para um tópico do Amazon Simple Notification Service (Amazon SNS) quando um trabalho é concluído. É possível especificar um tópico do SNS por cofre na configuração de notificação.

O Amazon Glacier armazena a configuração de notificação como um documento JSON. Este é um exemplo de configuração de notificação de cofre:

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

As configurações de notificação são associadas a cofres; pode-se ter uma para cada cofre. Cada recurso de configuração de notificação é identificado com exclusividade por um URI da forma:

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

O Amazon Glacier dá suporte a operações para definir, obter e excluir uma configuração de notificação. Quando se exclui uma configuração de notificação, nenhuma notificação é enviada quando qualquer operação de recuperação de dados no cofre é concluída.

Operações compatíveis no Amazon Glacier

Para trabalhar com cofres e arquivos (consulte Modelo de dados do Amazon Glacier), o Amazon Glacier oferece suporte a um conjunto de operações. Dentre todas as operações compatíveis, somente as seguintes operações são assíncronas:

- Recuperar um arquivo
- Recuperar um inventário de cofre (lista de arquivos)

Essas operações exigem que você inicie primeiro um trabalho e, em seguida, faça download da saída do trabalho. As seguintes seções resumem as operações do Amazon Glacier.

Operações de cofre

O Amazon Glacier oferece operações para criar e excluir cofres. Você pode obter uma descrição do cofre específico ou de todos os cofres em uma Região da AWS. A descrição do cofre fornece informações, como data de criação, número de arquivos no cofre, tamanho total em bytes usado por todos os arquivos no cofre e a data em que o Amazon Glacier gerou o inventário do cofre. O Amazon Glacier também oferece operações para definir, recuperar e excluir uma configuração de notificação no cofre. Para obter mais informações, consulte Trabalhar com cofres no Amazon Glacier.

Operações de arquivo

O Amazon Glacier oferece operações para fazer upload e excluir arquivos. Você não pode atualizar um arquivo existente; você deve excluir o arquivo existente e fazer upload de um novo arquivo. Sempre que é feito upload de um arquivo, o Amazon Glacier gera um novo ID de arquivo. Para obter mais informações, consulte Trabalhar com arquivos no Amazon Glacier.

Tarefas

Pode-se iniciar um trabalho do Amazon Glacier para recuperar um arquivo ou obter um inventário de um cofre.

Estes são os tipos de trabalhos do Amazon Glacier:

- `archive-retrieval`: recupera um arquivo.

  Para obter mais informações, consulte Baixando um arquivo no Amazon Glacier.
- `inventory-retrieval`: inventário de um cofre.

  Para obter mais informações, consulte Fazer download de um inventário de cofre no Amazon Glacier.

Acessar o Amazon Glacier

O Amazon Glacier é um serviço RESTful web que usa HTTP e HTTPS como protocolo de transporte e JavaScript Object Notation (JSON) como formato de serialização de mensagens. Seu código de aplicativo pode fazer solicitações diretamente à API do serviço web do Amazon Glacier. Quando usar

a API REST diretamente, você deverá gravar o código necessário para assinar e autenticar suas solicitações. Para obter mais informações sobre a API, consulte Referência de API para Amazon Glacier.

Como alternativa, você pode simplificar o desenvolvimento de aplicativos usando o AWS SDKs que agrupa as chamadas da API REST do Amazon Glacier. Você fornece suas credenciais, e essas bibliotecas cuidam da autenticação e da assinatura das solicitações. Para obter mais informações sobre como usar o AWS SDKs, consulte Usando o AWS SDKs com o Amazon Glacier.

O Amazon Glacier também fornece um console. No entanto, todas as operações de arquivamento e trabalho exigem que você escreva código e faça solicitações usando diretamente a API REST ou as bibliotecas de wrapper do AWS SDK. Para acessar o console do Amazon Glacier, acesse o console do Amazon Glacier.

Regiões e endpoints

Você cria um cofre em um local específico Região da AWS. Sempre se envia suas solicitações do Amazon Glacier para um endpoint específico de uma Região da AWS. Para obter uma lista das Regiões da AWS compatíveis com o Amazon Glacier, consulte Endpoints e cotas do Amazon Glacier na Referência geral da AWS .

Conceitos básicos do Amazon Glacier

Você pode começar a usar o Amazon Glacier trabalhando com cofres e arquivos. Um cofre é um contêiner para armazenar arquivos, e um arquivo é qualquer objeto, tais como uma foto, um vídeo ou um documento que você armazena em um cofre. Um arquivo é a unidade básica de armazenamento no Amazon Glacier. Este exercício de conceitos básicos fornece instruções para você explorar as operações básicas do Amazon Glacier nos cofres e nos recursos de arquivo. Para obter mais informações sobre tais recursos, consulte a seção Modelo de dados do Amazon Glacier.

No exercício de conceitos básicos, você vai criar um cofre, fazer upload e download de um arquivo e, por fim, excluir o arquivo e o cofre. Você pode fazer todas essas operações de maneira programática. No entanto, o exercício de conceitos básicos usa o console de gerenciamento do Amazon Glacier para criar e excluir um cofre. Para carregar e baixar um arquivo, esta seção de introdução usa a API de alto nível para AWS SDK para Java o. AWS SDK para .NET A API de nível superior proporciona uma experiência de programação simplificada ao trabalhar com o Amazon Glacier. Para obter mais informações sobre como usar a API de alto nível com o AWS SDKs, consulteUsando o AWS SDKs com o Amazon Glacier.

> **Important**
>
> O Amazon Glacier fornece um console. No entanto, qualquer operação de arquivamento, como upload, download ou exclusão, exige que você use a AWS Command Line Interface (CLI) ou escreva código. Não há suporte ao console para operações de arquivo. Por exemplo, para fazer upload de dados, como fotos, vídeos e outros documentos, você deve usar o código AWS CLI ou escrever o código para fazer solicitações, usando a API REST diretamente ou usando AWS SDKs o.
> Para instalar o AWS CLI, consulte AWS Command Line Interface. Para obter mais informações sobre o uso do Amazon Glacier com o AWS CLI, consulte a AWS CLI Referência do Amazon Glacier. Para exemplos de uso do AWS CLI para fazer upload de arquivos para o Amazon Glacier, consulte Usando o Amazon Glacier com o. AWS Command Line Interface

Este exercício de conceitos básicos oferece exemplos de código em Java e C # para você fazer upload e download de um arquivo. A última seção do exercício de introdução fornece etapas nas quais você pode aprender mais sobre a experiência do desenvolvedor com o Amazon Glacier.

Tópicos

- Etapa 1: antes de começar com o Amazon Glacier
- Etapa 2: criar um cofre no Amazon Glacier
- Etapa 3: fazer upload de um arquivo para um cofre no Amazon Glacier
- Etapa 4: fazer download de um arquivo desde um cofre no Amazon Glacier
- Etapa 5: excluir um arquivo de um cofre no Amazon Glacier
- Etapa 6: excluir um cofre no Amazon Glacier
- Para onde ir agora?

Etapa 1: antes de começar com o Amazon Glacier

Antes de começar com este exercício, você deve se inscrever em um Conta da AWS (se ainda não tiver um) e, em seguida, baixar um dos AWS SDKs. Para obter as instruções, consulte a seguinte seção.

Tópicos

- Configurar um usuário Conta da AWS e um administrador
- Baixe o AWS SDK apropriado

Configurar um usuário Conta da AWS e um administrador

Se ainda não tiver feito isso, você deve se inscrever Conta da AWS e criar um usuário administrador na conta.

Para concluir a configuração, siga as instruções nos seguintes tópicos:

Configurar Conta da AWS e criar um usuário administrador

Inscreva-se para AWS

Quando você se inscreve no Amazon Web Services (AWS), você se inscreve automaticamente em todos os serviços AWS, incluindo o Amazon Glacier. Conta da AWS A cobrança incorrerá apenas pelos serviços utilizados. Consulte mais informações sobre as taxas de uso do Amazon Glacier na página de preços do Amazon Glacier.

Se você já tem um Conta da AWS, vá paraBaixe o AWS SDK apropriado. Se você não tiver um Conta da AWS, use o procedimento a seguir para criar um.

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra a https://portal.aws.amazon.com/billing/inscrição.
2. Siga as instruções online.

   Parte do procedimento de inscrição envolve receber uma chamada telefônica ou uma mensagem de texto e inserir um código de verificação pelo teclado do telefone.

   Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWSé criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar tarefas que exigem acesso de usuário-raiz.

Para criar um usuário administrador, selecione uma das opções a seguir.

Selecionar uma forma de gerenciar o administrador	Para	Por	Você também pode
Centro de Identidade do IAM (Recomendado)	Usar credenciais de curto prazo para acessar a AWS. Isso está de acordo com as práticas recomendadas de segurança. Para obter informações sobre as	Seguindo as instruções em Conceitos básicos no Guia do usuário do Centro de Identidade do AWS IAM .	Configure o acesso programático configurando o AWS CLI para uso Centro de Identidade do AWS IAM no Guia do AWS Command Line Interface usuário.

Selecionar uma forma de gerenciar o administrador	Para	Por	Você também pode
	práticas recomendadas, consulte Práticas recomendadas de segurança no IAM no Guia do usuário do IAM.		
No IAM (Não recomendado)	Usar credenciais de longo prazo para acessar a AWS.	Seguindo as instruções em Criar um acesso de emergência para um usuário do IAM no Guia do usuário do IAM.	Configurar o acesso programático, com base em Gerenciar chaves de acesso para usuários do IAM no Guia do usuário do IAM.

Baixe o AWS SDK apropriado

Para experimentar o exercício de introdução, você deve decidir qual linguagem de programação deseja usar e, em seguida, baixar o AWS SDK apropriado para sua plataforma de desenvolvimento.

O exercício de conceitos básicos oferece exemplos em Java e C#.

Baixando o AWS SDK para Java

Para testar os exemplos do Java neste guia do desenvolvedor, você precisa do AWS SDK para Java. Você tem as seguintes opções de download:

- Se estiver usando o Eclipse, você pode baixar e instalar o AWS Toolkit for Eclipse usando o site de atualização http://aws.amazon.com/eclipse/. Para obter mais informações, consulte AWS Toolkit for Eclipse.

- Se você estiver usando qualquer outro IDE para criar o aplicativo, faça download do AWS SDK para Java.

Baixando o AWS SDK para .NET

Para testar os exemplos do C# neste guia do desenvolvedor, você precisa do AWS SDK para .NET. Você tem as seguintes opções de download:

- Se você estiver usando o Visual Studio, poderá instalar o AWS SDK para .NET e AWS Toolkit for Visual Studio o. O kit de ferramentas fornece o AWS Explorer for Visual Studio e modelos de projeto que você pode usar para desenvolvimento. Para baixar o AWS SDK para .NET, acesse http://aws.amazon.com/sdkfornet. Por padrão, o script de instalação instala o AWS SDK e o. AWS Toolkit for Visual Studio Para saber mais sobre as ferramentas do kit, consulte AWS Toolkit for Visual Studio Guia do usuário.
- Se estiver usando qualquer outro IDE para criar o aplicativo, você poderá usar o mesmo link fornecido na etapa anterior e instalar somente o AWS SDK para .NET.

Etapa 2: criar um cofre no Amazon Glacier

Cofre é um contêiner para armazenar arquivos. Sua primeira etapa é criar um cofre em um dos suportados Regiões da AWS. Para obter uma lista das Regiões da AWS que são suportadas pelo Amazon Glacier, consulte os endpoints e cotas do Amazon Glacier na Referência geral.AWS

Você pode criar cofres de maneira programática ou usando o console do Amazon Glacier. Esta seção usa o console para criar um cofre.

Para criar um cofre

1. Faça login Console de gerenciamento da AWS e abra o console do Amazon Glacier em casahttps://console.aws.amazon.com/glacier/.
2. No painel de navegação à esquerda, selecione Cofres.
3. Escolha Criar cofre.

   A página Criar cofre é aberta.
4. Em Selecionar uma região, selecione uma no seletor Região da AWS de região. Seu cofre estará localizado na região que você selecionar.
5. Em Nome do cofre, insira um nome para o seu cofre.

Estes são os requisitos de nomenclatura do cofre:

- O nome do cofre deve ser exclusivo dentro de um Conta da AWS e Região da AWS no qual o cofre foi criado.
- Nomes de atributos devem ter entre 1 e 255 caracteres de comprimento.
- O nome de um cofre pode conter somente os seguintes caracteres: a–z, A–Z, 0–9, _ (sublinhado), - (hífen) e . (ponto).

6. Em Notificações de eventos, para ativar ou desativar as notificações em um cofre para quando um trabalho for concluído, escolha uma das seguintes configurações:

   - Desativar notificações: as notificações são desativadas e não são enviadas para um tópico do Amazon Simple Notification Service (Amazon SNS) quando um trabalho específico é concluído.

   - Ativar notificações: as notificações são ativadas e enviadas para o tópico do Amazon SNS fornecido quando um trabalho específico é concluído.

     Se você escolheu Ativar notificações, consulte Configurar as notificações de cofre usando o console do Amazon Glacier.

7. Se o nome do cofre Região da AWS e estiver correto, escolha Criar cofre.

O novo cofre é listado na página Cofres no console do Amazon Glacier.

Etapa 3: fazer upload de um arquivo para um cofre no Amazon Glacier

Nesta etapa, você fará upload de um arquivo de exemplo para o cofre criado na etapa anterior (consulte Etapa 2: criar um cofre no Amazon Glacier). Dependendo da plataforma de desenvolvimento que você esteja usando, clique em um dos links ao final desta seção.

> **Important**
>
> Qualquer operação de arquivo, como upload, download, ou exclusão, exige que você use a AWS Command Line Interface (CLI) ou escreva um código. Não há suporte ao console para operações de arquivo. Por exemplo, para fazer upload de dados, como fotos, vídeos e outros

documentos, você deve usar o código AWS CLI ou escrever o código para fazer solicitações, usando a API REST diretamente ou usando AWS SDKs o.
Para instalar o AWS CLI, consulte AWS Command Line Interface. Para obter mais informações sobre o uso do Amazon Glacier com o AWS CLI, consulte AWS CLI Referência do Amazon Glacier. Para exemplos de uso do AWS CLI para fazer upload de arquivos para o Amazon Glacier, consulte Usando o Amazon Glacier com o. AWS Command Line Interface

Um arquivo é qualquer objeto, como uma foto, um vídeo ou um documento, armazenado por você em um cofre. Um arquivo é a unidade básica de armazenamento no Amazon Glacier. Você pode fazer upload de um arquivo em uma única solicitação. Para arquivos grandes, o Amazon Glacier fornece uma API de upload multiparte que permite fazer upload de um arquivo em partes.

Nesta seção de conceitos básicos, você faz upload de um arquivo de amostra em uma única solicitação. Para este exercício, você especifica um arquivo menor. Para arquivos maiores, multipart upload é indicado. Para obter mais informações, consulte Fazer upload de arquivos grandes em partes (Multipart Upload).

Tópicos

- Faça upload de um arquivo para um cofre no Amazon Glacier usando o AWS SDK para Java
- Faça upload de um arquivo para um cofre no Amazon Glacier usando o AWS SDK para .NET

Faça upload de um arquivo para um cofre no Amazon Glacier usando o AWS SDK para Java

O exemplo de código Java a seguir usa a API de alto nível do AWS SDK para Java para carregar um arquivo de amostra no cofre. No exemplo de código, observe o seguinte:

- O exemplo cria uma instância da classe `AmazonGlacierClient`.
- O exemplo usa o método `upload` da classe `ArchiveTransferManager` da API de nível superior do AWS SDK para Java.
- Esse exemplo usa a região Oeste dos EUA (Oregon) `us-west-2`.

Para step-by-step obter instruções sobre como executar esse exemplo, consulteExecutar exemplos do Java para o Amazon Glacier usando o Eclipse. Você precisa atualizar o código conforme mostrado com o nome do arquivo cujo upload deseja fazer.

> **Note**
>
> O Amazon Glacier mantém um inventário de todos os arquivos nos cofres. Quando você fizer upload do arquivo no exemplo a seguir, ele não será exibido em um cofre no console de gerenciamento até o inventário de cofre ter sido atualizado. Essa atualização normalmente acontece uma vez por dia.

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
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();
    }
}
```

- Para obter detalhes da API, consulte UploadArchivea Referência AWS SDK for Java 2.x da API.

Faça upload de um arquivo para um cofre no Amazon Glacier usando o AWS SDK para .NET

O exemplo de código C# a seguir usa a API de alto nível do AWS SDK para .NET para fazer upload de um arquivo de amostra para o cofre. No exemplo de código, observe o seguinte:

- O exemplo cria uma instância da classe `ArchiveTransferManager` para o endpoint especificado da região do Amazon Glacier.
- Esse exemplo de código usa a região Oeste dos EUA (Oregon) `us-west-2`.
- O exemplo usa a operação `Upload` da classe `ArchiveTransferManager` para fazer upload do arquivo. Para arquivos pequenos, esse método faz upload do arquivo diretamente no Amazon Glacier. Para arquivos maiores, esta operação usa a operação de API de upload multiparte no Amazon Glacier para dividir o upload multiparte para melhor recuperação de erros, se algum erro for encontrado durante o streaming dos dados para o Amazon Glacier.

Para step-by-step obter instruções sobre como executar o exemplo a seguir, consulteExecutar exemplos de código. Você precisa atualizar o código conforme mostrado com o nome do cofre e o nome do arquivo do qual deve ser feito upload.

> **Note**
>
> O Amazon Glacier mantém um inventário de todos os arquivos nos cofres. Quando você fizer upload do arquivo no exemplo a seguir, ele não será exibido em um cofre no console de gerenciamento até o inventário de cofre ter sido atualizado. Essa atualização normalmente acontece uma vez por dia.

Example— Carregar um arquivo usando a API de alto nível do AWS SDK para .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();
        }
    }
}
```

Etapa 4: fazer download de um arquivo desde um cofre no Amazon Glacier

Nesta etapa, você fará download do arquivo de exemplo do qual fez upload anteriormente em Etapa 3: fazer upload de um arquivo para um cofre no Amazon Glacier.

> **Important**
>
> O Amazon Glacier fornece um console. No entanto, qualquer operação de arquivamento, como upload, download ou exclusão, exige que você use a AWS Command Line Interface (CLI) ou escreva código. Não há suporte ao console para operações de arquivo. Por exemplo, para fazer upload de dados, como fotos, vídeos e outros documentos, você deve usar o código AWS CLI ou escrever o código para fazer solicitações, usando a API REST diretamente ou usando AWS SDKs o.
> Para instalar o AWS CLI, consulte AWS Command Line Interface. Para obter mais informações sobre o uso do Amazon Glacier com o AWS CLI, consulte AWS CLI Referência do Amazon Glacier. Para exemplos de uso do AWS CLI para fazer upload de arquivos para o Amazon Glacier, consulte Usando o Amazon Glacier com o. AWS Command Line Interface

Em geral, recuperar os dados do Amazon Glacier é um processo de duas etapas:

1. Inicie um trabalho de recuperação.
2. Depois que o trabalho for concluído, faça download dos bytes de dados.

Para recuperar um arquivo do Amazon Glacier, você primeiro inicia um trabalho. Depois que o trabalho for concluído, você fará download dos dados. Para obter mais informações sobre recuperações de arquivo, consulte Recuperação de arquivos do Amazon Glacier.

O tempo de acesso da solicitação depende da opção de recuperação escolhida por você: expressa, padrão ou em massa. Exceto para os arquivos maiores (mais de 250 MB), os dados acessados usando-se recuperações expressas costumam ser disponibilizados dentro de 1 a 5 minutos. Arquivos recuperados usando-se recuperações padrão normalmente são concluídos entre 3 e 5 horas. As recuperações em massa normalmente são concluídas dentro de 5 a 12 horas. Para obter mais informações sobre as opções de recuperação, consulte as Perguntas Frequentes do Amazon

Glacier. Para obter mais informações sobre cobranças pela recuperação de dados, consulte a Página de Detalhes do Amazon Glacier.

Os exemplos de código mostrados nos tópicos a seguir iniciam o trabalho, aguardam a conclusão e fazem download dos dados de arquivo.

Tópicos

- Baixe um arquivo de um cofre no Amazon Glacier usando o AWS SDK para Java
- Baixe um arquivo de um cofre no Amazon Glacier usando o AWS SDK para .NET

Baixe um arquivo de um cofre no Amazon Glacier usando o AWS SDK para Java

O exemplo de código Java a seguir usa a API de alto nível do AWS SDK para Java para baixar o arquivo que você carregou na etapa anterior. No exemplo de código, observe o seguinte:

- O exemplo cria uma instância da classe `AmazonGlacierClient`.
- O código usa a região Oeste dos EUA (`us-west-2`) de acordo com o local onde você criou o cofre em Etapa 2: criar um cofre no Amazon Glacier.
- O exemplo usa `download` operação API da classe `ArchiveTransferManager` oriundo de API de nível superior do AWS SDK para Java. O exemplo cria um tópico do Amazon Simple Notification Service (Amazon SNS) e uma fila do Amazon Simple Queue Service (Amazon SQS) que está inscrita nesse tópico. Se você criou um usuário administrador AWS Identity and Access Management (IAM) conforme instruído emEtapa 1: antes de começar com o Amazon Glacier, seu usuário tem as permissões IAM necessárias para a criação e o uso do tópico do Amazon SNS e da fila do Amazon SQS.

Para step-by-step obter instruções sobre como executar esse exemplo, consulteExecutar exemplos do Java para o Amazon Glacier usando o Eclipse. Você deve atualizar o código conforme mostrado com o ID do arquivo que você carregou em Etapa 3: fazer upload de um arquivo para um cofre no Amazon Glacier.

Example— Baixar um arquivo usando o AWS SDK para 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);
        }
    }
}
```

Baixe um arquivo de um cofre no Amazon Glacier usando o AWS SDK para .NET

O exemplo de código C# a seguir usa a API de alto nível do AWS SDK para .NET para baixar o arquivo que você carregou anteriormente. Faça upload de um arquivo para um cofre no Amazon Glacier usando o AWS SDK para .NET No exemplo de código, observe o seguinte:

- O exemplo cria uma instância da classe `ArchiveTransferManager` para o endpoint especificado da região do Amazon Glacier.
- O exemplo de código usa a Região Oeste dos EUA (Oregon) (`us-west-2`) de acordo com o local onde você previamente criou o cofre.
- O exemplo usa a operação `Download` API da classe `ArchiveTransferManager` para baixar seu arquivo. O exemplo cria um tópico do Amazon Simple Notification Service (Amazon SNS) e uma fila do Amazon Simple Queue Service (Amazon SQS) que está inscrita nesse tópico. Se você criou um usuário administrador AWS Identity and Access Management (IAM) conforme instruído emEtapa 1: antes de começar com o Amazon Glacier, seu usuário tem as permissões IAM necessárias para a criação e o uso do tópico do Amazon SNS e da fila do Amazon SQS.
- Em seguida, o exemplo inicia o trabalho de recuperação do arquivo e sonda a fila em busca do arquivo disponível. Assim que o arquivo estiver disponível, o download começará. Para obter informações sobre tempos de recuperação, consulte Opções de recuperação de arquivos.

Para step-by-step obter instruções sobre como executar esse exemplo, consulteExecutar exemplos de código. Você deve atualizar o código conforme mostrado com o ID do arquivo que você carregou em Etapa 3: fazer upload de um arquivo para um cofre no Amazon Glacier.

Example— Baixe um arquivo usando a API de alto nível do AWS SDK para .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);
            }
        }
    }
}
```

Etapa 5: excluir um arquivo de um cofre no Amazon Glacier

Nesta etapa, excluirá o arquivo de exemplo o qual você fez upload em Etapa 3: fazer upload de um arquivo para um cofre no Amazon Glacier.

> **Important**
>
> Você não pode excluir um arquivo usando o console do Amazon Glacier. Qualquer operação de arquivamento, como upload, download ou exclusão, exige que você use a AWS Command Line Interface (CLI) ou escreva código. Para fazer upload de dados, como fotos, vídeos e outros documentos, você deve usar o código AWS CLI ou escrever o código para fazer solicitações, usando diretamente a API REST ou usando AWS SDKs o.
> Para instalar o AWS CLI, consulte AWS Command Line Interface. Para obter mais informações sobre o uso do Amazon Glacier com o AWS CLI, consulte AWS CLI Referência do Amazon Glacier. Para exemplos de uso do AWS CLI para fazer upload de arquivos para o Amazon Glacier, consulte Usando o Amazon Glacier com o. AWS Command Line Interface

Exclua o arquivo de amostra seguindo um destes SDKs ou o seguinte AWS CLI:

- Exclua um arquivo de um cofre no Amazon Glacier usando o AWS SDK para Java
- Exclua um arquivo de um cofre no Amazon Glacier usando o AWS SDK para .NET
- Exclua um arquivo no Amazon Glacier usando o AWS CLI

Seções relacionadas

- Etapa 3: fazer upload de um arquivo para um cofre no Amazon Glacier
- Excluir um arquivo no Amazon Glacier

Exclua um arquivo de um cofre no Amazon Glacier usando o AWS SDK para Java

O exemplo de código a seguir usa o AWS SDK para Java para excluir o arquivo. No código, observe o seguinte:

- O objeto `DeleteArchiveRequest` descreve a solicitação de exclusão, inclusive o nome do cofre em que o arquivo está localizado e o ID do arquivo.
- A operação API `deleteArchive` envia a solicitação ao Amazon Glacier para excluir o arquivo.
- Esse exemplo usa a região Oeste dos EUA (Oregon) `us-west-2`.

Para step-by-step obter instruções sobre como executar esse exemplo, consulteExecutar exemplos do Java para o Amazon Glacier usando o Eclipse. Você deve atualizar o código conforme mostrado com o ID do arquivo que você carregou em Etapa 3: fazer upload de um arquivo para um cofre no Amazon Glacier.

Example— Excluindo um arquivo usando o AWS SDK para 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);
        }
    }
}
```

Exclua um arquivo de um cofre no Amazon Glacier usando o AWS SDK para .NET

O exemplo de código C# a seguir usa a API de alto nível do AWS SDK para .NET para excluir o arquivo que você carregou na etapa anterior. No exemplo de código, observe o seguinte:

- O exemplo cria uma instância da classe `ArchiveTransferManager` para o endpoint especificado da região do Amazon Glacier.
- Esse exemplo de código usa a região Oeste dos EUA (Oregon) `us-west-2`.
- O exemplo usa operação API `Delete` da classe `ArchiveTransferManager` fornecida como parte da API de nível superior do AWS SDK para .NET.

Para step-by-step obter instruções sobre como executar esse exemplo, consulteExecutar exemplos de código. Você deve atualizar o código conforme mostrado com o ID do arquivo que você carregou em Etapa 3: fazer upload de um arquivo para um cofre no Amazon Glacier.

Example— Excluindo um arquivo usando a API de alto nível do AWS SDK para .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();
    }
  }
}
```

Exclua um arquivo no Amazon Glacier usando o AWS CLI

Você pode excluir arquivos no Amazon Glacier usando o AWS Command Line Interface ()AWS CLI.

Tópicos

- (Pré-requisito) Configurando o AWS CLI
- Exemplo: Excluindo um arquivo usando o AWS CLI

(Pré-requisito) Configurando o AWS CLI

1. Faça download e configure a AWS CLI. Para obter instruções, consulte os seguintes tópicos no Guia do usuário do AWS Command Line Interface :

   Instalando o AWS Command Line Interface

   Configurando o AWS Command Line Interface
2. Verifique sua AWS CLI configuração inserindo os seguintes comandos no prompt de comando. Esses comandos não fornecem as credenciais explicitamente, de modo que as credenciais do perfil padrão são usadas.

   - Tente usar o comando de ajuda.

     ```
     aws help
     ```

   - Para obter uma lista dos cofres do Amazon Glacier na conta configurada, use o comando `list-vaults`. *123456789012*Substitua pelo seu Conta da AWS ID.

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

- Para ver os dados de configuração atuais do AWS CLI, use o `aws configure list` comando.

```
aws configure list
```

Exemplo: Excluindo um arquivo usando o AWS CLI

1. Use o comando `initiate-job` para iniciar um trabalho de recuperação de inventário. Para obter mais informações sobre os comando `initiate-job`, consulte Iniciar Trabalho.

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

Saída esperada:

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

2. Use o comando `describe-job` para verificar o status do trabalho de recuperação anterior. Para obter mais informações sobre o comando `describe-job`, consulte Descrever Trabalho.

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

Saída esperada:

```
{
    "InventoryRetrievalParameters": {
        "Format": "JSON"
    },
    "VaultARN": "*** vault arn ***",
    "Completed": false,
    "JobId": "*** jobid ***",
    "Action": "InventoryRetrieval",
```

```
    "CreationDate": "*** job creation date ***",
    "StatusCode": "InProgress"
}
```

3. Aguarde a conclusão do trabalho.

   Você deve aguardar até a saída do trabalho estar pronta para download. Se você definiu uma configuração de notificação no cofre ou especificou um tópico do Amazon Simple Notification Service (Amazon SNS) ao iniciar o trabalho, o Amazon Glacier envia uma mensagem para o tópico depois de concluir o trabalho.

   É possível definir a configuração de notificação para eventos específicos no cofre. Para obter mais informações, consulte Configurar notificações do cofre no Amazon Glacier. O Amazon Glacier enviará uma mensagem para o tópico do SNS especificado sempre que o evento específico ocorrer.

4. Quando estiver concluído, use o comando `get-job-output` para fazer download do trabalho de recuperação para o arquivo `output.json`. Para obter mais informações sobre o comando `get-job-output`, consulte Obter saída do Trabalho.

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

   Esse comando produz um arquivo com os campos a seguir.

   ```
   {
   "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. Use o comando `delete-archive` para excluir cada arquivamento de um cofre até que não haja mais nenhum.

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

Para obter mais informações sobre o comando `delete-archive`, consulte Excluir arquivo.

Etapa 6: excluir um cofre no Amazon Glacier

Cofre é um contêiner para armazenar arquivos. Para excluir um cofre do Amazon Glacier, é necessário primeiro excluir todos os arquivos existentes no cofre desde o inventário computado mais recentemente pelo Amazon Glacier.

Você pode excluir um cofre de maneira programática ou usando o console do Amazon Glacier. Para obter informações sobre como excluir um cofre de maneira programática, consulte Excluir um cofre no Amazon Glacier.

> **Important**
>
> Se você carregou um arquivo em um cofre ou excluiu um arquivo de um cofre nas últimas 24 horas, deverá esperar até que o último inventário do cofre seja atualizado para refletir as informações mais recentes. O Amazon Glacier prepara um inventário para cada cofre periodicamente, a cada 24 horas.

Como excluir um cofre vazio

1. Faça login Console de gerenciamento da AWS e abra o console do Amazon Glacier em casahttps://console.aws.amazon.com/glacier/.
2. No menu Selecionar uma região, escolha Região da AWS o cofre que você deseja excluir.

   Neste exercício de introdução, seu exemplo de cofre está na região Oeste dos EUA (Oregon).
3. Selecione o botão de opção ao lado do cofre vazio que você deseja excluir. Se o cofre não estiver vazio, será necessário excluir todos os arquivos antes de excluir o cofre. Para obter mais informações, consulte Excluir um arquivo no Amazon Glacier.

> **Important**
>
> A exclusão de um cofre não pode ser desfeita.

4. Escolha Excluir.
5. A caixa de diálogo Excluir cofre é exibida. Escolha Excluir.

Como excluir um cofre que não está vazio

1. Se for excluir um cofre que não está vazio, será necessário primeiro excluir todos os arquivos existentes antes de excluir o cofre. Você pode fazer isso escrevendo um código para fazer uma solicitação de exclusão de arquivamento usando a API REST AWS SDK para Java, a, a AWS SDK para .NET ou AWS CLI a. Para obter informações sobre como excluir arquivos, consulte Etapa 5: excluir um arquivo de um cofre no Amazon Glacier.
2. Depois que o cofre estiver vazio, siga as etapas para excluir um cofre vazio conforme mostrado no procedimento anterior.

Para onde ir agora?

Agora que você fez o exercício de conceitos básicos, é possível explorar as seções a seguir para saber mais sobre o Amazon Glacier.

- Trabalhar com cofres no Amazon Glacier
- Trabalhar com arquivos no Amazon Glacier

Trabalhar com cofres no Amazon Glacier

Cofre é um contêiner para armazenar arquivos. Ao criar um cofre, você especifica um nome de cofre e a Região da AWS na qual deseja criar o cofre. Para obter uma lista das Regiões da AWS compatíveis com o Amazon Glacier, consulte Endpoints e cotas do Amazon Glacier na Referência geral da AWS.

Você pode armazenar um número ilimitado de arquivos em um cofre.

> **Important**
>
> O Amazon Glacier fornece um console. No entanto, qualquer operação de arquivo, como upload, download ou exclusão, exige que você use a AWS Command Line Interface (AWS CLI) ou escreva um código. Não há suporte ao console para operações de arquivo. Por exemplo, para fazer upload de dados, como fotos, vídeos e outros documentos, você deve usar a AWS CLI ou escrever código para fazer solicitações usando a API REST diretamente ou usando os AWS SDKs.
>
> Para instalar a AWS CLI, consulte AWS Command Line Interface. Para obter mais informações sobre o uso do Amazon Glacier com a AWS CLI, consulte a Referência da AWS CLI para o Amazon Glacier. Para obter exemplos de como usar a AWS CLI para carregar arquivos no Amazon Glacier, consulte Usar o Amazon Glacier com a AWS Command Line Interface.

Tópicos

- Operações de cofre no Amazon Glacier
- Criar um cofre no Amazon Glacier
- Recuperar metadados do cofre no Amazon Glacier
- Fazer download de um inventário de cofre no Amazon Glacier
- Configurar notificações do cofre no Amazon Glacier
- Excluir um cofre no Amazon Glacier
- Marcar seus Cofres do Amazon Glacier
- Vault Lock do Amazon Glacier

Operações de cofre no Amazon Glacier

O Amazon Glacier dá suporte a diversas operações de cofre. As operações de cofre são específicas para Regiões da AWS em particular. Em outras palavras, ao criar um cofre, ele é criado em uma Região da AWS específica. Quando você lista cofres, o Amazon Glacier retorna a lista de cofres da Região da AWS especificada na solicitação.

Criar e excluir cofres

Uma Conta da AWS pode criar até 1.000 cofre por Região da AWS. Para obter uma lista das Regiões da AWS compatíveis com o Amazon Glacier, consulte Endpoints e cotas do Amazon Glacier na Referência geral da AWS.

Você poderá excluir um cofre somente se não houver arquivos nele desde que o último inventário foi computado pelo Amazon Glacier e se não houver gravações no cofre desde o último inventário.

> **Note**
>
> O Amazon Glacier prepara um inventário para cada cofre periodicamente, a cada 24 horas. Como o inventário talvez não reflita as informações mais recentes, o Amazon Glacier garante que o cofre esteja realmente vazio verificando se houve alguma operação de gravação desde o último inventário.

Para obter mais informações, consulte Criar um cofre no Amazon Glacier e Excluir um cofre no Amazon Glacier.

Recuperar metadados do cofre

Você pode recuperar informações do cofre, como a data de criação do cofre, o número de arquivos no cofre e o tamanho total de todos os arquivos no cofre. O Amazon Glacier fornece chamadas de API para que você recupere essas informações de um cofre específico ou de todos os cofres em determinada Região da AWS na conta. Para obter mais informações, consulte Recuperar metadados do cofre no Amazon Glacier.

Fazer download de um inventário de cofre

Um inventário de cofre se refere à lista de arquivos no cofre. Para cada arquivo na lista, o inventário fornece informações de arquivo, como ID, data de criação e tamanho. O Amazon Glacier atualiza o

inventário de cofre aproximadamente uma vez por dia, começando no primeiro dia de upload para o cofre. Deve haver um inventário de cofre para que você possa fazer download dele.

Fazer download de um inventário de cofre é uma operação assíncrona. Você deve primeiro iniciar um trabalho para fazer download do inventário. Depois de receber a solicitação de trabalho, o Amazon Glacier prepara o inventário para download. Depois que o trabalho for concluído, você poderá fazer download dos dados de inventário.

Dada a natureza assíncrona do trabalho, você pode usar as notificações do Amazon Simple Notification Service (Amazon SNS) para notificá-lo quando o trabalho for concluído. Você pode especificar um tópico do Amazon SNS para cada solicitação de trabalho individual ou configurar o cofre para enviar uma notificação quando ocorrerem eventos de cofre específicos.

O Amazon Glacier prepara um inventário para cada cofre periodicamente, a cada 24 horas. Se não houver adições ou exclusões de arquivo no cofre desde o último inventário, a data do inventário não será atualizada.

Quando você inicia um trabalho para um inventário de cofre, o Amazon Glacier retorna o último inventário gerado, que é um snapshot point-in-time, e não dados em tempo real. Talvez você não ache útil recuperar um inventário de cofre para cada upload de arquivo. No entanto, suponhamos que você mantenha um banco de dados no lado do cliente que contém metadados associados aos arquivos cujo upload fez para o Amazon Glacier. Em seguida, você talvez ache o inventário de cofre útil para reconciliar informações no banco de dados com o inventário de cofre.

Para obter mais informações sobre como recuperar um inventário de cofre, consulte Fazer download de um inventário de cofre no Amazon Glacier.

Configurar notificações de cofre

Recuperar qualquer dado do Amazon Glacier, como um arquivo de um cofre ou um inventário de cofre, é um processo de duas etapas. Primeiro, você inicia um trabalho. Depois que o trabalho for concluído, você poderá fazer download da saída. Para saber quando seu trabalho foi concluído, você pode usar as notificações do Amazon Glacier. O Amazon Glacier envia mensagens de notificação a um tópico do Amazon Simple Notification Service (Amazon SNS) fornecido por você.

Você pode configurar notificações em um cofre e identificar eventos de cofre, além do tópico do Amazon SNS a ser notificado quando o evento ocorrer. Sempre que o evento de cofre ocorrer, o Amazon Glacier enviará uma notificação para o tópico do Amazon SNS especificado. Para obter mais informações, consulte Configurar notificações do cofre no Amazon Glacier.

Criar um cofre no Amazon Glacier

Criar um cofre adiciona um cofre ao conjunto de cofres na conta. Uma conta da Conta da AWS pode criar até 1.000 cofres por região da AWS. Para obter uma lista das regiões da AWS compatíveis com o Amazon Glacier, consulte Regiões e endpoints na Referência geral da AWS.

Ao criar um cofre, você deve fornecer um nome. Estes são os requisitos de nomenclatura do cofre:

- Os nomes podem ter de 1 a 255 caracteres.
- Os caracteres válidos são a-z, A-Z, 0-9, '_' (sublinhado), '-' (hífen) e '.' (ponto).

Os nomes de cofres devem ser exclusivos dentro de uma conta e da região da AWS na qual o cofre está sendo criado. Ou seja, uma conta pode criar cofres com o mesmo nome em regiões da AWS diferentes, mas não na mesma região da AWS.

Tópicos

- Como criar um cofre no Amazon Glacier usando AWS SDK para Java
- Criação de um cofre no Amazon Glacier usando o AWS SDK para .NET
- Como criar um cofre no Amazon Glacier usando a API REST
- Como criar um cofre usando o console do Amazon Glacier
- Como criar um cofre no Amazon Glacier usando AWS Command Line Interface

Como criar um cofre no Amazon Glacier usando AWS SDK para Java

A API de nível inferior fornece métodos para todas as operações de cofre, incluindo criar e excluir cofres, obter uma descrição de cofre e uma lista de cofres criados em uma Região da AWS específica. Estas são as etapas para criar um cofre usando o console do AWS SDK para Java.

1. Crie uma instância da classe `AmazonGlacierClient` (o cliente).

   Você precisa especificar uma região da Região da AWS na qual deseja criar um cofre. Todas as operações realizadas por você usando esse cliente se aplicam a essa Região da AWS.
2. Forneça informações sobre a solicitação criando uma instância da classe `CreateVaultRequest`.

   O Amazon Glacier exige que você forneça um nome de cofre e o ID da conta. Se você não fornecer um ID da conta, o ID da conta associado às credenciais fornecidas por você para assinar

a solicitação será usado. Para obter mais informações, consulte Usando o AWS SDK para Java com o Amazon Glacier.

3. Execute o método `createVault` fornecendo o objeto de solicitação como um parâmetro.

   A resposta que o Amazon Glacier retorna está disponível no objeto `CreateVaultResult`.

O trecho de código Java a seguir ilustra as etapas anteriores. O trecho cria um cofre na Região `us-west-2`. O `Location` impresso por ele é o URI relativo do cofre que inclui o ID da sua conta, a Região da AWS e o nome do cofre.

```
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**
>
> Para obter informações sobre a API REST subjacente, consulte Create Vault (PUT vault).

Exemplos: como criar um cofre usando a AWS SDK para Java

O exemplo de código do Java a seguir cria um cofre na região `us-west-2` (para obter mais informações sobre Regiões da AWS, consulte Acessar o Amazon Glacier). Além disso, o exemplo de código recupera as informações do cofre, lista todos os cofres na mesma Região da AWS e exclui o cofre criado.

Para obter instruções passo a passo sobre como executar o exemplo a seguir, consulte Executar exemplos do Java para o Amazon Glacier usando o 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);
    }

}
```

Criação de um cofre no Amazon Glacier usando o AWS SDK para .NET

Tanto o alto quanto o baixo nível APIs fornecidos pelo Amazon SDK para .NET fornecem um método para criar um cofre.

Tópicos

- Criando um cofre usando a API de alto nível do AWS SDK para .NET
- Criando um cofre usando a API de baixo nível do AWS SDK para .NET

Criando um cofre usando a API de alto nível do AWS SDK para .NET

A classe `ArchiveTransferManager` da API de nível superior fornece o método `CreateVault` que você pode usar para criar um cofre em uma região da AWS .

Exemplo: operações do Vault usando a API de alto nível do AWS SDK para .NET

O exemplo de código do C# a seguir cria e exclui um cofre na região oeste dos EUA (Oregon). Para obter uma lista Regiões da AWS em que você pode criar cofres, consulte. Acessar o Amazon Glacier

Para step-by-step obter instruções sobre como executar o exemplo a seguir, consulteExecutar exemplos de código. Você precisa atualizar o código conforme mostrado com um nome de cofre.

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();
    }
  }
}
```

Criando um cofre usando a API de baixo nível do AWS SDK para .NET

A API de baixo nível fornece métodos para todas as operações do cofre, incluindo criar e excluir cofres, obter uma descrição do cofre e obter uma lista dos cofres criados em um cofre específico. Região da AWS Estas são as etapas para criar um cofre usando o console do AWS SDK para .NET.

1. Crie uma instância da classe `AmazonGlacierClient` (o cliente).

   Você precisa especificar um Região da AWS no qual deseja criar um cofre. Todas as operações realizadas por você usando esse cliente se aplicam a essa Região da AWS.
2. Forneça informações sobre a solicitação criando uma instância da classe `CreateVaultRequest`.

   O Amazon Glacier exige que você forneça um nome de cofre e o ID da conta. Se você não fornecer um ID da conta, o ID da conta associado às credenciais fornecidas por você para assinar a solicitação será pressuposto. Para obter mais informações, consulte Como usar o AWS SDK para .NET com o Amazon Glacier.
3. Execute o método `CreateVault` fornecendo o objeto de solicitação como um parâmetro.

   A resposta que o Amazon Glacier retorna está disponível no objeto `CreateVaultResponse`.

Exemplo: operações do Vault usando a API de baixo nível do AWS SDK para .NET

O exemplo do C# a seguir ilustra as etapas anteriores. O exemplo cria um cofre na região Oeste dos EUA (Oregon). Além disso, o exemplo de código recupera as informações do cofre, lista todos os

cofres no mesmo cofre e Região da AWS, em seguida, exclui o cofre criado. O `Location` impresso é o URI relativo do cofre que inclui o ID da sua conta Região da AWS, o e o nome do cofre.

> **Note**
>
> Para obter informações sobre a API REST subjacente, consulte Create Vault (PUT vault).

Para step-by-step obter instruções sobre como executar o exemplo a seguir, consulteExecutar exemplos de código. Você precisa atualizar o código conforme mostrado com um nome de cofre.

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);
    }
  }
}
```

Como criar um cofre no Amazon Glacier usando a API REST

Para criar um cofre usando a API REST, consulte Create Vault (PUT vault).

Como criar um cofre usando o console do Amazon Glacier

Para criar um cofre usando o console do Amazon Glacier, consulte Etapa 2: criar um cofre no Amazon Glacier no tutorial Conceitos básicos.

Como criar um cofre no Amazon Glacier usando AWS Command Line Interface

Siga estas etapas para criar um cofre no Amazon Glacier usando AWS Command Line Interface (AWS CLI).

Tópicos

- (Pré-requisito) Configurar a AWS CLI

- Exemplos: criar um cofre usando a AWS CLI

(Pré-requisito) Configurar a AWS CLI

1. Faça download e configure a AWS CLI. Para obter instruções, consulte os seguintes tópicos no Guia do usuário do AWS Command Line Interface:

   Instalar a AWS Command Line Interface

   Configurando o AWS Command Line Interface
2. Verifique a configuração digitando os comandos a seguir no prompt de comando. Esses comandos não fornecem as credenciais explicitamente, de modo que as credenciais do perfil padrão são usadas.

   - Tente usar o comando de ajuda.

     ```
     aws help
     ```

   - Para obter uma lista dos cofres do Amazon Glacier na conta configurada, use o comando `list-vaults`. Substitua *123456789012* pelo ID da sua Conta da AWS.

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

   - Para obter os dados da configuração atual para a AWS CLI, use o comando `aws configure list`.

     ```
     aws configure list
     ```

Exemplos: criar um cofre usando a AWS CLI

1. Use o comando `create-vault` para criar um cofre chamado *awsexamplevault* na conta *111122223333*.

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

   Saída esperada:

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

2. Verifique a criação usando o comando `describe-vault`.

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

Recuperar metadados do cofre no Amazon Glacier

Você pode recuperar informações do cofre, como a data de criação do cofre, o número de arquivos no cofre e o tamanho total de todos os arquivos no cofre. O Amazon Glacier (Amazon Glacier) fornece chamadas de API para você recuperar essas informações para um cofre específico ou para todos os cofres em uma região específica em sua conta. AWS

Se você recuperar uma lista de cofres, o Amazon Glacier retornará a lista classificada pelos valores ASCII dos nomes de cofre. A lista contém até 1.000 cofres. Você deve sempre verificar a resposta para um marcador no qual continuar a lista; se não houver mais itens, o campo do marcador será `null`. Você também pode limitar o número de cofres retornados na resposta. Se houver mais cofres do que os retornados na resposta, o resultado será paginado. Você precisa enviar solicitações adicionais para buscar o próximo conjunto de cofres.

Tópicos

- Recuperação de metadados do Vault no Amazon Glacier usando o AWS SDK para Java
- Recuperação de metadados do Vault no Amazon Glacier usando o AWS SDK para .NET
- Recuperar metadados do cofre usando a API REST
- Recuperar metadados do cofre no Amazon Glacier usando a AWS Command Line Interface

Recuperação de metadados do Vault no Amazon Glacier usando o AWS SDK para Java

Tópicos

- Recuperar metadados de um cofre
- Recuperar metadados de todos os cofres em uma região

- Exemplo: recuperar metadados de cofre usando o Amazon SDK para Java

Recuperar metadados de um cofre

Você pode recuperar metadados de um cofre específico ou de todos os cofres em uma região específica. AWS Estas são as etapas para recuperar metadados de um cofre específico usando a API de nível inferior do Amazon SDK para Java.

1. Crie uma instância da classe `AmazonGlacierClient` (o cliente).

   Você precisa especificar uma AWS região onde o cofre reside. Todas as operações que você executa usando esse cliente se aplicam a essa AWS região.
2. Forneça informações sobre a solicitação criando uma instância da classe `DescribeVaultRequest`.

   O Amazon Glacier exige que você forneça um nome de cofre e o ID da conta. Se você não fornecer um ID da conta, o ID da conta associado às credenciais fornecidas por você para assinar a solicitação será pressuposto. Para obter mais informações, consulte Usando o AWS SDK para Java com o Amazon Glacier.
3. Execute o método `describeVault` fornecendo o objeto de solicitação como um parâmetro.

   As informações de metadados de cofre retornadas pelo Amazon Glacier estão disponíveis no objeto `DescribeVaultResult`.

O trecho de código Java a seguir ilustra as etapas anteriores.

```
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**
>
> Para obter informações sobre a API REST subjacente, consulte Describe Vault (GET vault).

Recuperar metadados de todos os cofres em uma região

Você também pode usar o `listVaults` método para recuperar metadados de todos os cofres em uma região específica. AWS

O trecho de código do Java a seguir recupera uma lista de cofres na região `us-west-2`. A solicitação limita o número de cofres retornados na resposta a 5. O trecho de código então faz uma série de `listVaults` chamadas para recuperar toda a lista de cofres da região. 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);
```

No segmento de código anterior, se você não especificar o valor `Limit` na solicitação, o Amazon Glacier retornará até 10 cofres, conforme definido pela API do Amazon Glacier. Se houver mais

cofres a serem listados, o campo `marker` de resposta conterá o nome de recurso da Amazon (ARN) do cofre no qual deve continuar a lista com uma nova solicitação; do contrário, o campo `marker` será nulo.

As informações retornadas para cada cofre na lista são as mesmas obtidas por você chamando o método `describeVault` para um cofre específico.

> **Note**
>
> O método `listVaults` chama a API REST subjacente (consulte List Vaults (GET vaults)).

Exemplo: recuperar metadados de cofre usando o Amazon SDK para Java

Para obter um exemplo de código funcional, consulte Exemplos: como criar um cofre usando a AWS SDK para Java. O exemplo de código do Java cria um cofre e recupera os metadados do cofre.

Recuperação de metadados do Vault no Amazon Glacier usando o AWS SDK para .NET

Tópicos

- Recuperar metadados de um cofre
- Recuperar metadados de todos os cofres em uma região
- Exemplo: Recuperação de metadados do Vault usando a API de baixo nível do AWS SDK para .NET

Recuperar metadados de um cofre

Você pode recuperar metadados de um cofre específico ou de todos os cofres em uma região específica. AWS Estas são as etapas para recuperar metadados de um cofre específico usando a API de nível inferior do AWS SDK para .NET.

1. Crie uma instância da classe `AmazonGlacierClient` (o cliente).

   Você precisa especificar uma AWS região onde o cofre reside. Todas as operações que você executa usando esse cliente se aplicam a essa AWS região.

2. Forneça informações sobre a solicitação criando uma instância da classe `DescribeVaultRequest`.

   O Amazon Glacier exige que você forneça um nome de cofre e o ID da conta. Se você não fornecer um ID da conta, o ID da conta associado às credenciais fornecidas por você para assinar a solicitação será pressuposto. Para obter mais informações, consulte Como usar o AWS SDK para .NET com o Amazon Glacier.

3. Execute o método `DescribeVault` fornecendo o objeto de solicitação como um parâmetro.

   As informações de metadados de cofre retornadas pelo Amazon Glacier estão disponíveis no objeto `DescribeVaultResult`.

O trecho de código do C# a seguir ilustra as etapas anteriores. O trecho recupera informações de metadados de um cofre existente na região oeste dos EUA (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**
>
> Para obter informações sobre a API REST subjacente, consulte Describe Vault (GET vault).

Recuperar metadados de todos os cofres em uma região

Você também pode usar o `ListVaults` método para recuperar metadados de todos os cofres em uma região específica. AWS

O trecho de código do C# recupera uma lista de cofres na região oeste dos EUA (Oregon). A solicitação limita o número de cofres retornados na resposta a 5. O trecho de código então faz uma série de `ListVaults` chamadas para recuperar toda a lista de cofres da região. 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);
```

No segmento de código anterior, se você não especificar o valor `Limit` na solicitação, o Amazon Glacier retornará até 10 cofres, conforme definido pela API do Amazon Glacier.

As informações retornadas para cada cofre na lista são as mesmas obtidas por você chamando o método `DescribeVault` para um cofre específico.

> **Note**
>
> O método `ListVaults` chama a API REST subjacente (consulte List Vaults (GET vaults)).

Exemplo: Recuperação de metadados do Vault usando a API de baixo nível do AWS SDK para .NET

Para obter um exemplo de código funcional, consulte Exemplo: operações do Vault usando a API de baixo nível do AWS SDK para .NET. O exemplo de código do C# cria um cofre e recupera os metadados do cofre.

Recuperar metadados do cofre usando a API REST

Para listar cofres usando a API REST, consulte List Vaults (GET vaults). Para descrever um cofre, consulte Describe Vault (GET vault).

Recuperar metadados do cofre no Amazon Glacier usando a AWS Command Line Interface

Este exemplo mostra como recuperar informações e metadados do cofre no Amazon Glacier usando a AWS Command Line Interface (AWS CLI).

Tópicos

- (Pré-requisito) Configurar a AWS CLI
- Recuperar metadados do cofre no usando o AWS CLI

(Pré-requisito) Configurar a AWS CLI

1. Faça download e configure a AWS CLI. Para obter instruções, consulte os seguintes tópicos no Guia do usuário do AWS Command Line Interface:

   Instalar a AWS Command Line Interface

   Configurando a AWS Command Line Interface
2. Verifique a configuração digitando os comandos a seguir no prompt de comando. Esses comandos não fornecem as credenciais explicitamente, de modo que as credenciais do perfil padrão são usadas.

   - Tente usar o comando de ajuda.

   ```
   aws help
   ```

- Para obter uma lista dos cofres do Amazon Glacier na conta configurada, use o comando `list-vaults`. Substitua *123456789012* pelo ID da sua Conta da AWS.

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

- Para obter os dados da configuração atual para a AWS CLI, use o comando `aws configure list`.

```
aws configure list
```

Recuperar metadados do cofre no usando o AWS CLI

- Use o comando `describe-vault` para descrever um cofre chamado *awsexamplevault* na conta *111122223333*.

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

Fazer download de um inventário de cofre no Amazon Glacier

Depois que você fizer upload do primeiro arquivo para o cofre, o Amazon Glacier vai criar e atualizar automaticamente um inventário de cofre uma vez por dia, aproximadamente. Depois que o Amazon Glacier criar o primeiro inventário, normalmente levará de meio dia a um dia até que esse inventário esteja disponível para recuperação. Você pode recuperar um inventário de cofre do Amazon Glacier com o seguinte processo de duas etapas:

1. Inicie um trabalho de recuperação de inventário usando a operação Initiate Job (trabalhos POST).

> **Important**
>
> Uma política de recuperação de dados pode causar uma falha na solicitação do trabalho de recuperação de inicialização com uma exceção `PolicyEnforcedException`. Para obter mais informações sobre políticas de recuperação de dados, consulte Políticas de recuperação de dados do Amazon Glacier. Para obter mais informações sobre a exceção `PolicyEnforcedException`, consulte Respostas de erro.

2. Após a conclusão do trabalho, faça download dos bytes usando a operação Get Job Output (GET output).

Por exemplo, recuperar um arquivo ou um inventário de cofre exige que você primeiro inicie um trabalho de recuperação. A solicitação de trabalho é executada de maneira assíncrona. Quando você inicia um trabalho de recuperação, o Amazon Glacier cria um trabalho e retorna um ID de trabalho na resposta. Quando o Amazon Glacier conclui o trabalho, você pode receber a saída do trabalho, os bytes do arquivo ou os dados do inventário de cofre.

O trabalho deverá ser concluído para você obter a saída. Para determinar o status do trabalho, você tem as seguintes opções:

- Aguardar a notificação de conclusão de um trabalho: você pode especificar um tópico do Amazon Simple Notification Service (Amazon SNS) no qual o Amazon Glacier poderá publicar uma notificação depois que o trabalho for concluído. Você pode especificar o tópico do Amazon SNS usando os seguintes métodos:
  - Especifique um tópico do Amazon SNS por trabalho.

    Ao iniciar um trabalho, você também pode especificar um tópico do Amazon SNS.
  - Defina a configuração de notificação no cofre.

    Você pode definir a configuração de notificação para eventos específicos no cofre (consulte Configurar notificações do cofre no Amazon Glacier). O Amazon Glacier envia uma mensagem para o tópico do SNS especificado sempre que o evento específico ocorrer.

  Se você tiver a configuração de notificação definida no cofre e também especificar um tópico do Amazon SNS ao iniciar um trabalho, o Amazon Glacier enviará uma mensagem de conclusão do trabalho para ambos os tópicos.

  Você pode configurar o tópico do SNS para notificá-lo por e-mail ou armazenar a mensagem em um Amazon Simple Queue Service (Amazon SQS) que o aplicativo possa sondar. Quando uma mensagem for exibida na fila, você poderá verificar se o trabalho foi concluído com êxito e, em seguida, fazer download da saída do trabalho.
- Solicitar informações do trabalho explicitamente: o Amazon Glacier também oferece uma descrição da operação do trabalho (Trabalho de descrição (GET JobID)) que permite sondar informações desse trabalho. Periodicamente, você pode enviar essa solicitação para obter informações de trabalho. No entanto, usar notificações do Amazon SNS é a opção recomendada.

> **Note**
>
> As informações obtidas por você por meio da notificação do SNS são as mesmas recebidas quando se chama Describe Job.

Tópicos

- Sobre o inventário
- Baixando um inventário do Vault no Amazon Glacier usando o AWS SDK para Java
- Baixando um inventário do Vault no Amazon Glacier usando o AWS SDK para .NET
- Fazer download de um inventário de cofre usando a API REST
- Baixando um inventário do Vault no Amazon Glacier usando o AWS Command Line Interface

Sobre o inventário

O Amazon Glacier atualiza um inventário de cofre pelo menos uma vez por dia, começando no dia em que você faz o primeiro upload no cofre. Se não houver adições ou exclusões de arquivo no cofre desde o último inventário, a data do inventário não será atualizada. Quando você inicia um trabalho para um inventário do cofre, o Amazon Glacier retorna o último inventário gerado, que é point-in-time um instantâneo e não dados em tempo real. Depois que o Amazon Glacier criar o primeiro inventário para o cofre, normalmente levará de meio dia a um dia até que esse inventário esteja disponível para recuperação.

Talvez você não ache útil recuperar um inventário de cofre para cada upload de arquivo. No entanto, suponhamos que você mantenha um banco de dados no lado do cliente associando metadados sobre os arquivos cujo upload fez para o Amazon Glacier. Nesse caso, talvez você ache o inventário de cofre útil para reconciliar informações, conforme necessário, no seu banco de dados com o inventário de cofre real. É possível limitar o número de itens do inventário recuperados filtrando a data de criação do arquivamento ou definindo uma cota. Para obter mais informações sobre como limitar a recuperação do inventário, consulte Recuperação do inventário de intervalo.

O inventário pode ser retornado em dois formatos: valores separados por vírgula (CSV) ou JSON. Você também pode especificar o formato ao iniciar o trabalho de inventário. O formato padrão é JSON. Para obter mais informações sobre os campos de dados retornados em uma saída do trabalho de inventário, consulte Corpo da resposta da API Obter saída de trabalho.

Baixando um inventário do Vault no Amazon Glacier usando o AWS SDK para Java

Estas são as etapas para recuperar um inventário de cofre usando a API de nível inferior do AWS SDK para Java. A API de nível superior não dá suporte à recuperação de um inventário de cofre.

1. Crie uma instância da classe `AmazonGlacierClient` (o cliente).

   Você precisa especificar uma AWS região em que o cofre reside. Todas as operações que você executa usando esse cliente se aplicam a essa AWS região.

2. Inicie um trabalho de recuperação de inventário executando o método `initiateJob`.

   Execute `initiateJob` fornecendo informações de trabalho em um objeto `InitiateJobRequest`.

   > **Note**
   >
   > Se um inventário não tiver sido concluído para o cofre, um erro será retornado. O Amazon Glacier prepara um inventário para cada cofre periodicamente, a cada 24 horas.

   Em resposta, o Amazon Glacier retorna um ID de trabalho. A resposta está disponível em uma instância da classe `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. Aguarde a conclusão do trabalho .

Você deve aguardar até a saída do trabalho estar pronta para download. Se você tiver definido uma configuração de notificação no cofre ou especificado um tópico do Amazon Simple Notification Service (Amazon SNS) ao iniciar o trabalho, o Amazon Glacier enviará uma mensagem para o tópico depois de concluir o trabalho.

Você também pode sondar o Amazon Glacier chamando o método `describeJob` para determinar o status de conclusão do trabalho. No entanto, usar um tópico do Amazon SNS para notificação é a abordagem recomendada. O exemplo de código indicado na seção a seguir usa o Amazon SNS para que o Amazon Glacier publique uma mensagem.

4. Faça download da saída do trabalho (dados de inventário de cofre) executando o método `getJobOutput`.

   Você fornece o ID da conta, o ID do trabalho e o nome do cofre criando uma instância da classe `GetJobOutputRequest`. Se você não fornecer um ID da conta, o ID da conta associado às credenciais fornecidas por você para assinar a solicitação será usado. Para obter mais informações, consulte Usando o AWS SDK para Java com o Amazon Glacier.

   A saída retornada pelo Amazon Glacier está disponível no objeto `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**
>
> Para obter informações sobre a API REST subjacente relacionada ao trabalho, consulte Operações de trabalho.

Exemplo: recuperar um inventário de cofre usando o Amazon SDK para Java

O exemplo de código do Java a seguir recupera o inventário do cofre especificado.

O exemplo realiza as seguintes tarefas:

- Cria um tópico do Amazon Simple Notification Service (Amazon SNS).

  O Amazon Glacier enviará uma notificação para esse tópico depois de concluir o trabalho.
- Cria uma fila do Amazon Simple Queue Service (Amazon SQS)

  O exemplo anexa uma política à fila para permitir que o tópico do Amazon SNS publique mensagens na fila.
- Inicia um trabalho para fazer download do arquivo especificado.

  Na solicitação de trabalho, o tópico do Amazon SNS criado é especificado para que o Amazon Glacier possa publicar uma notificação no tópico depois de concluir o trabalho.
- Verifique o Amazon SQS em busca de uma mensagem que contenha o ID do trabalho.

  Se houver uma mensagem, analise o JSON e verifique se o trabalho foi concluído com êxito. Se for o caso, faça download do arquivo.
- Realiza uma limpeza excluindo o tópico do Amazon SNS e a fila do Amazon SQS criada por ele.

```
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));
    }
}
```

Baixando um inventário do Vault no Amazon Glacier usando o AWS SDK para .NET

Estas são as etapas para recuperar um inventário de cofre usando a API de nível inferior do AWS SDK para .NET. A API de nível superior não dá suporte à recuperação de um inventário de cofre.

1. Crie uma instância da classe `AmazonGlacierClient` (o cliente).

   Você precisa especificar uma AWS região em que o cofre reside. Todas as operações que você executa usando esse cliente se aplicam a essa AWS região.
2. Inicie um trabalho de recuperação de inventário executando o método `InitiateJob`.

   Pode-se fornecer informações de trabalho em um objeto `InitiateJobRequest`. O Amazon Glacier retorna um ID do trabalho em resposta. A resposta está disponível em uma instância da classe `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. Aguarde a conclusão do trabalho .

   Você deve aguardar até a saída do trabalho estar pronta para download. Se você tiver definido uma configuração de notificação no cofre identificando um tópico do Amazon Simple Notification Service (Amazon SNS) ou especificado um tópico do Amazon SNS quando tiver iniciado um trabalho, o Amazon Glacier enviará uma mensagem para esse tópico depois de concluir o trabalho. O exemplo de código indicado na seção a seguir usa o Amazon SNS para que o Amazon Glacier publique uma mensagem.

   Você também pode sondar o Amazon Glacier chamando o método `DescribeJob` para determinar o status de conclusão do trabalho. Apesar disso, usar o tópico do Amazon SNS para notificação é a abordagem recomendada.

4. Faça download da saída do trabalho (dados de inventário de cofre) executando o método `GetJobOutput`.

   Você fornece o ID da conta, o nome do cofre e as informações do ID do trabalho criando uma instância da classe `GetJobOutputRequest`. Se você não fornecer um ID da conta, o ID da conta associado às credenciais fornecidas por você para assinar a solicitação será pressuposto. Para obter mais informações, consulte Como usar o AWS SDK para .NET com o Amazon Glacier.

   A saída retornada pelo Amazon Glacier está disponível no objeto `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**
>
> Para obter informações sobre a API REST subjacente relacionada ao trabalho, consulte Operações de trabalho.

Exemplo: Recuperação de um inventário do Vault usando a API de baixo nível do AWS SDK para .NET

O exemplo de código do C# a seguir recupera o inventário do cofre especificado.

O exemplo realiza as seguintes tarefas:

- Configure um tópico do SNS.

  O Amazon Glacier enviará uma notificação para esse tópico depois de concluir o trabalho.
- Configurar uma fila do Amazon SQS.

  O exemplo anexa uma política à fila para permitir que o tópico do Amazon SNS publique mensagens.
- Inicie um trabalho para fazer download do arquivo especificado.

  Na solicitação de trabalho, o exemplo especifica o tópico do Amazon SNS, de maneira que o Amazon Glacier possa enviar uma mensagem depois de concluir o trabalho.
- Periodicamente, verifique a fila do Amazon SQS em busca de uma mensagem.

Se houver uma mensagem, analise o JSON e verifique se o trabalho foi concluído com êxito. Se for o caso, faça download do arquivo. O exemplo de código usa a biblioteca do JSON.NET (consulte JSON.NET) para analisar o JSON.

- Limpe excluindo o tópico do Amazon SNS e a fila do Amazon SQS criada por ele.

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);
      }
    }
  }
}
```

Fazer download de um inventário de cofre usando a API REST

Para fazer download de um inventário de cofre usando a API REST

Fazer download de um inventário de cofre é um processo de duas etapas.

1. Inicie um trabalho do tipo `inventory-retrieval`. Para obter mais informações, consulte Initiate Job (trabalhos POST).
2. Depois que o trabalho for concluído, faça download dos dados de inventário. Para obter mais informações, consulte Get Job Output (GET output).

Baixando um inventário do Vault no Amazon Glacier usando o AWS Command Line Interface

Siga estas etapas para fazer download de um inventário de cofre no Amazon Glacier usando a AWS Command Line Interface (AWS CLI).

Tópicos

- (Pré-requisito) Configurando o AWS CLI
- Exemplo: Baixar um inventário do Vault usando o AWS CLI

(Pré-requisito) Configurando o AWS CLI

1. Faça download e configure a AWS CLI. Para obter instruções, consulte os seguintes tópicos no Guia do usuário do AWS Command Line Interface :

   Instalando o AWS Command Line Interface

Configurando o AWS Command Line Interface

2. Verifique sua AWS CLI configuração inserindo os seguintes comandos no prompt de comando. Esses comandos não fornecem as credenciais explicitamente, de modo que as credenciais do perfil padrão são usadas.

   - Tente usar o comando de ajuda.

     ```
     aws help
     ```

   - Para obter uma lista dos cofres do Amazon Glacier na conta configurada, use o comando `list-vaults`. *123456789012*Substitua pelo seu Conta da AWS ID.

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

   - Para ver os dados de configuração atuais do AWS CLI, use o `aws configure list` comando.

     ```
     aws configure list
     ```

Exemplo: Baixar um inventário do Vault usando o AWS CLI

1. Use o comando `initiate-job` para iniciar um trabalho de recuperação de inventário.

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

   Saída esperada:

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

2. Use o comando `describe-job` para verificar o status do trabalho de recuperação anterior.

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

Saída esperada:

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

3. Aguarde a conclusão do trabalho .

   Você deve aguardar até a saída do trabalho estar pronta para download. O ID do trabalho não expira por pelo menos 24 horas depois que o Amazon Glacier conclui o trabalho. Se você tiver definido uma configuração de notificação no cofre ou especificado um tópico do Amazon Simple Notification Service (Amazon SNS) ao iniciar o trabalho, o Amazon Glacier enviará uma mensagem para o tópico depois de concluir o trabalho.

   É possível definir a configuração de notificação para eventos específicos no cofre. Para obter mais informações, consulte Configurar notificações do cofre no Amazon Glacier. O Amazon Glacier enviará uma mensagem para o tópico do SNS especificado sempre que o evento específico ocorrer.

4. Quando estiver concluído, use o comando `get-job-output` para fazer download do trabalho de recuperação para o arquivo `output.json`.

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

Esse comando produz um arquivo com os campos a seguir.

```
{
"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":
...
]}
```

Configurar notificações do cofre no Amazon Glacier

Recuperar qualquer dado do Amazon Glacier, como um arquivo de um cofre ou um inventário de cofre, é um processo de duas etapas.

1. Inicie um trabalho de recuperação.
2. Após a conclusão do trabalho, faça o download da saída do mesmo.

Você pode definir uma configuração de notificação em um cofre, de maneira que, quando um trabalho for concluído, uma mensagem será enviada para um tópico do Amazon Simple Notification Service (Amazon SNS).

Tópicos

- Configurar notificações de cofre no Amazon Glacier: conceitos gerais
- Configurando notificações do Vault no Amazon Glacier usando o AWS SDK para Java
- Configurando notificações do Vault no Amazon Glacier usando o AWS SDK para .NET
- Configurar notificações de cofre no Amazon Glacier usando a API REST
- Configurar notificações de cofre usando o console do Amazon Glacier
- Configurando as notificações do Vault usando o AWS Command Line Interface

Configurar notificações de cofre no Amazon Glacier: conceitos gerais

Uma solicitação de trabalho de recuperação do Amazon Glacier é executada de maneira assíncrona. Você deve aguardar até que o Amazon Glacier conclua o trabalho para obter a saída. Você pode sondar periodicamente o Amazon Glacier para determinar o status do trabalho, mas essa não é uma abordagem ideal. O Amazon Glacier também oferece suporte a notificações. Quando um trabalho é concluído, ele pode publicar uma mensagem em um tópico do Amazon Simple Notification Service (Amazon SNS). Esse atributo exige que você defina a configuração de notificação no cofre. Na configuração, é possível identificar um ou mais eventos e um tópico do Amazon SNS para o qual você quer que o Amazon Glacier envie uma mensagem quando o evento ocorrer.

O Amazon Glacier define eventos especificamente relacionados à conclusão do trabalho (`ArchiveRetrievalCompleted`, `InventoryRetrievalCompleted`) que você pode adicionar à configuração de notificação do cofre. Quando um trabalho específico é concluído, o Amazon Glacier publica uma mensagem de notificação no tópico do SNS.

A configuração de notificação é um documento JSON conforme mostrado no exemplo a seguir.

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

Você pode configurar somente um tópico do Amazon SNS para um cofre.

> **Note**
>
> Adicionar uma configuração de notificação a um cofre faz o Amazon Glacier enviar uma notificação sempre que o evento especificado na configuração de notificação ocorrer. Você também pode especificar um tópico do Amazon SNS em cada solicitação de iniciação do trabalho. Se você adicionar a configuração de notificação no cofre e também especificar um tópico do Amazon SNS na solicitação do trabalho de iniciação, o Amazon Glacier enviará ambas as notificações.

A mensagem de conclusão do trabalho que o Amazon Glacier envia inclui informações como o tipo de trabalho (`InventoryRetrieval`, `ArchiveRetrieval`), o status de conclusão do trabalho, o

nome do tópico do SNS, o código de status do trabalho e o ARN do cofre. Esta é uma notificação de exemplo que o Amazon Glacier enviou para um tópico do SNS após a conclusão de um trabalho `InventoryRetrieval`.

```
{
 "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"
}
```

Se o campo `Completed` for verdadeiro, você também deverá verificar o `StatusCode` para examinar se o trabalho foi concluído com êxito ou falhou.

> **Note**
>
> O tópico do Amazon SNS deve permitir que o cofre publique uma notificação. Por padrão, somente o proprietário do tópico do Amazon SNS pode publicar uma mensagem no tópico. No entanto, se o tópico do Amazon SNS e o cofre forem de propriedade de diferentes pessoas Contas da AWS, você deverá configurar o tópico do Amazon SNS para aceitar publicações do cofre. Você pode configurar a política de tópico do Amazon SNS no console do Amazon SNS.

Para obter mais informações sobre o Amazon SNS, consulte Conceitos básicos do Amazon SNS.

Configurando notificações do Vault no Amazon Glacier usando o AWS SDK para Java

Estas são as etapas para configurar notificações em um cofre usando a API de nível inferior do AWS SDK para Java.

1. Crie uma instância da classe `AmazonGlacierClient` (o cliente).

   Você precisa especificar uma AWS região em que o cofre reside. Todas as operações que você executa usando esse cliente se aplicam a essa AWS região.

2. Forneça informações de configuração da notificação criando uma instância da classe `SetVaultNotificationsRequest`.

   Você precisa fornecer o nome do cofre, as informações de configuração da notificação e o ID da conta. Especificando uma configuração de notificação, você fornece o nome do recurso da Amazon (ARN) de um tópico do Amazon SNS existente e um ou mais eventos para os quais deseja ser notificado. Para obter uma lista de eventos compatíveis, consulte Definir configuração de notificação de cofre (PUT notification-configuration).

3. Execute o método `setVaultNotifications` fornecendo o objeto de solicitação como um parâmetro.

O trecho de código Java a seguir ilustra as etapas anteriores. O trecho define uma configuração de notificação em um cofre. A configuração solicita que o Amazon Glacier envie uma notificação para o tópico Amazon SNS especificado quando o evento `ArchiveRetrievalCompleted` ou `InventoryRetrievalCompleted` ocorre.

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

> **Note**
>
> Para obter informações sobre a API REST subjacente, consulte Operações de cofre.

Exemplo: Definindo a configuração de notificação em um cofre usando o AWS SDK para Java

O exemplo de código do Java a seguir define uma configuração de notificação de cofre, exclui a configuração e restaura a configuração. Para step-by-step obter instruções sobre como executar o exemplo a seguir, consulteUsando o AWS SDK para Java com o 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);
    }
}
```

Configurando notificações do Vault no Amazon Glacier usando o AWS SDK para .NET

Estas são as etapas para configurar notificações em um cofre usando a API de nível inferior do AWS SDK para .NET.

1. Crie uma instância da classe `AmazonGlacierClient` (o cliente).

   Você precisa especificar uma AWS região em que o cofre reside. Todas as operações que você executa usando esse cliente se aplicam a essa AWS região.
2. Forneça informações de configuração da notificação criando uma instância da classe `SetVaultNotificationsRequest`.

   Você precisa fornecer o nome do cofre, as informações de configuração da notificação e o ID da conta. Se você não fornecer um ID da conta, o ID da conta associado às credenciais fornecidas por você para assinar a solicitação será pressuposto. Para obter mais informações, consulte Como usar o AWS SDK para .NET com o Amazon Glacier.

   Especificando uma configuração de notificação, você fornece o nome do recurso da Amazon (ARN) de um tópico do Amazon SNS existente e um ou mais eventos para os quais deseja ser notificado. Para obter uma lista de eventos compatíveis, consulte Definir configuração de notificação de cofre (PUT notification-configuration).
3. Execute o método `SetVaultNotifications` fornecendo o objeto de solicitação como um parâmetro.
4. Depois de definir a configuração de notificação em um cofre, você poderá recuperar informações de configuração chamando o método `GetVaultNotifications` e removê-la chamando o método `DeleteVaultNotifications` fornecido pelo cliente.

Exemplo: Definindo a configuração de notificação em um cofre usando o AWS SDK para .NET

O exemplo de código do C# a seguir ilustra as etapas anteriores. O exemplo define a configuração de notificação no cofre (“`examplevault`”) na região oeste dos EUA (Oregon), recupera a configuração e a exclui. A configuração solicita que o Amazon Glacier envie uma notificação para o tópico Amazon SNS especificado quando o evento `ArchiveRetrievalCompleted` ou `InventoryRetrievalCompleted` ocorre.

> **Note**
>
> Para obter informações sobre a API REST subjacente, consulte Operações de cofre.

Para step-by-step obter instruções sobre como executar o exemplo a seguir, consulteExecutar exemplos de código. Você precisa atualizar o código conforme mostrado e fornecer um nome de cofre existente, além de um tópico do 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);
    }
```

```
  }
}
```

Configurar notificações de cofre no Amazon Glacier usando a API REST

Para configurar notificações de cofre usando a API REST, consulte Definir configuração de notificação de cofre (PUT notification-configuration). Além disso, você também pode obter (Get Vault Notifications (GET notification-configuration)) e excluir notificações de cofre (Delete Vault Notifications (DELETE notification-configuration)).

Configurar notificações de cofre usando o console do Amazon Glacier

Esta seção descreve como configurar notificações de cofre usando o console do Amazon Glacier. Ao configurar notificações, você especifica eventos de conclusão do trabalho que enviam uma notificação para um tópico do Amazon Simple Notification Service (Amazon SNS). Além de configurar notificações de cofre, você também pode especificar um tópico para publicação de uma notificação ao iniciar um trabalho. Se o cofre estiver configurado para notificação de um evento específico e você também configurar notificações na solicitação de iniciação do trabalho, duas notificações serão enviadas.

Para configurar uma notificação de cofre

1. Faça login Console de gerenciamento da AWS e abra o console do Amazon Glacier em casahttps://console.aws.amazon.com/glacier/.
2. No painel de navegação à esquerda, selecione Cofres.
3. Na lista Cofres, escolha um cofre.
4. Na seção Notifications, escolha Edit.
5. Na página Notificações de eventos, selecione Ativar notificações.
6. Na seção Notifications, escolha uma das seguintes opções do Amazon Simple Notification Service (Amazon SNS) e siga as etapas correspondentes:

Opções do Amazon SNS	Ação
Criar novo tópico do SNS	1. Selecione Criar novo tópico do SNS. 2. Em Topic name, insira um nome de um novo tópico.

Opções do Amazon SNS	Ação
	Os nomes dos tópicos podem ter até 256 caracteres. Caracteres alfanuméricos, hifens (-), sublinhados (_) são permitidos. Os nomes dos tópicos devem ser exclusivos na conta Região da AWS e. 3. (Opcional) Se você quiser se inscrever no tópico usando mensagens SMS, insira um nome para Display name. O nome de exibição pode ter até 100 caracteres.
Escolha um tópico do SNS existente	1. Escolha Escolha um tópico do SNS existente. 2. Em Specify SNS topic, escolha uma das seguintes opções: • Escolha entre seus tópicos do SNS Uma lista SNS topic é exibida. Escolha um tópico existente, na lista suspensa. • Inserir ARN do tópico do SNS Uma caixa de texto Amazon SNS topic ARN é exibida. Entre o nome do recurso da Amazon (ARN) do tópico SNS. O ARN do tópico SNS tem o seguinte formato: `arn:aws:sns: region:account-id :topic-name` Você pode descobrir o ARN do tópico SNS usando o console do Amazon SNS:

7. Em Eventos, selecione um ou ambos os eventos para os quais você deseja enviar notificações:

   - Para enviar uma notificação somente quando os trabalhos de recuperação de arquivo estiverem concluídos, selecione Trabalho de recuperação de arquivo concluído.
   - Para enviar uma notificação somente quando os trabalhos de inventário de cofre estiverem concluídos, selecione Trabalho de recuperação do inventário de cofre concluído.

Configurando as notificações do Vault usando o AWS Command Line Interface

Esta seção descreve como configurar notificações de cofre usando o AWS Command Line Interface. Ao configurar notificações, você especifica os eventos de conclusão do trabalho que acionam a notificação para um tópico do Amazon Simple Notification Service (Amazon SNS). Além de configurar notificações do cofre, você também pode especificar um tópico para publicação de uma notificação ao iniciar um trabalho. Se o cofre estiver configurado para notificação de um evento específico e você especificar a notificação na solicitação de iniciação do trabalho, duas notificações serão enviadas.

Siga estas etapas para configurar a notificação de cofre usando o AWS CLI.

Tópicos

- (Pré-requisito) Configurando o AWS CLI
- Exemplo: Configurar notificações do Vault usando o AWS CLI

(Pré-requisito) Configurando o AWS CLI

1. Faça download e configure a AWS CLI. Para obter instruções, consulte os seguintes tópicos no Guia do usuário do AWS Command Line Interface :

   Instalando o AWS Command Line Interface

   Configurando o AWS Command Line Interface

2. Verifique sua AWS CLI configuração inserindo os seguintes comandos no prompt de comando. Esses comandos não fornecem as credenciais explicitamente, de modo que as credenciais do perfil padrão são usadas.

   - Tente usar o comando de ajuda.

```
aws help
```

- Para obter uma lista dos cofres do Amazon Glacier na conta configurada, use o comando `list-vaults`. *123456789012*Substitua pelo seu Conta da AWS ID.

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

- Para ver os dados de configuração atuais do AWS CLI, use o `aws configure list` comando.

```
aws configure list
```

Exemplo: Configurar notificações do Vault usando o AWS CLI

1. Use o comando `set-vault-notifications` para configurar as notificações que serão enviadas quando eventos específicos acontecerem em um cofre. Por padrão, você não recebe nenhuma notificação.

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

2. A configuração de notificação é um documento JSON conforme mostrado no exemplo a seguir.

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

   Para obter mais informações sobre o uso dos tópicos do Amazon SNS para o Amazon Glacier, consulte Configurar notificações de cofre no Amazon Glacier: conceitos gerais

   Para obter mais informações sobre o Amazon SNS, consulte Conceitos básicos do Amazon SNS.

Excluir um cofre no Amazon Glacier

O Amazon Glacier excluirá um cofre somente se não houver arquivos no cofre até o último inventário computado e se não houver gravações no cofre desde o inventário mais recente. Para obter informações sobre como excluir arquivos, consulte . Excluir um arquivo no Amazon Glacier. Para obter mais informações sobre como fazer download de um inventário de cofre, consulte ., Fazer download de um inventário de cofre no Amazon Glacier.

> **Note**
>
> O Amazon Glacier prepara um inventário para cada cofre periodicamente, a cada 24 horas. Como o inventário pode não refletir as informações mais recentes, o Amazon Glacier garante que o cofre esteja realmente vazio verificando se houve alguma operação de gravação desde o último inventário do cofre.

> **Note**
>
> Para exclusão automática de arquivos do cofre, consulte Exclusão automática dos arquivos do cofre no Amazon S3 Glacier.

Tópicos

- Excluir um cofre no Amazon Glacier usando AWS SDK para Java
- Excluir um cofre no Amazon Glacier usando AWS SDK para .NET
- Excluir um cofre no Amazon Glacier usando a API REST
- Excluir um cofre vazio usando o console do Amazon Glacier
- Excluindo um cofre no Amazon Glacier usando o AWS Command Line Interface

Excluir um cofre no Amazon Glacier usando AWS SDK para Java

Estas são as etapas para excluir um cofre usando-se a API de nível inferior do AWS SDK para Java.

1. Crie uma instância da classe `AmazonGlacierClient` (o cliente).

Você precisa especificar uma região da AWS onde deseja excluir um cofre. Todas as operações realizadas por você usando esse cliente se aplicam a essa regiãoAWS.

2. Forneça informações sobre a solicitação criando uma instância da classe `DeleteVaultRequest`.

   Você precisa fornecer o nome do cofre e o ID da conta. Se você não fornecer um ID da conta, o ID da conta associado às credenciais fornecidas por você para assinar a solicitação será pressuposto. Para obter mais informações, consulte Usando o AWS SDK para Java com o Amazon Glacier.

3. Execute o método `deleteVault` fornecendo o objeto de solicitação como um parâmetro.

   O Amazon Glacier exclui o cofre somente se ele estiver vazio. Para obter mais informações, consulte Delete Vault (DELETE vault).

O trecho de código Java a seguir ilustra as etapas anteriores.

```
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**
>
> Para obter informações sobre a API REST subjacente, consulte Delete Vault (DELETE vault).

Exemplo: excluir um cofre usando o AWS SDK para Java

Para obter um exemplo de código funcional, consulte Exemplos: como criar um cofre usando a AWS SDK para Java. O exemplo de código do Java mostra as operações de cofre básicas, inclusive criar e excluir cofre.

Excluir um cofre no Amazon Glacier usando AWS SDK para .NET

As APIs de nível superior inferior disponibilizadas pelo Amazon SDK for .NET fornecem um método para excluir um cofre.

Tópicos

- Excluir um cofre usando a API de nível superior do AWS SDK para .NET
- Excluir um cofre usando a API de nível inferior do AWS SDK para .NET

Excluir um cofre usando a API de nível superior do AWS SDK para .NET

A classe `ArchiveTransferManager` da API de nível superior fornece o método `DeleteVault` que você pode usar para excluir um cofre.

Exemplo: excluir um cofre usando a API de nível superior do AWS SDK para .NET

Para obter um exemplo de código funcional, consulte Exemplo: operações do Vault usando a API de alto nível do AWS SDK para .NET. O exemplo de código do C# mostra as operações de cofre básicas, inclusive criar e excluir cofre.

Excluir um cofre usando a API de nível inferior do AWS SDK para .NET

Veja a seguir as etapas para excluir um cofre usando o console do AWS SDK para .NET.

1. Crie uma instância da classe `AmazonGlacierClient` (o cliente).

   Você precisa especificar uma região da AWS onde deseja excluir um cofre. Todas as operações realizadas por você usando esse cliente se aplicam a essa regiãoAWS.
2. Forneça informações sobre a solicitação criando uma instância da classe `DeleteVaultRequest`.

   Você precisa fornecer o nome do cofre e o ID da conta. Se você não fornecer um ID da conta, o ID da conta associado às credenciais fornecidas por você para assinar a solicitação será pressuposto. Para obter mais informações, consulte Como usar o AWS SDK para .NET com o Amazon Glacier.
3. Execute o método `DeleteVault` fornecendo o objeto de solicitação como um parâmetro.

   O Amazon Glacier exclui o cofre somente se ele estiver vazio. Para obter mais informações, consulte Delete Vault (DELETE vault).

O trecho de código do C# a seguir ilustra as etapas anteriores. O trecho recupera informações de metadados de um cofre existente na região AWS padrão.

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

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

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

> **Note**
>
> Para obter informações sobre a API REST subjacente, consulte Delete Vault (DELETE vault).

Exemplo: excluir um cofre usando a API de nível inferior do AWS SDK para .NET

Para obter um exemplo de código funcional, consulte Exemplo: operações do Vault usando a API de baixo nível do AWS SDK para .NET. O exemplo de código do C# mostra as operações de cofre básicas, inclusive criar e excluir cofre.

Excluir um cofre no Amazon Glacier usando a API REST

Para excluir um cofre usando a API REST, consulte Delete Vault (DELETE vault).

Excluir um cofre vazio usando o console do Amazon Glacier

> **Note**
>
> Se for excluir um cofre que não esteja vazio, será necessário primeiro excluir todos os arquivos existentes antes de excluir o cofre. Você pode fazer isso escrevendo um código para fazer uma solicitação de exclusão de arquivamento usando a API REST AWS SDK para Java, a AWS SDK para .NET, a ou usando a AWS Command Line Interface (AWS CLI). Para obter informações sobre como excluir arquivos, consulte Etapa 5: excluir um arquivo de um cofre no Amazon Glacier.

Depois que o cofre estiver vazio, você poderá excluí-lo utilizando as seguintes etapas.

Para excluir um cofre vazio usando o console do Amazon Glacier

1. Faça login Console de gerenciamento da AWS e abra o console do Amazon Glacier no Amazon Glacier Console.
2. Em Selecionar uma região, escolha Região da AWS onde o cofre existe.
3. No painel de navegação à esquerda, selecione Cofres.
4. Na lista Cofres, selecione a opção ao lado do nome do cofre que você deseja excluir e escolha Excluir na parte superior da página.
5. Na caixa de diálogo Excluir cofre, confirme que deseja excluir o cofre escolhendo Excluir.

   > **Important**
   >
   > A exclusão de um cofre não pode ser desfeita.

6. Para verificar se você excluiu o cofre, abra a lista Cofres e insira o nome do cofre que você excluiu. Se não for possível encontrar o cofre, sua exclusão foi bem-sucedida.

Excluindo um cofre no Amazon Glacier usando o AWS Command Line Interface

É possível excluir cofres vazios e não vazios no Amazon Glacier usando AWS Command Line Interface (AWS CLI).

Tópicos

- (Pré-requisito) Configurando o AWS CLI
- Exemplo: Excluindo um cofre vazio usando o AWS CLI
- Exemplo: Excluindo um cofre não vazio usando o AWS CLI

(Pré-requisito) Configurando o AWS CLI

1. Faça download e configure a AWS CLI. Para obter instruções, consulte os seguintes tópicos no Guia do usuário do AWS Command Line Interface :

   Instalando o AWS Command Line Interface

Configurando o AWS Command Line Interface

2. Verifique sua AWS CLI configuração inserindo os seguintes comandos no prompt de comando. Esses comandos não fornecem as credenciais explicitamente, de modo que as credenciais do perfil padrão são usadas.

   - Tente usar o comando de ajuda.

     ```
     aws help
     ```

   - Para obter uma lista dos cofres do Amazon Glacier na conta configurada, use o comando `list-vaults`. *123456789012*Substitua pelo seu Conta da AWS ID.

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

   - Para ver os dados de configuração atuais do AWS CLI, use o `aws configure list` comando.

     ```
     aws configure list
     ```

Exemplo: Excluindo um cofre vazio usando o AWS CLI

- Use o comando `delete-vault` para excluir um cofre que não contém arquivamentos.

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

Exemplo: Excluindo um cofre não vazio usando o AWS CLI

O Amazon Glacier excluirá um cofre somente se não houve arquivamentos nem gravações nesse cofre desde o último inventário computado. A exclusão de um cofre não vazio é um processo de três etapas: recuperar o arquivo IDs do relatório de inventário de um cofre, excluir cada arquivo e, em seguida, excluir o cofre.

1. Use o comando `initiate-job` para iniciar um trabalho de recuperação de inventário.

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

Saída esperada:

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

2. Use o comando `describe-job` para verificar o status do trabalho de recuperação anterior.

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

Saída esperada:

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

3. Aguarde a conclusão do trabalho .

Você deve aguardar até a saída do trabalho estar pronta para download. Se você definiu uma configuração de notificação no cofre ou especificou um tópico do Amazon Simple Notification Service (Amazon SNS) ao iniciar o trabalho, o Amazon Glacier envia uma mensagem para o tópico depois de concluir o trabalho.

É possível definir a configuração de notificação para eventos específicos no cofre. Para obter mais informações, consulte Configurar notificações do cofre no Amazon Glacier. O Amazon

Glacier enviará uma mensagem para o tópico do SNS especificado sempre que o evento determinado ocorrer.

4. Quando estiver concluído, use o comando `get-job-output` para fazer download do trabalho de recuperação para o arquivo `output.json`.

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

Esse comando produz um arquivo com os campos a seguir.

```
{
"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. Use o comando `delete-archive` para excluir cada arquivamento de um cofre até que não haja mais nenhum.

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

> **Note**
>
> Se o seu ID de arquivo começar com um hífen ou outro caractere especial, você precisará colocá-lo entre aspas para executar esse comando.

6. Use o comando `initiate-job` para iniciar um novo trabalho de recuperação de inventário.

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

7. Quando estiver concluído, use o comando `delete-vault` para excluir um cofre sem arquivamentos.

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

Marcar seus Cofres do Amazon Glacier

Você pode atribuir os próprios metadados a cofres do Amazon Glacier na forma de tags. Uma tag é um par de chave/valor definido por você para um cofre. Para obter informações básicas sobre como marcar, inclusive restrições sobre tags, consulte Marcação de recursos do Amazon Glacier.

Os tópicos a seguir descrevem como você pode adicionar, listar e remover tags de cofres.

Tópicos

- Marcação de cofres usando o console do Amazon Glacier
- Marcar cofres usando a API do AWS CLI
- Marcação de cofres usando a API do Amazon Glacier
- Seções relacionadas

Marcação de cofres usando o console do Amazon Glacier

Você pode adicionar, listar e remover tags usando o console do Amazon Glacier conforme descrito nos procedimentos a seguir.

Para visualizar as tags de um cofre

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon Glacier em https://console.aws.amazon.com/glacier/home.
2. Em Selecionar uma região, selecione Região da AWS no seletor de região.
3. No painel de navegação à esquerda, selecione Cofres.
4. Na lista Cofres, escolha um cofre.

5. Escolha a guia Propriedades do cofre. Role até a seção Tags para ver as tags associadas ao cofre.

Para adicionar uma tag a um cofre

É possível associar até 50 tags a um cofre. As tags associadas a um objeto devem ter chaves de tag exclusivas.

Para obter mais informações sobre restrições de tags, consulte Marcação de recursos do Amazon Glacier.

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon Glacier em https://console.aws.amazon.com/glacier/home.
2. Em Selecionar uma região, selecione Região da AWS no seletor de região.
3. No painel de navegação à esquerda, selecione Cofres.
4. Na lista Cofre, escolha o nome do cofre ao qual você deseja adicionar tags.
5. Escolha a guia Propriedades do cofre.
6. Na seção Tags, escolha Add (Adicionar). A página Adicionar tags é exibida.
7. Na página Adicionar tags, especifique a chave da tag no campo Chave e, opcionalmente, especifique um valor de tag no campo Valor.
8. Escolha Salvar alterações.

Para editar uma tag

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon Glacier em https://console.aws.amazon.com/glacier/home.
2. Em Selecionar uma região, selecione Região da AWS no seletor de região.
3. No painel de navegação à esquerda, selecione Cofres.
4. Na lista de Cofre, escolha um nome de cofre.
5. Escolha a guia Propriedades do cofre e, em seguida, role para baixo até a seção Etiquetas.
6. Em Tags, marque a caixa de seleção ao lado das tags que você deseja alterar e depois escolha Editar. A página Editar tags é exibida.
7. Atualize a chave da tag no campo Chave e, opcionalmente, atualize o valor da tag no campo Valor.

8. Escolha Salvar alterações.

Para remover uma tag de um cofre

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon Glacier em https://console.aws.amazon.com/glacier/home.
2. Em Selecionar uma região, selecione Região da AWS no seletor de região.
3. No painel de navegação à esquerda, selecione Cofres.
4. Na lista Cofres, escolha o nome do cofre do qual você deseja remover as tags.
5. Escolha a guia Propriedades do cofre. Role até a seção Tags.
6. Em Tags, marque a caixa de seleção ao lado das tags que você deseja remover e depois escolha Excluir.
7. A caixa de diálogo Excluir tags é exibida. Para confirmar que deseja excluir a pilha selecionada, escolha Excluir.

Marcar cofres usando a API do AWS CLI

Siga estas etapas para adicionar, listar ou remover tags usando a AWS Command Line Interface (AWS CLI).

Cada tag é composta de uma chave e de um valor. Cada cofre pode ter até 50 tags.

1. Para adicionar tags a um cofre, use o comado `add-tags-to-vault`.

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

   Para obter mais informações sobre essa operação de cofre, consulte Adicionar tags ao cofre.
2. Para listar todas as tags anexadas a um cofre, use o comando `list-tags-for-vault`.

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

   Para obter mais informações sobre essa operação de cofre, consulte Listar tags para o cofre.
3. Remove uma ou mais tags do conjunto de tags anexado a um cofre, use o comando `remove-tags-from-vault`.

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

Para obter mais informações sobre essa operação de cofre, consulte Remover tags do cofre.

Marcação de cofres usando a API do Amazon Glacier

Você pode adicionar, listar e remover tags usando a API do Amazon Glacier. Para obter exemplos, consulte a seguinte documentação:

Add Tags To Vault (POST tags add)

Adiciona ou atualiza tags do cofre especificado.

Listar tags para cofre (GET tags)

Lista as tags do cofre especificado.

Remove Tags From Vault (POST tags remove)

Remove as tags do cofre especificado.

Seções relacionadas

- Marcação de recursos do Amazon Glacier

Vault Lock do Amazon Glacier

Os tópicos a seguir descrevem como bloquear um cofre no Amazon Glacier e como usar políticas do Vault Lock.

Tópicos

- Visão geral do bloqueio de cofre
- Bloquear um cofre usando a API do Amazon Glacier
- Bloqueando um cofre usando o AWS Command Line Interface
- Bloquear um cofre usando a API do Amazon Glacier

Visão geral do bloqueio de cofre

O Vault Lock do Amazon Glacier permite implantar e impor facilmente controles de conformidade para cofres individuais do Amazon Glacier com uma política do Vault Lock. Você pode especificar controles como "Write Once Read Many" (WORM – Uma gravação e muitas leituras) em uma política de bloqueio do cofre e bloquear a política de edições futuras.

> **Important**
>
> Depois que uma política do Vault Lock é bloqueada, ela não pode mais ser alterada ou excluída.

O Amazon Glacier impõe os controles definidos na política do Vault Lock para ajudar a atingir os objetivos de conformidade. Por exemplo, você pode usar as políticas do Vault Lock para impor a retenção de dados. Você pode implantar uma variedade de controles de conformidade em uma política do Vault Lock usando a linguagem de política AWS Identity and Access Management (IAM). Para obter mais informações sobre políticas do Vault Lock, consulte Políticas do Vault Lock.

A política do Vault Lock é diferente de uma política de acesso de cofre. Ambas as políticas regem os controles de acesso ao cofre. No entanto, uma política do Vault Lock pode ser bloqueada para impedir alterações futuras, o que proporciona uma imposição rígida para os controles de conformidade. Use a política do Vault Lock para implantar controles regulatórios e de conformidade, que normalmente exigem controles rígidos sobre o acesso aos dados.

> **Important**
>
> Recomendamos que primeiro você crie um cofre, preencha uma política do Vault Lock e, em seguida, carregue os arquivos no cofre para que a política seja aplicada a eles.

Por outro lado, você usa uma política de acesso ao cofre para implementar controles de acesso não relacionados à conformidade, temporários e sujeitos a uma modificação frequente. As políticas de bloqueio e acesso ao cofre podem ser usadas juntas. Por exemplo, pode-se implementar regras de retenção de dados baseadas em tempo na política do Vault Lock (deny deletes) e conceder acesso de leitura a terceiros designados ou parceiros de negócios (allow reads) na política de acesso de cofre.

Bloquear um cofre utiliza duas etapas:

1. Inicie o bloqueio anexando uma política do Vault Lock, que define o bloqueio como um estado em andamento e retorna um ID de bloqueio. Enquanto a política estiver em andamento, você tem 24 horas para validar sua política do Vault Lock antes que o ID do bloqueio expire. Para evitar que seu cofre saia do estado em andamento, você deve concluir o processo do Vault Lock dentro dessas 24 horas. Caso contrário, sua política do Vault Lock será excluída.
2. Use o ID de bloqueio para concluir o processo de bloqueio. Se a política do Vault Lock não funcionar conforme o esperado, é possível anular o bloqueio e reiniciar desde o início. Para obter informações sobre como usar a API do Amazon Glacier para bloquear um cofre, consulte Bloquear um cofre usando a API do Amazon Glacier.

Bloquear um cofre usando a API do Amazon Glacier

Para bloquear o cofre com a API do Amazon Glacier, você primeiro chama Initiate Vault Lock (POST lock-policy) com uma política do Vault Lock que especifica os controles que deseja implantar. O `Initiate Vault Lock` anexa a política ao cofre, faz a transição do Vault Lock para o estado em andamento e retorna um ID de bloqueio exclusivo. Depois que o Vault Lock entra no estado em andamento, tem-se 24 horas para concluir o bloqueio chamando Complete Vault Lock (POST lockId) com o ID de bloqueio retornado de `Initiate Vault Lock`.

> **Important**
>
> - Recomendamos que primeiro você crie um cofre, preencha uma política do Vault Lock e, em seguida, carregue os arquivos no cofre para que a política seja aplicada a eles.
> - Depois que a política do Vault Lock for bloqueada, ela não poderá mais ser alterada ou apagada.

Se você não concluir o processo do Vault Lock dentro de 24 horas depois de inserir o estado em andamento, o cofre sairá automaticamente do estado em andamento, e a política de bloqueio será removida. Você pode chamar `Initiate Vault Lock` novamente para instalar uma nova política do Vault Lock e fazer a transição para o estado em andamento.

O estado em andamento oferece a oportunidade de testar a política do Vault Lock antes de efetivar o bloqueio. A política do Vault Lock entra totalmente em vigor durante o estado em andamento como se o cofre tivesse sido bloqueado, exceto por você poder remover a política chamando Abort Vault Lock (DELETE lock-policy). Para ajustar a política, você pode repetir a combinação `Abort Vault`

`Lock/Initiate Vault Lock` quantas vezes forem necessárias a fim de validar as alterações feitas na política do Vault Lock.

Depois de validar a política do Vault Lock, você poderá chamar Complete Vault Lock (POST lockId) com o ID de bloqueio mais recente para concluir o processo de bloqueio. O cofre faz a transição para um estado bloqueado em que a política do Vault Lock é inalterável e não pode mais ser removida chamando-se `Abort Vault Lock`.

Seções relacionadas

- Políticas do Vault Lock
- Abort Vault Lock (DELETE lock-policy)
- Complete Vault Lock (POST lockId)
- Get Vault Lock (GET lock-policy)
- Initiate Vault Lock (POST lock-policy)

Bloqueando um cofre usando o AWS Command Line Interface

Você pode trancar seu cofre usando o. AWS Command Line Interface Isso instalará uma política de bloqueio do Vault Lock no cofre especificado e retornará o ID do bloqueio. Você deve concluir o processo de bloqueio do cofre dentro de 24 horas, caso contrário, a política de bloqueio será removida do cofre.

(Pré-requisito) Configurando o AWS CLI

1. Faça download e configure a AWS CLI. Para obter instruções, consulte os seguintes tópicos no Guia do usuário do AWS Command Line Interface :

   Instalando o AWS Command Line Interface

   Configurando o AWS Command Line Interface
2. Verifique sua AWS CLI configuração inserindo os seguintes comandos no prompt de comando. Esses comandos não fornecem as credenciais explicitamente, de modo que as credenciais do perfil padrão são usadas.

   - Tente usar o comando de ajuda.

```
aws help
```

- Para obter uma lista dos cofres do Amazon Glacier na conta configurada, use o comando `list-vaults`. *123456789012*Substitua pelo seu Conta da AWS ID.

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

- Para ver os dados de configuração atuais do AWS CLI, use o `aws configure list` comando.

```
aws configure list
```

1. Use `initiate-vault-lock` para instalar uma política de bloqueio de cofre e definir o estado de bloqueio do cofre como `InProgress`.

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

2. A configuração de notificação é um documento JSON conforme mostrado no exemplo a seguir. Antes de usar esse comando, substitua *VAULT_ARN* e *Principal* pelos valores apropriados para seu caso de uso.

   Para encontrar o ARN do cofre que você deseja bloquear, é possível usar o comando `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. Depois de iniciar o bloqueio do cofre, você deverá ver o `lockId` retornado.

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

Para concluir o bloqueio do cofre, execute `complete-vault-lock` dentro de 24 horas, caso contrário, a política de bloqueio será removida.

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

Seções relacionadas

- initiate-vault-lock na AWS CLI Command Reference
- list-vaults na Referência de comandos da AWS CLI
- complete-vault-lock na AWS CLI Command Reference
- Políticas do Vault Lock
- Abort Vault Lock (DELETE lock-policy)
- Complete Vault Lock (POST lockId)
- Get Vault Lock (GET lock-policy)
- Initiate Vault Lock (POST lock-policy)

Bloquear um cofre usando a API do Amazon Glacier

O Vault Lock do Amazon Glacier permite implantar e impor facilmente controles de conformidade para cofres individuais do Amazon Glacier com uma política do Vault Lock. Para obter mais informações sobre o Vault Lock do Amazon Glacier, consulte Controle de acesso do Amazon Glacier com políticas do Vault Lock.

> **Important**
>
> - Recomendamos que primeiro você crie um cofre, preencha uma política do Vault Lock e, em seguida, carregue os arquivos no cofre para que a política seja aplicada a eles.
> - Depois que a política do Vault Lock for bloqueada, ela não poderá mais ser alterada ou apagada.

Para iniciar uma política do Vault Lock em seu cofre usando o console do Amazon Glacier

Inicie o bloqueio anexando uma política do Vault Lock ao seu cofre, que define o bloqueio como um estado em andamento e retorna um ID de bloqueio. Enquanto a política estiver em andamento, você tem 24 horas para validar sua política do Vault Lock antes que o ID do bloqueio expire.

1. Faça login Console de gerenciamento da AWS e abra o console do Amazon Glacier em casahttps://console.aws.amazon.com/glacier/.
2. Em Selecionar uma região, selecione uma no seletor Região da AWS de região.
3. No painel de navegação à esquerda, selecione Cofres.
4. Na página Cofres, escolha Criar cofre.
5. Crie um novo cofre.

   > **Important**
   >
   > Recomendamos que primeiro você crie um cofre, preencha uma política do Vault Lock e, em seguida, carregue seus arquivos no cofre para que a política seja aplicada a eles.

6. Escolha seu novo cofre na lista Cofres.
7. Escolha a guia Políticas do cofre.
8. Na seção Política do Vault Lock, escolha Iniciar política do Vault Lock.
9. Na página Iniciar política do Vault Lock, especifique os controles de retenção de registros em sua política do Vault Lock em formato de texto na caixa de texto padrão.

   > **Note**
   >
   > Você pode especificar os controles de retenção de registros em uma política do Vault Lock em formato de texto e iniciar o Vault Lock chamando a operação `Initiate Vault Lock` da API ou por meio da interface interativa no console do Amazon Glacier. Para obter informações sobre como formatar sua política do Vault Lock, consulte Exemplos de políticas do Vault Lock do Amazon Glacier.

10. Escolha Salvar alterações.
11. Na caixa de diálogo Registrar ID do Vault Lock, copie o ID de bloqueio e salve-o em um local seguro.

> **Important**
>
> Depois que a política do Vault Lock for iniciada, você terá 24 horas para validar a política e concluir o processo de bloqueio. Para concluir o processo de bloqueio, forneça o ID do bloqueio. Se não for fornecido em 24 horas, o ID de bloqueio expira e sua política em andamento será excluída.

12. Depois de salvar seu ID de bloqueio em um local seguro, escolha Fechar.
13. Teste sua política do Vault Lock nas próximas 24 horas. Se a política estiver funcionando conforme o esperado, escolha Concluir a política do Vault Lock.
14. Na caixa de diálogo Concluir Vault Lock, marque a caixa de seleção para confirmar que a conclusão do processo de política do Vault Lock é irreversível.
15. Insira o ID de bloqueio fornecido na caixa de texto.
16. Escolha Concluir Vault Lock.

Trabalhar com arquivos no Amazon Glacier

Um arquivo é qualquer objeto, como uma foto, um vídeo ou um documento, armazenado por você em um cofre. É uma unidade básica de armazenamento no Amazon Glacier. Cada arquivo tem uma ID exclusiva e uma descrição opcional. Ao fazer upload de um arquivo, o Amazon Glacier retorna uma resposta que inclui um ID de arquivo. Esse ID de arquivamento é exclusivo na AWS região em que o arquivo está armazenado. Este é um ID de arquivo de exemplo.

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

O arquivo IDs tem 138 bytes. Ao fazer upload de um arquivo, você pode apresentar uma descrição opcional. Você pode recuperar um arquivo usando o ID, mas não a descrição.

> **Important**
>
> O Amazon Glacier oferece um console de gerenciamento. Você pode usar o console para criar e excluir cofres. No entanto, todas as outras interações com o Amazon Glacier exigem que você use a AWS Command Line Interface (CLI) ou escreva o código. Por exemplo, para fazer upload de dados, como fotos, vídeos e outros documentos, você deve usar AWS CLI ou escrever código para fazer solicitações, usando a API REST diretamente ou usando a Amazon SDKs. Para obter mais informações sobre o uso do Amazon Glacier com o AWS CLI, acesse AWS CLI Referência do Amazon Glacier. Para instalar o AWS CLI, acesse AWS Command Line Interface.

Tópicos

- Operações de arquivamento no Amazon Glacier
- Manter metadados de arquivo no lado do cliente
- Carregar um arquivo no Amazon Glacier
- Baixando um arquivo no Amazon Glacier
- Excluir um arquivo no Amazon Glacier

Operações de arquivamento no Amazon Glacier

O Amazon Glacier dá suporte às seguintes operações de arquivo básicas: upload, download e exclusão. Fazer download de um arquivo é uma operação assíncrona.

Carregar um arquivo no Amazon Glacier

Você pode fazer upload de um arquivo em uma única operação ou fazer upload dele em partes. A chamada à API usada por você para fazer upload de um arquivo em partes é conhecida como Multipart Upload. Para obter mais informações, consulte Carregar um arquivo no Amazon Glacier.

> **Important**
>
> O Amazon Glacier oferece um console de gerenciamento. Você pode usar o console para criar e excluir cofres. No entanto, todas as outras interações com o Amazon Glacier exigem que você use a (AWS Command Line Interface CLI) ou escreva código. Por exemplo, para fazer upload de dados, como fotos, vídeos e outros documentos, você deve usar AWS CLI ou escrever código para fazer solicitações, usando a API REST diretamente ou usando a Amazon SDKs. Para obter mais informações sobre o uso do Amazon Glacier com o AWS CLI, acesse AWS CLI Referência do Amazon Glacier. Para instalar o AWS CLI, acesse AWS Command Line Interface.

Encontre um ID de arquivo no Amazon Glacier

Você pode obter o ID de arquivo fazendo download do inventário do cofre que contém o arquivo. Para obter mais informações sobre como fazer download do inventário de cofre, consulte Fazer download de um inventário de cofre no Amazon Glacier.

Baixando um arquivo no Amazon Glacier

Fazer download de um arquivo é uma operação assíncrona. Você deve primeiro iniciar um trabalho para fazer download de um arquivo específico. Depois de receber a solicitação de trabalho, o Amazon Glacier prepara o arquivo para download. Depois que o trabalho é concluído, você pode fazer download dos dados do arquivo. Por causa da natureza assíncrona do trabalho, você pode solicitar que o Amazon Glacier envie uma notificação para um tópico do Amazon Simple Notification Service (Amazon SNS) quando o trabalho for concluído. Você pode especificar um tópico do SNS

para cada solicitação de trabalho individual ou configurar o cofre para enviar uma notificação quando ocorrerem eventos específicos. Para obter mais informações sobre como fazer download de um arquivo, consulte Baixando um arquivo no Amazon Glacier.

Excluir um arquivo no Amazon Glacier

O Amazon Glacier fornece uma chamada de API que pode ser usada para excluir um arquivo por vez. Para obter mais informações, consulte Excluir um arquivo no Amazon Glacier.

Atualizar um arquivo no Amazon Glacier

Depois de fazer upload de um arquivo, você não poderá atualizar o conteúdo nem a descrição. A única maneira como você pode atualizar o conteúdo do arquivo ou a descrição é excluindo o arquivo e fazendo upload de outro arquivo. Sempre que você faz upload de um arquivo, o Amazon Glacier retorna um ID de arquivo exclusivo.

Manter metadados de arquivo no lado do cliente

Exceto para a descrição de arquivo opcional, o Amazon Glacier não dá suporte a metadados adicionais dos arquivos. Quando se faz upload de um arquivo, o Amazon Glacier atribui um ID, uma sequência de caracteres invisível, do qual não se pode inferir nenhum significado sobre o arquivo. Você pode manter metadados sobre os arquivos no lado do cliente. Os metadados podem incluir o nome do arquivo e algumas outras informações significativas sobre o arquivo.

> **Note**
>
> Se for um cliente do Amazon Simple Storage Service (Amazon S3), você saberá que, ao fazer upload de um objeto em um bucket, você pode atribuir ao objeto uma chave, como `MyDocument.txt` ou `SomePhoto.jpg`. No Amazon Glacier, não é possível atribuir um chave de objeto aos arquivos com upload.

Se você mantém metadados de arquivamento do lado do cliente, observe que o Amazon Glacier mantém um inventário do cofre que inclui o arquivo IDs e todas as descrições fornecidas durante o upload do arquivo. Ocasionalmente, você pode fazer download do inventário de cofre para reconciliar todos os problemas no banco de dados no lado do cliente dos metadados do arquivo. No entanto, o Amazon Glacier faz o inventário de cofre quase que diariamente. Quando você solicita um inventário

de cofre, o Amazon Glacier retorna o inventário preparado mais recentemente, um snapshot point-in-time.

Carregar um arquivo no Amazon Glacier

O Amazon Glacier fornece um console de gerenciamento que você pode usar para criar e excluir cofres. No entanto, não é possível fazer upload de arquivos para o Amazon Glacier usando o console de gerenciamento. Para fazer upload de dados, como fotos, vídeos e outros documentos, você deve usar o código AWS CLI ou escrever o código para fazer solicitações, usando diretamente a API REST ou usando a Amazon SDKs.

Para obter informações sobre o uso do Amazon Glacier com o AWS CLI, acesse AWS CLI Referência do Amazon Glacier. Para instalar o AWS CLI, acesse AWS Command Line Interface. Os tópicos Fazer upload a seguir descrevem como fazer upload de arquivos para o Amazon Glacier usando o Amazon SDK para Java, o Amazon SDK para .NET e a API REST.

Tópicos

- Opções de upload de um arquivo para o Amazon Glacier
- Fazer upload de um arquivo em uma única operação
- Fazer upload de arquivos grandes em partes (Multipart Upload)

Opções de upload de um arquivo para o Amazon Glacier

Dependendo do tamanho de dados que estão no upload, o Amazon Glacier oferece as seguintes opções:

- Fazer upload de arquivos em uma única operação – Em uma única operação, você pode fazer upload de arquivos de 1 byte até 4 GB. No entanto, recomendamos que os clientes do Amazon Glacier usem Upload Multiparte para fazer upload de arquivos maiores que 100 MB. Para obter mais informações, consulte Fazer upload de um arquivo em uma única operação.
- Fazer upload de arquivos em partes – Usando a multipart upload API, você pode fazer upload de arquivos grandes de até aproximadamente 40.000 GB (10.000 * 4 GB).

  A chamada da API de multipart upload foi projetada para melhorar a experiência de upload para arquivos maiores. Você pode fazer upload de arquivos em partes. O upload dessas partes pode ser feito de maneira independente, em qualquer ordem, e em paralelo. Em caso de falha de um upload da parte, você precisa refazer upload dessa parte, e não de todo o arquivo. Você pode usar

um multipart upload para arquivos de 1 byte até aproximadamente 40.000 GB. Para obter mais informações, consulte Fazer upload de arquivos grandes em partes (Multipart Upload).

> **Important**
>
> O inventário de cofre do Amazon Glacier é atualizado somente uma vez por dia. Ao fazer upload de um arquivo, você não verá imediatamente o novo arquivo adicionado ao cofre (no console ou na lista de inventários de cofre obtido por download) até o inventário de cofre ter sido atualizado.

Usando o AWS Snowball Edge serviço

AWS Snowball Edge acelera a transferência de grandes quantidades de dados para dentro e para fora do AWS uso de dispositivos de propriedade da Amazon, contornando a Internet. Para obter mais informações, consulte a página de detalhes do AWS Snowball Edge.

Para fazer upload de dados existentes para o Amazon Glacier, considere usar um dos tipos de dispositivo de AWS Snowball Edge para importar dados para o Amazon S3, e movê-los para a classe de armazenamento do Amazon Glacier para fins de arquivamento usando regras de ciclo de vida. Quando você faz a transição de objetos do Amazon S3 para a classe de armazenamento do Amazon Glacier, o Amazon S3 usa internamente o Amazon Glacier no armazenamento durável por um custo menor. Embora os objetos sejam armazenados no Amazon Glacier, eles continuam sendo objetos do Amazon S3 que você gerencia no Amazon S3, e não é possível acessá-los diretamente por meio do Amazon Glacier.

Para obter mais informações sobre a configuração do ciclo de vida do Amazon S3 e a transição de objetos para a classe de armazenamento do Amazon Glacier, consulte Gerenciamento do ciclo de vida de objetos e Transição de objetos no Guia do usuário do Amazon Simple Storage Service.

Fazer upload de um arquivo em uma única operação

Conforme descrito em Carregar um arquivo no Amazon Glacier, você pode fazer upload de arquivos menores em uma única operação. No entanto, recomendamos que os clientes do Amazon Glacier usem Upload Multiparte para fazer upload de arquivos maiores que 100 MB.

Tópicos

- Carregando um arquivo em uma única operação usando o AWS Command Line Interface

- Carregando um arquivamento em uma única operação usando o AWS SDK para Java
- Carregando um arquivo em uma única operação usando o AWS SDK para .NET no Amazon Glacier
- Fazer upload de um arquivo em uma única operação usando a API REST

Carregando um arquivo em uma única operação usando o AWS Command Line Interface

Você pode fazer upload de um arquivo no Amazon Glacier (Amazon Glacier) usando AWS Command Line Interface o ().AWS CLI

Tópicos

- (Pré-requisito) Configurando o AWS CLI
- Exemplo: Carregar um arquivo usando o AWS CLI

(Pré-requisito) Configurando o AWS CLI

1. Faça download e configure a AWS CLI. Para obter instruções, consulte os seguintes tópicos no Guia do usuário do AWS Command Line Interface :

   Instalando o AWS Command Line Interface

   Configurando o AWS Command Line Interface
2. Verifique sua AWS CLI configuração inserindo os seguintes comandos no prompt de comando. Esses comandos não fornecem as credenciais explicitamente, de modo que as credenciais do perfil padrão são usadas.

   - Tente usar o comando de ajuda.

     ```
     aws help
     ```

   - Para obter uma lista dos cofres do Amazon Glacier na conta configurada, use o comando `list-vaults`. *123456789012*Substitua pelo seu Conta da AWS ID.

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

   - Para ver os dados de configuração atuais do AWS CLI, use o `aws configure list` comando.

```
aws configure list
```

Exemplo: Carregar um arquivo usando o AWS CLI

Para fazer o upload de um arquivo, você deve ter um cofre criado. Para obter mais informações sobre a criação de cofres, consulte Criar um cofre no Amazon Glacier.

1. Use o comando `upload-archive` para adicionar um arquivo a um cofre existente. No exemplo abaixo, substitua `vault name` e `account ID`. Para o parâmetro `body`, especifique um caminho para o arquivo que você deseja carregar.

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

2. Saída esperada:

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

   Quando concluído, o comando exibirá o ID do arquivo, a soma de verificação e a localização no Amazon Glacier. Para obter mais informações sobre o comando upload-arquivo, consulte upload-arquivo na Referência de comandos da AWS CLI .

Carregando um arquivamento em uma única operação usando o AWS SDK para Java

Tanto o alto quanto o baixo nível APIs fornecidos pelo Amazon SDK for Java fornecem um método para carregar um arquivo.

Tópicos

- Fazendo o upload de um arquivo usando a API de alto nível do AWS SDK para Java

- Carregando um arquivo em uma única operação usando a API de baixo nível do AWS SDK para Java

Fazendo o upload de um arquivo usando a API de alto nível do AWS SDK para Java

A classe `ArchiveTransferManager` da API de nível superior fornece o método `upload`, que você pode usar para fazer upload de um arquivo em um cofre.

> **Note**
>
> Você pode usar o método `upload` para fazer upload de arquivos grandes ou pequenos. Dependendo do tamanho do arquivo que você estiver fazendo upload, esse método determina se é necessário fazer upload dele em uma única operação ou usar a multipart upload API para fazer upload do arquivo em partes.

Exemplo: fazer o upload de um arquivo usando a API de alto nível do AWS SDK para Java

O exemplo de código Java a seguir faz upload de um arquivo em um cofre (`examplevault`) na Região (`us-west-2`) Oeste dos EUA (Oregon). Para obter uma lista de AWS regiões e endpoints compatíveis, consulteAcessar o Amazon Glacier.

Para step-by-step obter instruções sobre como executar esse exemplo, consulteExecutar exemplos do Java para o Amazon Glacier usando o Eclipse. É preciso atualizar o código conforme mostrado com o nome do cofre cujo upload você deseja fazer e o nome do arquivo cujo upload quer fazer.

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);
        }
    }
}
```

Carregando um arquivo em uma única operação usando a API de baixo nível do AWS SDK para Java

A API de nível inferior fornece métodos para todas as operações de arquivo. Veja a seguir as etapas para carregar um arquivo usando o AWS SDK para Java.

1. Crie uma instância da classe `AmazonGlacierClient` (o cliente).

   Você precisa especificar uma AWS região na qual deseja fazer o upload do arquivo. Todas as operações que você executa usando esse cliente se aplicam a essa AWS região.
2. Forneça informações sobre a solicitação criando uma instância da classe `UploadArchiveRequest`.

Além dos dados cujo upload deseja fazer, você precisa fornecer uma soma de verificação (hash de árvore SHA-256) da carga útil, o nome do cofre, o tamanho do conteúdo dos dados e o ID da conta.

Se você não fornecer um ID da conta, o ID da conta associado às credenciais fornecidas por você para assinar a solicitação será pressuposto. Para obter mais informações, consulte Usando o AWS SDK para Java com o Amazon Glacier.

3. Execute o método `uploadArchive` fornecendo o objeto de solicitação como um parâmetro.

   Em resposta, o Amazon Glacier retorna um ID do arquivo recém-carregado.

O trecho de código Java a seguir ilustra as etapas anteriores.

```
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());
```

Exemplo: fazer upload de um arquivo em uma única operação usando a API de baixo nível do AWS SDK para Java

O exemplo de código Java a seguir usa o AWS SDK para Java para carregar um arquivo em um vault (`examplevault`). Para step-by-step obter instruções sobre como executar esse exemplo, consulteExecutar exemplos do Java para o Amazon Glacier usando o Eclipse. É preciso atualizar o código conforme mostrado com o nome do cofre cujo upload você deseja fazer e o nome do arquivo cujo upload quer fazer.

```
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);
        }
```

```
    }
}
```

Carregando um arquivo em uma única operação usando o AWS SDK para .NET no Amazon Glacier

Tanto o alto quanto o baixo nível APIs fornecidos pelo Amazon SDK para .NET fornecem um método para carregar um arquivo em uma única operação.

Tópicos

- Fazendo o upload de um arquivo usando a API de alto nível do AWS SDK para .NET
- Carregando um arquivo em uma única operação usando a API de baixo nível do AWS SDK para .NET

Fazendo o upload de um arquivo usando a API de alto nível do AWS SDK para .NET

A classe `ArchiveTransferManager` da API de nível superior fornece o método `Upload` que você pode usar para fazer upload de um arquivo em um cofre.

> **Note**
>
> Você pode usar o método `Upload` para fazer upload de arquivos pequenos ou grandes. Dependendo do tamanho do arquivo que você estiver fazendo upload, esse método determina se é necessário fazer upload dele em uma única operação ou usar a API de multipart upload para fazer upload do arquivo em partes.

Exemplo: fazer upload de um arquivo usando a API de alto nível do AWS SDK para .NET

O exemplo de código C# a seguir faz upload de um arquivo em um cofre (`examplevault`) na Região Oeste dos EUA (Oregon).

Para step-by-step obter instruções sobre como executar esse exemplo, consulteExecutar exemplos de código. É preciso atualizar o código conforme mostrado com o nome do arquivo cujo upload deseja fazer.

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();
    }
  }
}
```

Carregando um arquivo em uma única operação usando a API de baixo nível do AWS SDK para .NET

A API de nível inferior fornece métodos para todas as operações de arquivo. Veja a seguir as etapas para carregar um arquivo usando o AWS SDK para .NET.

1. Crie uma instância da classe `AmazonGlacierClient` (o cliente).

Você precisa especificar uma AWS região na qual deseja fazer o upload do arquivo. Todas as operações que você executa usando esse cliente se aplicam a essa AWS região.

2. Forneça informações sobre a solicitação criando uma instância da classe `UploadArchiveRequest`.

   Além dos dados cujo upload deseja fazer, você precisa fornecer uma soma de verificação (hash de árvore SHA-256) da carga útil, o nome do cofre e o ID da conta.

   Se você não fornecer um ID da conta, o ID da conta associado às credenciais fornecidas por você para assinar a solicitação será pressuposto. Para obter mais informações, consulte Como usar o AWS SDK para .NET com o Amazon Glacier.
3. Execute o método `UploadArchive` fornecendo o objeto de solicitação como um parâmetro.

   Em resposta, o Amazon Glacier retorna um ID do arquivo recém-carregado.

Exemplo: fazer upload de um arquivo em uma única operação usando a API de baixo nível do AWS SDK para .NET

O exemplo de código do C# a seguir ilustra as etapas anteriores. O exemplo usa o AWS SDK para .NET para carregar um arquivo em um cofre (`examplevault`).

> **Note**
>
> Para obter informações sobre a API REST subjacente para fazer upload de um arquivo em uma única solicitação, consulte Upload Archive (POST archive).

Para step-by-step obter instruções sobre como executar esse exemplo, consulteExecutar exemplos de código. É preciso atualizar o código conforme mostrado com o nome do arquivo cujo upload deseja fazer.

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;
      }
    }
```

```
    }
}
```

Fazer upload de um arquivo em uma única operação usando a API REST

Você pode usar a chamada de API Upload Archive para fazer upload de um arquivo em uma única operação. Para obter mais informações, consulte Upload Archive (POST archive).

Fazer upload de arquivos grandes em partes (Multipart Upload)

Tópicos

- Processo Multipart Upload
- Fatos rápidos
- Fazendo upload de arquivos grandes usando o AWS CLI
- Fazer upload de arquivos grandes em partes usando o Amazon SDK para Java
- Fazendo upload de arquivos grandes usando o AWS SDK para .NET
- Fazer upload de arquivos grandes em partes usando a API REST

Processo Multipart Upload

Conforme descrito em Carregar um arquivo no Amazon Glacier, recomendamos que os clientes do Amazon Glacier usem upload multiparte para fazer upload de arquivos maiores que 100 mebibytes (MiB).

1. Iniciar multipart upload

   Quando você envia uma solicitação para iniciar um upload multiparte, o Amazon Glacier retorna um ID desse upload, que é um identificador exclusivo. Todas as operações de multipart upload subsequentes exigirão esse ID. Esse ID não vai expirar por pelo menos 24 horas depois que o Amazon Glacier concluir o trabalho.

   Na solicitação para iniciar um multipart upload, especifique o tamanho da parte em número de bytes. Cada parte do upload, exceto a última, deve ser desse tamanho.

   > **Note**
   >
   > Você não precisa saber o tamanho do arquivo geral ao usar o multipart upload. Isso significa que é possível usar multipart uploads nos casos em que, ao começar a carregar

o arquivo, o tamanho do arquivo não é conhecido. É necessário decidir apenas o tamanho da parte ao iniciar o multipart upload.

Na solicitação de multipart upload de iniciação, você também pode fornecer uma descrição de arquivo opcional.

2. Partes de upload

   Para cada solicitação de upload da parte, você deve incluir o ID de multipart upload obtido na etapa 1. Na solicitação, você também deve especificar o intervalo de conteúdo, em bytes, identificando a posição da parte no arquivo final. O Amazon Glacier usará as informações do intervalo de conteúdo para montar o arquivo na sequência apropriada. Como você fornece o intervalo de conteúdo para cada parte do upload, ele determina a posição da parte na montagem final do arquivo e, assim, pode fazer upload de partes em qualquer ordem. Você também pode fazer upload de partes em paralelo. Se você fizer upload de uma nova parte usando o mesmo intervalo de conteúdo como uma parte carregada anteriormente, a parte cujo upload foi feito anteriormente será substituída.

3. Concluir (ou anular) o Multipart Upload

   Depois de fazer upload de todas as partes do arquivo, você usará a operação completa. Mais uma vez, você deve especificar o ID de upload na solicitação. O Amazon Glacier cria um arquivo concatenando partes em ordem crescente com base no intervalo de conteúdo fornecido por você. A resposta do Amazon Glacier a uma solicitação para upload multiparte completo inclui um ID do arquivo recém-criado. Caso uma descrição de arquivo opcional tenha sido fornecida na solicitação Iniciar Upload Multiparte, o Amazon Glacier a associará ao arquivo montado. Depois de concluir com êxito um multipart upload, você não poderá se referir ao ID de multipart upload. Isso significa você não pode acessar partes associadas ao ID de multipart upload.

   Se anular um multipart upload, você não poderá fazer mais upload de partes usando esse ID de upload multipartes. Todo o armazenamento consumido por todas as partes associadas ao multipart upload anulado é liberado. Se algum upload de parte estiver em andamento, ele ainda poderá ser bem-sucedido ou falhar mesmo depois da interrupção.

Operações de multipart upload adicionais

O Amazon Glacier fornece as seguintes chamadas adicionais de API de upload multiparte.

- List Parts: usando essa operação, você pode listar as partes de um multipart upload. Isso retorna informações sobre as partes cujo upload você fez para um multipart upload. Para cada solicitação de partes da lista, o Amazon Glacier retorna informações para até 1.000 partes. Se houver mais partes a serem listadas para o multipart upload, o resultado será paginado, e um marcador será retornado na resposta na qual a lista deve continuar. Você precisa enviar solicitações adicionais para recuperar partes subsequentes. Observe que a lista de partes retornada não inclui partes que não tiveram o upload concluído.
- List Multipart Uploads: usando essa operação, você pode obter uma lista de multipart upload em andamento. Um multipart upload em andamento é um upload que você iniciou, mas que ainda não concluiu nem interrompeu. Para cada solicitação de multipart uploads da lista, o Amazon Glacier retorna até 1.000 multipart uploads. Se houver mais multipart uploads a serem listadas, o resultado será paginado, e um marcador será retornado na resposta na qual a lista deve continuar. Você precisa enviar solicitações adicionais para recuperar os multipart uploads restantes.

Fatos rápidos

A tabela a seguir fornece especificações básicas do multipart upload.

Item	Especificação
Tamanho de arquivo máximo	10.000 x 4 gibibytes (GiB)
Número máximo de partes por upload	10.000
Tamanho da parte	De 1 MiB a 4 GiB, a última parte pode ser < 1 MiB. Você especifica o valor de tamanho em bytes. O tamanho da peça deve ser um mebibyte (1024 kibibytes [KiB]) multiplicado por uma potência de 2. Por exemplo, `1048576` (1 MiB), `2097152` (2 MiB), `4194304` (4 MiB), `8388608` (8 MiB).
Número máximo de partes retornadas em uma solicitação de listagem de partes	1.000

Item	Especificação
Número máximo de multipart uploads retornados em uma solicitação de listagem de multipart uploads	1.000

Fazendo upload de arquivos grandes usando o AWS CLI

Você pode fazer upload de um arquivo no Amazon Glacier (Amazon Glacier) usando o AWS Command Line Interface ().AWS CLI Para melhorar a experiência de upload de arquivos maiores, o Amazon Glacier fornece várias operações de API compatíveis com uploads multiparte. Ao usar essas operações de API, você pode fazer upload de arquivos em partes. O upload dessas partes pode ser feito de maneira independente, em qualquer ordem, e em paralelo. Em caso de falha de um upload da parte, você precisa refazer o upload dessa parte, e não de todo o arquivo. Você pode usar um multipart uploads para arquivos de 1 byte até aproximadamente 40.000 GiB.

Para obter mais informações sobre uploads multiparte do Amazon Glacier, consulte Fazer upload de arquivos grandes em partes (Multipart Upload).

Tópicos

- (Pré-requisito) Configurando o AWS CLI
- (Pré-requisito) Instalar Python
- (Pré-requisito) Crie um cofre do Amazon Glacier
- Exemplo: Fazer upload de arquivos grandes em partes usando o AWS CLI

(Pré-requisito) Configurando o AWS CLI

1. Faça download e configure a AWS CLI. Para obter instruções, consulte os seguintes tópicos no Guia do usuário do AWS Command Line Interface :

   Instalando o AWS Command Line Interface

   Configurando o AWS Command Line Interface
2. Verifique sua AWS CLI configuração inserindo os seguintes comandos no prompt de comando. Esses comandos não fornecem as credenciais explicitamente, de modo que as credenciais do perfil padrão são usadas.

- Tente usar o comando de ajuda.

```
aws help
```

- Para obter uma lista dos cofres do Amazon Glacier na conta configurada, use o comando `list-vaults`. *123456789012*Substitua pelo seu Conta da AWS ID.

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

- Para ver os dados de configuração atuais do AWS CLI, use o `aws configure list` comando.

```
aws configure list
```

(Pré-requisito) Instalar Python

Para concluir um upload de várias partes, você deve calcular o hash da SHA256 árvore do arquivo que você está carregando. Fazer isso é diferente de calcular o hash da SHA256 árvore do arquivo que você deseja carregar. Para calcular o hash da SHA256 árvore do arquivo que você está carregando, você pode usar Java C# (com .NET) ou. Python Neste exemplo, usaremos Python. Para obter instruções sobre como usar Java ou C#, consulteComputar somas de verificação.

Para obter informações sobre como instalar ou atualizar Python, consulte Instalar a Python no Guia do usuário Boto3.

(Pré-requisito) Crie um cofre do Amazon Glacier

Para usar o exemplo a seguir, você deve ter pelo menos um cofre do Amazon Glacier criado. Para obter mais informações sobre a criação de cofres, consulte Criar um cofre no Amazon Glacier.

Exemplo: Fazer upload de arquivos grandes em partes usando o AWS CLI

Neste exemplo, você criará um arquivo e usará operações de API de multipart upload para fazer upload desse arquivo, em partes, no Amazon Glacier.

> **Important**
>
> Antes de iniciar este procedimento, verifique se você executou todas as etapas pré-requisitos. Para fazer upload de um arquivo, você deve ter um cofre criado, AWS CLI

configurado e estar preparado para usar Java C# ou Python calcular um hash de SHA256 árvore.

O procedimento a seguir usa os `complete-multipart-upload` AWS CLI comandos `initiate-multipart-uploadupload-multipart-part`, e.

Para obter mais informações sobre esses comandos, consulte initiate-multipart-upload,upload-multipart-part, na seção complete-multipart-upload da Referência de comandos da AWS CLI .

1. Use o comando initiate-multipart-upload para criar um recurso de multipart upload. Na solicitação para iniciar um upload multipartes, especifique o tamanho da parte em número de bytes. Cada parte do upload, exceto a última, deve ser desse tamanho. Você não precisa saber o tamanho do arquivo geral ao usar o upload multipartes. No entanto, você precisará do tamanho total, em bytes, de cada parte ao concluir o upload na etapa final.

   No comando a seguir, substitua os valores dos parâmetros `--vault-name` e `--account-ID` por suas próprias informações. Esse comando especifica que você fará o upload de um arquivo com um tamanho de 1 mebibyte (MiB) (1024 x 1024 bytes) por arquivo. Substitua esse valor de parâmetro `--part-size`, se necessário.

   ```
   aws glacier initiate-multipart-upload --vault-name awsexamplevault --part-
   size 1048576 --account-id 123456789012
   ```

   Saída esperada:

   ```
   {
   "location": "/123456789012/vaults/awsexamplevault/multipart-uploads/uploadId",
   "uploadId": "uploadId"
   }
   ```

   Quando concluído, o comando exibirá o ID de upload e a localização do recurso de upload multiparte no Amazon Glacier. Você usará este ID de carregamento nas etapas subsequentes.

2. Neste exemplo, você pode usar os comandos a seguir para criar um arquivo de 4,4 MiB, dividi-lo em partes de 1 MiB e fazer upload de cada parte. Para fazer upload de seus próprios arquivos, você pode seguir um procedimento semelhante de dividir seus dados em partes e fazer o upload de cada parte.

Linux ou macOS

O comando a seguir cria um arquivo de 4,4 MiB, chamado `file_to_upload`, no Linux ou no macOS.

```
mkfile -n 9000b file_to_upload
```

Windows

O comando a seguir cria um arquivo de 4,4 MiB, chamado `file_to_upload`, no Windows.

```
fsutil file createnew file_to_upload 4608000
```

3. Em seguida, você dividirá esse arquivo em pedaços de 1 MiB.

```
split -b 1048576 file_to_upload chunk
```

Você agora tem os cinco blocos a seguir. Os quatro primeiros têm 1 MiB e o último tem aproximadamente 400 kibibytes (KiB).

```
chunkaa
chunkab
chunkac
chunkad
chunkae
```

4. Use o comando upload-multipart-part para carregar uma parte de um arquivo. Você pode fazer upload de arquivos em partes. Você também pode fazer upload de partes em paralelo. Você pode fazer upload de até 10.000 partes de um multipart upload.

   No comando a seguir, substitua os valores dos parâmetros `--vault-name`, `--account-ID` e `--upload-id`. O ID de upload deve corresponder ao ID fornecido como saída do comando `initiate-multipart-upload`. O parâmetro `--range` especifica que você fará o upload de uma peça com um tamanho de 1 MiB (1024 x 1024 bytes). Esse tamanho deve corresponder ao que você especificou no comando `initiate-multipart-upload`. Ajuste esse valor de tamanho, se necessário. O parâmetro `--body` especifica o nome da parte que você está carregando.

```
aws glacier upload-multipart-part --body chunkaa --range='bytes 0-1048575/*' --
vault-name awsexamplevault --account-id 123456789012 --upload-id upload_ID
```

Se for bem-sucedido, o comando produzirá uma saída que contém a soma de verificação da peça carregada.

5. Execute o comando `upload-multipart-part` novamente para carregar as partes restantes do multipart upload. Atualize os valores dos parâmetros `--range` e `--body` de cada comando para que correspondam à parte que você está carregando.

```
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**
>
> O valor do parâmetro do comando final `--range` é menor porque a parte final do nosso upload é menor que 1 MiB. Se for bem-sucedido, cada comando produzirá uma saída que contém a soma de verificação de cada parte carregada.

6. Em seguida, você montará o arquivo e finalizará o upload. Você deve incluir o tamanho total e o hash da SHA256 árvore do arquivo.

   Para calcular o hash da SHA256 árvore do arquivo, você pode usar Java C# ou. Python Neste exemplo, usaremos Python. Para obter instruções sobre como usar Java ou C#, consulteComputar somas de verificação.

   Crie o Python arquivo `checksum.py` e insira o código a seguir. Se necessário, substitua o nome do arquivo original.

```
from botocore.utils import calculate_tree_hash

checksum = calculate_tree_hash(open('file_to_upload', 'rb'))
print(checksum)
```

7. Execute `checksum.py` para calcular o hash SHA256 da árvore. O hash a seguir pode não corresponder à sua saída.

```
$ python3 checksum.py
$ 3d760edb291bfc9d90d35809243de092aea4c47b308290ad12d084f69988ae0c
```

8. Use o comando complete-multipart-upload para finalizar o upload do arquivo. Substitua os valores dos parâmetros `--vault-name`, `--account-ID`, `--upload-ID` e `--checksum`. O valor do parâmetro `--archive` especifica o tamanho total, em bytes, do arquivo. Esse valor deve ser a soma de todos os tamanhos das partes individuais cujo upload você fez. Substitua esse valor, se necessário.

```
aws glacier complete-multipart-upload --archive-size 4608000 --vault-
name awsexamplevault --account-id 123456789012 --upload-id upload_ID --
checksum checksum
```

   Quando concluído, o comando exibirá o ID, a soma de verificação e a localização do arquivo no Amazon Glacier.

Fazer upload de arquivos grandes em partes usando o Amazon SDK para Java

Tanto o alto quanto o baixo nível APIs fornecidos pelo Amazon SDK for Java fornecem um método para fazer o upload de um grande arquivo (consulte). Carregar um arquivo no Amazon Glacier

- A API de nível superior fornece um método que você pode usar para fazer upload de arquivos de qualquer tamanho. Dependendo do arquivo que você estiver fazendo upload, o método faz upload de um arquivo em uma única operação ou usa o suporte a upload multiparte no Amazon Glacier para fazer upload do arquivo em partes.
- A API de nível inferior é mapeada junto à implementação de REST subjacente. Dessa forma, ela fornece um método para fazer upload de arquivos menores em uma operação e um grupo de métodos que dão suporte a multipart upload para arquivos maiores. Esta seção explica como fazer upload de arquivos grandes em partes usando a API de nível inferior.

Para obter mais informações sobre alto e baixo nível APIs, consulte. Usando o AWS SDK para Java com o Amazon Glacier

Tópicos

- Carregando arquivos grandes em partes usando a API de alto nível do AWS SDK para Java
- Faça upload de arquivos grandes em partes usando a API de baixo nível do AWS SDK para Java

Carregando arquivos grandes em partes usando a API de alto nível do AWS SDK para Java

Você pode usar os mesmos métodos da API de nível superior para fazer upload de arquivos grandes ou pequenos. Com base no tamanho do arquivo, os métodos da API de nível superior decidem se é necessário fazer upload do arquivo em uma única operação ou usar a API de upload multiparte fornecida pelo Amazon Glacier. Para obter mais informações, consulte Fazendo o upload de um arquivo usando a API de alto nível do AWS SDK para Java.

Faça upload de arquivos grandes em partes usando a API de baixo nível do AWS SDK para Java

Para controle granular do upload, você pode usar a API de nível inferior, em que pode configurar a solicitação e processar a resposta. Estas são as etapas para fazer upload de arquivos grandes em partes usando o AWS SDK para Java.

1. Crie uma instância da classe `AmazonGlacierClient` (o cliente).

   Você precisa especificar uma AWS região na qual deseja salvar o arquivo. Todas as operações que você executa usando esse cliente se aplicam a essa AWS região.
2. Inicie multipart upload chamando o método `initiateMultipartUpload`.

   Você precisa fornecer o nome do cofre no qual deseja fazer upload do arquivo, o tamanho da parte que deseja usar para fazer upload das partes do arquivo e uma descrição opcional. Você precisa fornecer essas informações criando uma instância da classe `InitiateMultipartUploadRequest`. Em resposta, o Amazon Glacier retorna um ID de upload.
3. Faça upload de partes chamando o método `uploadMultipartPart`.

   Para cada parte cujo upload faz, você precisa fornecer o nome do cofre, o intervalo de bytes no arquivo montado final cujo upload será feito nessa parte, a soma de verificação dos dados da parte e o ID de upload.
4. Conclua multipart upload chamando o método `completeMultipartUpload`.

Você precisa fornecer o ID de upload, a soma de verificação de todo o arquivo, o tamanho do arquivo (tamanho combinado de todas as partes cujo upload você fez) e o nome do cofre. O Amazon Glacier cria o arquivo a partir das partes carregadas e retorna um ID de arquivo.

Exemplo: Fazer upload de um arquivo grande em um Parts usando o AWS SDK para Java

O exemplo de código Java a seguir usa o AWS SDK para Java para carregar um arquivo em um vault (`examplevault`). Para step-by-step obter instruções sobre como executar esse exemplo, consulteExecutar exemplos do Java para o Amazon Glacier usando o Eclipse. É preciso atualizar o código conforme mostrado com o nome do arquivo cujo upload deseja fazer.

> **Note**
>
> Este exemplo é válido para tamanhos de parte de 1 MB a 1 GB. No entanto, o Amazon Glacier dá suporte a tamanhos de parte de até 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();
    }
}
```

Fazendo upload de arquivos grandes usando o AWS SDK para .NET

Tanto o alto quanto o baixo nível APIs fornecidos pelo Amazon SDK para .NET fornecem um método para carregar grandes arquivos em partes (consulte). Carregar um arquivo no Amazon Glacier

- A API de nível superior fornece um método que você pode usar para fazer upload de arquivos de qualquer tamanho. Dependendo do arquivo que você estiver fazendo upload, o método faz upload de um arquivo em uma única operação ou usa o suporte a upload multiparte no Amazon Glacier para fazer upload do arquivo em partes.
- A API de nível inferior é mapeada junto à implementação de REST subjacente. Dessa forma, ela fornece um método para fazer upload de arquivos menores em uma operação e um grupo de métodos que dão suporte a multipart upload para arquivos maiores. Esta seção explica como fazer upload de arquivos grandes em partes usando a API de nível inferior.

Para obter mais informações sobre alto e baixo nível APIs, consulte. Como usar o AWS SDK para .NET com o Amazon Glacier

Tópicos

- Carregando arquivos grandes em partes usando a API de alto nível do AWS SDK para .NET
- Carregando arquivos grandes em partes usando a API de baixo nível do AWS SDK para .NET

Carregando arquivos grandes em partes usando a API de alto nível do AWS SDK para .NET

Você pode usar os mesmos métodos da API de nível superior para fazer upload de arquivos grandes ou pequenos. Com base no tamanho do arquivo, os métodos da API de nível superior decidem se é necessário fazer upload do arquivo em uma única operação ou usar a API de upload multiparte fornecida pelo Amazon Glacier. Para obter mais informações, consulte Fazendo o upload de um arquivo usando a API de alto nível do AWS SDK para .NET.

Carregando arquivos grandes em partes usando a API de baixo nível do AWS SDK para .NET

Para controle granular do upload, você pode usar a API de nível inferior, em que pode configurar a solicitação e processar a resposta. Estas são as etapas para fazer upload de arquivos grandes em partes usando o AWS SDK para .NET.

1. Crie uma instância da classe `AmazonGlacierClient` (o cliente).

   Você precisa especificar uma AWS região na qual deseja salvar o arquivo. Todas as operações que você executa usando esse cliente se aplicam a essa AWS região.
2. Inicie multipart upload chamando o método `InitiateMultipartUpload`.

   Você precisa fornecer o nome do cofre para o qual deseja fazer upload do arquivo, o tamanho da parte que deseja usar para fazer upload das partes do arquivo e uma descrição opcional. Você precisa fornecer essas informações criando uma instância da classe `InitiateMultipartUploadRequest`. Em resposta, o Amazon Glacier retorna um ID de upload.
3. Faça upload de partes chamando o método `UploadMultipartPart`.

   Para cada parte cujo upload faz, você precisa fornecer o nome do cofre, o intervalo de bytes no arquivo montado final cujo upload será feito nessa parte, a soma de verificação dos dados da parte e o ID de upload.
4. Conclua o multipart upload chamando o método `CompleteMultipartUpload`.

   Você precisa fornecer o ID de upload, a soma de verificação de todo o arquivo, o tamanho do arquivo (tamanho combinado de todas as partes cujo upload você fez) e o nome do cofre. O Amazon Glacier cria o arquivo a partir das partes carregadas e retorna um ID de arquivo.

Exemplo: fazer upload de um arquivo grande em partes usando o Amazon SDK para .NET

O exemplo de código C# a seguir usa o AWS SDK para .NET para carregar um arquivo em um vault ()`examplevault`. Para step-by-step obter instruções sobre como executar esse exemplo, consulteExecutar exemplos de código. Você precisa atualizar o código conforme mostrado com o nome de um arquivo cujo upload deseja fazer.

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;
    }
  }
}
```

Fazer upload de arquivos grandes em partes usando a API REST

Conforme descrito em Fazer upload de arquivos grandes em partes (Multipart Upload), o multipart upload se refere a um conjunto de operações que permite fazer upload de um arquivo em partes e realizar operações relacionadas. Para obter mais informações sobre essas operações, consulte os seguintes tópicos de referência da API:

- Initiate Multipart Upload (POST multipart-uploads)
- Upload Part (PUT uploadID)
- Complete Multipart Upload (POST uploadID)
- Abort Multipart Upload (DELETE uploadID)
- List Parts (GET uploadID)
- List Multipart Uploads (GET multipart-uploads)

Baixando um arquivo no Amazon Glacier

O Amazon Glacier oferece um console de gerenciamento que pode ser usado para criar e excluir cofres. No entanto, você não pode fazer download de arquivos do Amazon Glacier usando o console de gerenciamento. Para baixar dados, como fotos, vídeos e outros documentos, você deve usar o AWS Command Line Interface (AWS CLI) ou escrever código para fazer solicitações, usando a API REST diretamente ou usando AWS SDKs o.

Para obter informações sobre o uso do Amazon Glacier com o AWS CLI, consulte a AWS CLI Referência do Amazon Glacier. Para instalar o AWS CLI, consulte AWS Command Line Interface. Os tópicos a seguir descrevem como baixar arquivos para o Amazon Glacier usando a AWS SDK para Java, a e a API AWS SDK para .NET REST do Amazon Glacier.

Tópicos

- Recuperação de arquivos do Amazon Glacier
- Baixando um arquivo no Amazon Glacier usando o AWS SDK para Java
- Baixando um arquivo no Amazon Glacier usando o AWS SDK para .NET
- Baixando um arquivo grande usando processamento paralelo com Python
- Fazer download de um arquivo usando a API REST
- Baixando um arquivo no Amazon Glacier usando o AWS CLI

Recuperação de arquivos do Amazon Glacier

Recuperar um arquivo do Amazon Glacier é uma operação assíncrona na qual você primeiro inicia um trabalho e, em seguida, faz download da saída depois de concluí-lo. Para iniciar um trabalho de recuperação de arquivamento, você usa a operação da API Initiate Job (trabalhos POST) REST ou o equivalente no AWS CLI, ou o. AWS SDKs

Tópicos

- Opções de recuperação de arquivos
- Recuperações de arquivo no intervalo

Recuperar um arquivo do Amazon Glacier é um processo de duas etapas. Esta é uma visão geral do processo de implantação:

Para recuperar um arquivo

1. Inicie um trabalho de recuperação de arquivo.

   a. Obtenha o ID do arquivo que você deseja recuperar. Você pode obter o ID de arquivo de um inventário de cofre. Você pode obter o ID do arquivo com a API REST AWS CLI, ou AWS SDKs. Para obter mais informações, consulte Fazer download de um inventário de cofre no Amazon Glacier.

   b. Inicie um trabalho solicitando que o Amazon Glacier prepare um arquivo inteiro ou uma parte dele para download subsequente usando a operação Initiate Job (trabalhos POST).

   Quando você inicia um trabalho, o Amazon Glacier cria um ID de trabalho na resposta e o executa de maneira assíncrona. (Você não poderá fazer download da saída do trabalho depois da conclusão desse trabalho conforme descrito na Etapa 2.)

   > **Important**
   >
   > Somente para recuperações padrão, uma política de recuperação de dados pode causar uma falha na solicitação `Initiate Job` do trabalho de recuperação de inicialização com uma exceção `PolicyEnforcedException` . Para obter mais informações sobre políticas de recuperação de dados, consulte Políticas de recuperação de dados do Amazon Glacier. Para obter mais informações sobre a exceção `PolicyEnforcedException`, consulte Respostas de erro.

   Quando necessário, é possível restaurar grandes segmentos dos dados armazenados no Amazon Glacier. Para obter informações sobre as Classes de armazenamento do Amazon Glacier, consulte Classes de armazenamento de objetos no Guia do Desenvolvedor do Amazon Simple Storage Service.

2. Após a conclusão do trabalho, faça download dos bytes usando a operação Get Job Output (GET output).

   Você pode fazer download de todos os bytes ou especificar um intervalo de bytes para fazer download somente de uma parte da saída do trabalho. Para uma saída maior, fazer download da saída em blocos ajudará se houver uma falha no download, como uma falha de rede. Se obtiver a saída do trabalho em uma única solicitação e houver uma falha na rede, você precisará reiniciar o download da saída desde o início. No entanto, se fizer download da saída em blocos, em caso de alguma falha, você precisará somente reiniciar o download da parte menor, e não de toda a saída.

O Amazon Glacier deve concluir o trabalho para você obter o resultado. Após a conclusão, um trabalho não vai expirar por pelo menos 24 horas, o que significa que você pode fazer download da saída dentro do período de 24 horas depois do término do trabalho. A restauração pode expirar a qualquer momento após 24 horas da conclusão do trabalho. Para determinar se o trabalho está concluído, verifique o status usando uma das seguintes opções:

- Aguardar a notificação de conclusão do trabalho: você pode especificar um tópico do Amazon Simple Notification Service (Amazon SNS) no qual o Amazon Glacier poderá publicar uma notificação depois que o trabalho for concluído. O Amazon Glacier enviará uma notificação somente depois de concluir o trabalho.

  Você pode especificar um tópico do Amazon SNS para um trabalho ao iniciá-lo. Além de especificar um tópico do Amazon SNS na solicitação de trabalho, se o cofre tiver uma configuração de notificação definida para eventos de recuperação do arquivo, o Amazon Glacier também publicará uma notificação para esse tópico do SNS. Para obter mais informações, consulte Configurar notificações do cofre no Amazon Glacier.
- Solicitar informações do trabalho explicitamente: você também pode usar a operação da API `Describe Job` do Amazon Glacier (Trabalho de descrição (GET JobID)) para pesquisar periodicamente informações do trabalho. No entanto, recomendamos usar as notificações do Amazon SNS.

> **Note**
>
> As informações obtidas por você usando a notificação do Amazon SNS são as mesmas recebidas quando se chama `Describe Job` a operação API.

Opções de recuperação de arquivos

Você pode especificar um dos seguintes ao iniciar um trabalho para recuperar um arquivo com base nos requisitos de tempo e custo de acesso. Para obter informações sobre preços de recuperação, consulte Preços do Amazon Glacier.

- Expedited: as recuperações expressas permitem que você acesse rapidamente os dados armazenados na classe de armazenamento S3 Glacier Flexible Retrieval ou no nível S3 Intelligent-Tiering Archive Access quando solicitações urgentes ocasionais para restauração de arquivos são necessárias. Exceto para os arquivos maiores (mais de 250 MB), os dados acessados usando-se recuperações expressas costumam ser disponibilizados dentro de 1 a 5 minutos. A capacidade provisionada garante que sua capacidade de recuperação para recuperações expressas esteja disponível quando você precisar dela. Para obter mais informações, consulte Capacidade provisionada.
- Padrão: as recuperações padrão permitem acessar qualquer um dos arquivos em várias horas. As recuperações padrão normalmente são concluídas dentro de 3 a 5 horas. Padrão é a opção assumida para solicitações de recuperação que não especificam a opção de recuperação.
- Em massa: as recuperações em massa são a opção de recuperação de menor custo do Amazon Glacier que você pode usar para recuperar grandes quantidades de dados, até mesmo petabytes, em um dia. As recuperações em massa (Bulk retrievals) normalmente são concluídas dentro de 5 a 12 horas.

A tabela a seguir resume as opções de recuperação de arquivos. Para obter mais informações sobre preços, consulte Preços do Amazon Glacier.

Para fazer uma `Bulk` recuperação `ExpeditedStandard`, ou, defina o elemento de `Tier` solicitação na solicitação de operação da API RestoreObjectREST com a opção desejada ou o equivalente em AWS Command Line Interface (AWS CLI) ou AWS SDKs. Se você adquiriu a capacidade provisionada, todas as recuperações expressas serão automaticamente fornecidas por meio de sua capacidade provisionada.

Capacidade provisionada

A capacidade provisionada ajuda a garantir que sua capacidade de recuperação para recuperações expressas esteja disponível quando você precisar dela. Cada unidade de capacidade permite que pelo menos três recuperações expressas possam ser realizadas a cada 5 minutos e fornece até 150 megabytes por segundo () de taxa de transferência de recuperação. MBps

Se sua workload exigir acesso altamente confiável e previsível a um subconjunto de seus dados em minutos, será necessário adquirir a capacidade de recuperação provisionada. Sem capacidade provisionada, as recuperações expressas são normalmente aceitas, exceto em raras situações de demanda excepcionalmente alta. Contudo, se precisar de acesso a recuperações expressas em todas as circunstâncias, você deve comprar a capacidade de recuperação provisionada.

Comprar capacidade provisionada

Você pode comprar unidades de capacidade provisionada usando o console do Amazon Glacier, Purchase Provisioned Capacity (POST provisioned-capacity) a operação da API REST, AWS SDKs o ou o. AWS CLI Para obter informações sobre a definição de preços da capacidade provisionada, consulte Definição de preços do Amazon Glacier.

Uma unidade de capacidade provisionada dura um mês a partir da data e hora da compra.

Se a data de início é dia 31 de um mês, a data de expiração será o último dia do mês seguinte. Por exemplo, se a data de início é dia 31 de agosto, a data de expiração será dia 30 de setembro. Se a data de início é dia 31 de janeiro, a data de expiração será dia 28 de fevereiro.

Para comprar capacidade provisionada usando o console do Amazon Glacier

1. Faça login Console de gerenciamento da AWS e abra o console do Amazon Glacier em casahttps://console.aws.amazon.com/glacier/.
2. No painel de navegação à esquerda, escolha Configurações de recuperação de dados.
3. Em Unidades de capacidade provisionada (PCUs), escolha Comprar PCU. A caixa de diálogo Comprar PCU é exibida.
4. Se quiser comprar capacidade provisionada, entre **`confirm`** na caixa Para confirmar a compra.
5. Escolha Comprar PCU.

Recuperações de arquivo no intervalo

Ao recuperar um arquivo do Amazon Glacier, você pode especificar um intervalo, ou mesmo um segmento, do arquivo a ser recuperado. O padrão é recuperar o arquivo todo. Especificar um intervalo de bytes pode ser útil quando você deseja fazer o seguinte:

- Gerenciar os downloads de dados: o Amazon Glacier permite que os dados recuperados sejam baixados por 24 horas após a conclusão da solicitação de recuperação. Portanto, convém

recuperar somente partes do arquivo, de maneira que você possa gerenciar a programação de downloads dentro da janela de download indicada.

- Recuperar uma parte segmentada de um arquivo grande: por exemplo, suponhamos que você tenha agregado muitos arquivos anteriormente e feito upload deles como um único arquivo e agora queira recuperar alguns deles. Nesse caso, pode-se especificar um intervalo do arquivo que contenha os arquivos que lhe interessam usando uma solicitação de recuperação. Ou você pode iniciar várias solicitações de recuperação, cada uma com um intervalo para um ou mais arquivos.

Ao iniciar um trabalho de recuperação usando recuperações de intervalo, você deve fornecer um intervalo alinhado a megabytes. Em outras palavras, o intervalo de bytes pode começar em zero (o início do arquivo) ou em qualquer intervalo de 1 MB posterior (1 MB, 2 MB, 3 MB e assim por diante).

O final do intervalo pode ser o final do arquivo ou qualquer intervalo de 1 MB maior que o início do intervalo. Além disso, se você quiser obter valores da soma de verificação ao fazer download dos dados (depois da conclusão do trabalho de recuperação), o intervalo solicitado por você na iniciação do trabalho também deverá estar alinhado ao tree-hash. Somas de verificação são uma maneira de você garantir que os dados não tenham sido corrompidos durante a transmissão. Para obter mais informações sobre os alinhamentos a megabytes e ao hash de árvore, consulte Receber somas de verificação durante o download de dados.

Baixando um arquivo no Amazon Glacier usando o AWS SDK para Java

Tanto o alto quanto o baixo nível APIs fornecidos pelo Amazon SDK for Java fornecem um método para baixar um arquivo.

Tópicos

- Baixando um arquivo usando a API de alto nível do AWS SDK para Java
- Baixando um arquivo usando a API de baixo nível do AWS SDK para Java

Baixando um arquivo usando a API de alto nível do AWS SDK para Java

A classe `ArchiveTransferManager` da API de nível superior fornece o método `download` que você pode usar para fazer download de um arquivo.

> **Important**
>
> A classe `ArchiveTransferManager` cria um tópico do Amazon Simple Notification Service (Amazon SNS) e uma fila do Amazon Simple Queue Service (Amazon SQS) que está inscrita nesse tópico. Em seguida, ela inicia o trabalho de recuperação do arquivo e sonda a fila em busca do arquivo disponível. Assim que o arquivo estiver disponível, o download começará. Para obter informações sobre tempos de recuperação, consulte Opções de recuperação de arquivos.

Exemplo: Baixar um arquivo usando a API de alto nível do AWS SDK para Java

O exemplo de código do Java a seguir faz download de um arquivo de um cofre (`examplevault`) na região Oeste dos EUA (Oregon) (`us-west-2`).

Para step-by-step obter instruções sobre como executar esse exemplo, consulteExecutar exemplos do Java para o Amazon Glacier usando o Eclipse. Você precisa atualizar o código conforme mostrado com um ID de arquivo existente e o caminho do arquivo local onde deseja salvar o arquivo obtido por download.

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);
        }
    }
}
```

Baixando um arquivo usando a API de baixo nível do AWS SDK para Java

Estas são as etapas para recuperar um inventário de cofre usando a API de nível inferior do AWS SDK para Java .

1. Crie uma instância da classe `AmazonGlacierClient` (o cliente).

   Você precisa especificar uma AWS região de onde deseja baixar o arquivo. Todas as operações que você executa usando esse cliente se aplicam a essa AWS região.

2. Inicie um trabalho `archive-retrieval` executando o método `initiateJob`.

   Forneça informações do trabalho, como o ID do arquivo cujo download deseja fazer e o tópico do Amazon SNS no qual deseja que o Amazon Glacier publique uma mensagem de conclusão do trabalho criando uma instância da classe `InitiateJobRequest`. Em resposta, o Amazon

Glacier retorna um ID de trabalho. A resposta está disponível em uma instância da classe `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();
```

Você também pode especificar um intervalo de bytes para solicitar ao Amazon Glacier que prepare somente uma parte do arquivo. Por exemplo, você pode atualizar a solicitação anterior adicionando a instrução a seguir para solicitar que o Amazon Glacier prepare somente a parte de 1 MB a 2 MB do arquivo.

```
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. Aguarde a conclusão do trabalho .

   Você deve aguardar até a saída do trabalho estar pronta para download. Se você tiver definido uma configuração de notificação no cofre identificando um tópico do Amazon Simple Notification

Service (Amazon SNS) ou especificado um tópico do Amazon SNS quando tiver iniciado um trabalho, o Amazon Glacier enviará uma mensagem para esse tópico depois de concluir o trabalho.

Você também pode sondar o Amazon Glacier chamando o método `describeJob` para determinar o status de conclusão do trabalho. Apesar disso, usar um tópico do Amazon SNS para notificação é a abordagem recomendada.

4. Faça download da saída do trabalho (dados de arquivo) executando o método `getJobOutput`.

   Você fornece as informações da solicitação, como o ID de trabalho e o nome do cofre, criando uma instância da classe `GetJobOutputRequest`. A saída retornada pelo Amazon Glacier está disponível no objeto `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.
```

   O trecho de código anterior faz download de toda a saída do trabalho. Opcionalmente, você pode recuperar somente uma parte da saída ou fazer download de toda a saída em blocos menores especificando o intervalo de bytes em `GetJobOutputRequest`.

```
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 ****");
```

   Em resposta à chamada `GetJobOutput`, o Amazon Glacier retorna a soma de verificação da parte dos dados obtidos por download, se determinadas condições forem atendidas. Para obter mais informações, consulte Receber somas de verificação durante o download de dados.

   Para verificar se não há erros no download, você pode computar a soma de verificação no lado do cliente e compará-la com a soma de verificação enviada pelo Amazon Glacier na resposta.

Para um trabalho de recuperação de arquivamento com o intervalo opcional especificado, quando você obtém a descrição do trabalho, ela inclui a soma de verificação do intervalo que você está recuperando (). SHA256 TreeHash Você pode usar esse valor para verificar a precisão de todo o intervalo de bytes cujo download fará depois. Por exemplo, se iniciar um trabalho para recuperar um intervalo alinhado ao hash de árvore e fizer download de saída em blocos, de maneira que cada uma das solicitações `GetJobOutput` retorne uma soma de verificação, você poderá computar a soma de verificação de cada parte cujo download você faz no lado do cliente e o hash de árvore. Você pode compará-la com a soma de verificação que o Amazon Glacier retorna em resposta à sua solicitação de descrição de trabalho para verificar se todo o intervalo de bytes que você baixou é igual ao intervalo de bytes armazenado no Amazon Glacier.

Para obter um exemplo funcional, consulte Exemplo 2: Recuperação de um arquivamento usando a API de baixo nível da saída AWS SDK para Java—Download em partes.

Exemplo 1: Recuperação de um arquivamento usando a API de baixo nível do AWS SDK para Java

O exemplo de código do Java a seguir faz download de um arquivo do cofre especificado. Depois que o trabalho for concluído, o exemplo fará download de toda a saída em uma única chamada `getJobOutput`. Para obter um exemplo de como fazer download de saída em blocos, consulte Exemplo 2: Recuperação de um arquivamento usando a API de baixo nível da saída AWS SDK para Java—Download em partes.

O exemplo realiza as seguintes tarefas:

- Cria um tópico do Amazon Simple Notification Service (Amazon SNS).

  O Amazon Glacier enviará uma notificação para esse tópico depois de concluir o trabalho.
- Cria uma fila do Amazon Simple Queue Service (Amazon SQS)

  O exemplo anexa uma política à fila para permitir que o tópico do Amazon SNS publique mensagens na fila.
- Inicia um trabalho para fazer download do arquivo especificado.

  Na solicitação de trabalho, o tópico do Amazon SNS criado é especificado para que o Amazon Glacier possa publicar uma notificação no tópico depois de concluir o trabalho.
- Verifica periodicamente a fila do Amazon SQS em busca de uma mensagem que contenha o ID do trabalho.

Se houver uma mensagem, analise o JSON e verifique se o trabalho foi concluído com êxito. Se for o caso, faça download do arquivo.

- Realiza uma limpeza excluindo o tópico do Amazon SNS e a fila do Amazon SQS criada por ele.

```
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));
    }
}
```

Exemplo 2: Recuperação de um arquivamento usando a API de baixo nível da saída AWS SDK para Java—Download em partes

O exemplo de código do Java a seguir recupera um arquivo do Amazon Glacier. O exemplo de código faz download da saída do trabalho em blocos especificando um intervalo de bytes em um objeto `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));
    }
}
```

Baixando um arquivo no Amazon Glacier usando o AWS SDK para .NET

Tanto o alto quanto o baixo nível APIs fornecidos pelo Amazon SDK para .NET fornecem um método para baixar um arquivo.

Tópicos

- Baixando um arquivo usando a API de alto nível do AWS SDK para .NET
- Baixando um arquivo usando a API de baixo nível do AWS SDK para .NET

Baixando um arquivo usando a API de alto nível do AWS SDK para .NET

A classe `ArchiveTransferManager` da API de nível superior fornece o método `Download` que você pode usar para fazer download de um arquivo.

> **Important**
>
> A classe `ArchiveTransferManager` cria um tópico do Amazon Simple Notification Service (Amazon SNS) e uma fila do Amazon Simple Queue Service (Amazon SQS) que está inscrita nesse tópico. Em seguida, ela inicia o trabalho de recuperação do arquivo e sonda a fila em busca do arquivo disponível. Assim que o arquivo estiver disponível, o download começará. Para obter informações sobre tempos de recuperação, consulte Opções de recuperação de arquivos

Exemplo: Baixar um arquivo usando a API de alto nível do AWS SDK para .NET

O exemplo de código C# a seguir faz download de um arquivo de um cofre (`examplevault`) na região Oeste dos EUA (Oregon).

Para step-by-step obter instruções sobre como executar esse exemplo, consulteExecutar exemplos de código. Você precisa atualizar o código conforme mostrado com um ID de arquivo existente e o caminho do arquivo local onde deseja salvar o arquivo obtido por download.

```
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);
      }
    }
  }
}
```

Baixando um arquivo usando a API de baixo nível do AWS SDK para .NET

Estas são as etapas para fazer download de um arquivo do Amazon Glacier usando a API de nível inferior de AWS SDK para .NET.

1. Crie uma instância da classe `AmazonGlacierClient` (o cliente).

   Você precisa especificar uma AWS região de onde deseja baixar o arquivo. Todas as operações que você executa usando esse cliente se aplicam a essa AWS região.

2. Inicie um trabalho `archive-retrieval` executando o método `InitiateJob`.

   Você fornece informações do trabalho, como o ID do arquivo cujo download você deseja fazer e o tópico do Amazon SNS no qual deseja que o Amazon Glacier publique uma mensagem de conclusão do trabalho criando uma instância da classe `InitiateJobRequest`. Em resposta, o

Amazon Glacier retorna um ID de trabalho. A resposta está disponível em uma instância da classe `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;
```

Você também pode especificar um intervalo de bytes a fim de solicitar que o Amazon Glacier prepare somente uma parte do arquivo, conforme mostrado na solicitação a seguir. A solicitação especifica que o Amazon Glacier prepare somente o segmento de 1 a 2 MB do arquivo.

```
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. Aguarde a conclusão do trabalho .

   Você deve aguardar até a saída do trabalho estar pronta para download. Se você tiver definido uma configuração de notificação no cofre identificando um tópico do Amazon Simple Notification Service (Amazon SNS) ou especificado um tópico do Amazon SNS quando tiver iniciado um trabalho, o Amazon Glacier enviará uma mensagem para esse tópico depois de concluir o trabalho. O exemplo de código indicado na seção a seguir usa o Amazon SNS para que o Amazon Glacier publique uma mensagem.

   Você também pode sondar o Amazon Glacier chamando o método `DescribeJob` para determinar o status de conclusão do trabalho. Apesar disso, usar um tópico do Amazon SNS para notificação é a abordagem recomendada.

4. Faça download da saída do trabalho (dados de arquivo) executando o método `GetJobOutput`.

   Você fornece as informações da solicitação, como o ID de trabalho e o nome do cofre, criando uma instância da classe `GetJobOutputRequest`. A saída retornada pelo Amazon Glacier está disponível no objeto `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);
  }
}
```

   O trecho de código anterior faz download de toda a saída do trabalho. Opcionalmente, você pode recuperar somente uma parte da saída ou fazer download de toda a saída em blocos menores especificando o intervalo de bytes em `GetJobOutputRequest`.

```
GetJobOutputRequest getJobOutputRequest = new GetJobOutputRequest()
{
  JobId = jobId,
  VaultName = vaultName
};
getJobOutputRequest.SetRange(0, 1048575); // Download only the first 1 MB chunk of
 the output.
```

Em resposta à chamada `GetJobOutput`, o Amazon Glacier retorna a soma de verificação da parte dos dados obtidos por download, se determinadas condições forem atendidas. Para obter mais informações, consulte Receber somas de verificação durante o download de dados.

Para verificar se não há erros no download, você pode computar a soma de verificação no lado do cliente e compará-la com a soma de verificação enviada pelo Amazon Glacier na resposta.

Para um trabalho de recuperação de arquivamento com o intervalo opcional especificado, quando você obtém a descrição do trabalho, ela inclui a soma de verificação do intervalo que você está recuperando (SHA256TreeHash). Você pode usar esse valor para verificar ainda mais a precisão de todo o intervalo de bytes que você baixará posteriormente. Por exemplo, se iniciar um trabalho para recuperar um intervalo alinhado ao hash de árvore e fizer download de saída em blocos, de maneira que cada uma das solicitações `GetJobOutput` retorne uma soma de verificação, você poderá computar a soma de verificação de cada parte cujo download você faz no lado do cliente e o hash de árvore. Você pode compará-la com a soma de verificação que o Amazon Glacier retorna em resposta à sua solicitação de descrição de trabalho para verificar se todo o intervalo de bytes que você baixou é igual ao intervalo de bytes armazenado no Amazon Glacier.

Para obter um exemplo funcional, consulte Exemplo 2: Recuperação de um arquivamento usando a API de baixo nível da saída AWS SDK para .NET—Download em partes.

Exemplo 1: Recuperando um arquivo usando a API de baixo nível do AWS SDK para .NET

O exemplo de código do C# a seguir faz download de um arquivo do cofre especificado. Depois que o trabalho for concluído, o exemplo fará download de toda a saída em uma única chamada `GetJobOutput`. Para obter um exemplo de como fazer download de saída em blocos, consulte Exemplo 2: Recuperação de um arquivamento usando a API de baixo nível da saída AWS SDK para .NET—Download em partes.

O exemplo realiza as seguintes tarefas:

- Configure o tópico do Amazon Simple Notification Service (Amazon SNS)

  O Amazon Glacier enviará uma notificação para esse tópico depois de concluir o trabalho.
- Configure a Fila do Amazon Simple Queue Service (Amazon SQS)

  O exemplo anexa uma política à fila para permitir que o tópico do Amazon SNS publique mensagens.
- Inicia um trabalho para fazer download do arquivo especificado.

  Na solicitação de trabalho, o exemplo especifica o tópico do Amazon SNS, de maneira que o Amazon Glacier possa enviar uma mensagem depois de concluir o trabalho.
- Verifica periodicamente a fila do Amazon SQS em busca de uma mensagem.

  Se houver uma mensagem, analise o JSON e verifique se o trabalho foi concluído com êxito. Se for o caso, faça download do arquivo. O exemplo de código usa a biblioteca do JSON.NET (consulte JSON.NET) para analisar o JSON.
- Limpa excluindo o tópico do Amazon SNS e a fila do Amazon SQS criada por ele.

```
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);
      }
    }
  }
}
```

Exemplo 2: Recuperação de um arquivamento usando a API de baixo nível da saída AWS SDK para .NET—Download em partes

O exemplo de código do C# a seguir recupera um arquivo do Amazon Glacier. O exemplo de código faz download da saída do trabalho em blocos especificando o intervalo de bytes em um objeto `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);
      }
    }
  }
}
```

Baixando um arquivo grande usando processamento paralelo com Python

Este tópico descreve como baixar um arquivo grande do Amazon S3 Glacier (S3 Glacier) usando processamento paralelo com Python. Essa abordagem permite que você baixe arquivos de qualquer tamanho de forma confiável, dividindo-os em partes menores que podem ser processadas de forma independente.

Visão geral do

O script Python fornecido nesse exemplo realiza as seguintes tarefas:

1. Configura os AWS recursos necessários (tópico do Amazon SNS e filas do Amazon SQS) para notificações
2. Inicia um trabalho de recuperação de arquivo com o Amazon Glacier
3. Monitora uma fila do Amazon SQS em busca de notificações de conclusão de trabalho
4. Divide o grande arquivo em partes gerenciáveis
5. Baixa partes em paralelo usando vários threads de trabalho
6. Salva cada pedaço em disco para posterior remontagem

Pré-requisitos

Antes de começar, você deve ter o seguinte:

- Python 3.6 ou superior instalado.
- AWS SDK para Python (Boto3) instalado
- AWS credenciais configuradas com as permissões apropriadas para Amazon Glacier, Amazon SNS e Amazon SQS
- Espaço em disco suficiente para armazenar os pedaços de arquivo baixados

Exemplo: baixando um arquivo usando processamento paralelo com Python

O script Python a seguir demonstra como fazer o download de um grande arquivo do Amazon Glacier usando processamento paralelo:

```
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)
```

Uso de um script

Para usar esse script, siga estas etapas:

1. Substitua os valores de espaço reservado no script por suas informações específicas:
   - *output_file_path*: Diretório onde os arquivos fragmentados serão salvos
   - *vault_name*: Nome do seu cofre S3 Glacier
   - *notify_queue_name*: Nome da fila de notificação de trabalho
   - *chunk_download_queue_name*: Nome da fila de download do fragmento
   - *sns_topic_name*: Nome do tópico do SNS
   - *region*: AWS região onde seu cofre está localizado
   - *archive_id*: ID do arquivo a ser recuperado
2. Execute o script :

   ```
   python download_large_archive.py
   ```
3. Depois que todos os fragmentos forem baixados, você poderá combiná-los em um único arquivo usando um comando como:

   ```
   cat /path/to/chunks/*.chunk > complete_archive.file
   ```

Considerações importantes

Ao usar esse script, lembre-se do seguinte:

- A recuperação de arquivo do S3 Glacier pode levar várias horas, dependendo do nível de recuperação selecionado.
- O script é executado indefinidamente, pesquisando continuamente as filas. Você pode adicionar uma condição de enceramento com base em seus requisitos específicos.
- Certifique-se de ter espaço em disco suficiente para armazenar todos os fragmentos do seu arquivo.

- Se o script for interrompido, você poderá reiniciá-lo com `retrieve_archive=False` para continuar baixando fragmentos sem iniciar um novo trabalho de recuperação.
- Ajuste os *`workers`* parâmetros *`chunk_size`* e com base na largura de banda da rede e nos recursos do sistema.
- AWS As cobranças padrão se aplicam às recuperações do Amazon S3, ao uso do Amazon SNS e do Amazon SQS.

Fazer download de um arquivo usando a API REST

Para fazer download de um arquivo usando a API REST

Fazer download de um arquivo é um processo de duas etapas.

1. Inicie um trabalho do tipo `archive-retrieval`. Para obter mais informações, consulte Initiate Job (trabalhos POST).
2. Depois que o trabalho for concluído, faça download dos dados de arquivo. Para obter mais informações, consulte Get Job Output (GET output).

Baixando um arquivo no Amazon Glacier usando o AWS CLI

Você pode baixar arquivos no Amazon Glacier (Amazon Glacier) usando AWS Command Line Interface o ().AWS CLI

Tópicos

- (Pré-requisito) Configurando o AWS CLI
- Exemplo: Baixe um arquivo usando o AWS CLI

(Pré-requisito) Configurando o AWS CLI

1. Faça download e configure a AWS CLI. Para obter instruções, consulte os seguintes tópicos no Guia do usuário do AWS Command Line Interface :

   Instalando o AWS Command Line Interface

   Configurando o AWS Command Line Interface

2. Verifique sua AWS CLI configuração inserindo os seguintes comandos no prompt de comando. Esses comandos não fornecem as credenciais explicitamente, de modo que as credenciais do perfil padrão são usadas.

   - Tente usar o comando de ajuda.

     ```
     aws help
     ```

   - Para obter uma lista dos cofres do Amazon Glacier na conta configurada, use o comando `list-vaults`. *123456789012*Substitua pelo seu Conta da AWS ID.

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

   - Para ver os dados de configuração atuais do AWS CLI, use o `aws configure list` comando.

     ```
     aws configure list
     ```

Exemplo: Baixe um arquivo usando o AWS CLI

> **Note**
>
> Para baixar seus arquivos, você deve saber seus IDs de arquivo. As etapas 1 a 4 recuperar seus IDs de arquivo. Se você já conhece os IDs de arquivo que deseja baixar, vá para a etapa 5.

1. Use o comando `initiate-job` para iniciar um trabalho de recuperação de inventário. O relatório de inventário listará seus IDs de arquivo.

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

   Saída esperada:

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

```
    "jobId": "*** jobid ***"
}
```

2. Use o comando `describe-job` para verificar o status do comando de trabalho anterior.

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

Saída esperada:

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

3. Aguarde a conclusão do trabalho .

Você deve aguardar até a saída do trabalho estar pronta para download. Se você definiu uma configuração de notificação no cofre ou especificou um tópico do Amazon Simple Notification Service (Amazon SNS) ao iniciar o trabalho, o Amazon Glacier envia uma mensagem para o tópico depois de concluir o trabalho.

É possível definir a configuração de notificação para eventos específicos no cofre. Para obter mais informações, consulte Configurar notificações do cofre no Amazon Glacier. O Amazon Glacier enviará uma mensagem para o tópico do SNS especificado sempre que o evento determinado ocorrer.

4. Quando estiver concluído, use o comando `get-job-output` para fazer download do trabalho de recuperação para o arquivo `output.json`. Esse arquivo conterá seus IDs de arquivo.

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

Esse comando produz um arquivo com os campos a seguir.

```
{
"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. Use o comando `initiate-job` para iniciar o processo de recuperação de cada arquivo a partir de um cofre. Você precisará especificar o parâmetro do trabalho conforme a `archive-retrieval` mostrada abaixo.

```
aws glacier initiate-job --vault-name awsexamplevault --account-id 111122223333
 --job-parameters="{\"Type\":\"archive-retrieval\",\"ArchiveId\":\"*** archiveId
 ***\"}"
```

6. Aguarde a conclusão do trabalho `archive-retrieval`. Use o comando `describe-job` para verificar o status do trabalho de recuperação anterior.

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

7. Quando o trabalho acima estiver concluído, use o comando `get-job-output` para baixar seu arquivo.

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

Excluir um arquivo no Amazon Glacier

Você não pode excluir um arquivo usando o console de gerenciamento do Amazon Glacier. Para excluir um arquivo, você deve usar a AWS Command Line Interface (CLI) ou escrever código para fazer uma solicitação de exclusão usando diretamente a API REST ou as AWS SDK para Java bibliotecas wrapper e.NET. Os tópicos a seguir explicam como usar AWS SDK para Java as bibliotecas wrapper e.NET, a API REST e a. AWS CLI

Tópicos

- Excluir um arquivo no Amazon Glacier usando o AWS SDK para Java
- Excluindo um arquivo no Amazon Glacier usando o AWS SDK para .NET
- Excluir um arquivo do Amazon Glacier usando a API REST
- Excluindo um arquivo no Amazon Glacier usando o AWS Command Line Interface

Você pode excluir um arquivo por vez de um cofre. Para excluir o arquivo, você deve fornecer o ID de arquivo na solicitação de exclusão. Você pode obter o ID de arquivo fazendo download do inventário do cofre que contém o arquivo. Para obter mais informações sobre como fazer download do inventário de cofre, consulte Fazer download de um inventário de cofre no Amazon Glacier.

Depois de excluir um arquivo, você ainda poderá fazer uma solicitação bem-sucedida para iniciar um trabalho a fim de recuperar o arquivo excluído, mas o trabalho de recuperação de arquivo falhará.

Recuperações de arquivo que estejam em andamento para um ID de arquivo quando você exclui o arquivo podem ser bem-sucedidas ou não, de acordo com os seguintes cenários:

- Se o trabalho de recuperação de arquivo estiver preparando ativamente os dados para download quando o Amazon Glacier receber a solicitação de exclusão de arquivo, a operação de recuperação de arquivo poderá falhar.
- Se o trabalho de recuperação de arquivo tiver preparado com êxito o arquivo para download quando o Amazon Glacier receber a solicitação de exclusão de arquivo, você poderá fazer o download da saída.

Para obter mais informações sobre recuperação de arquivo, consulte Baixando um arquivo no Amazon Glacier.

Essa operação é idempotente. Excluir um arquivo já excluído não resulta em um erro.

Depois de excluir um arquivo, se você fizer o download do inventário de cofre imediatamente, ele poderá incluir o arquivo excluído na lista porque o Amazon Glacier prepara o inventário de cofre somente uma vez por dia.

> **Note**
>
> Para exclusão automática de arquivos do cofre, consulte Exclusão automática dos arquivos do cofre no Amazon S3 Glacier.

Excluir um arquivo no Amazon Glacier usando o AWS SDK para Java

Estas são as etapas para excluir um arquivo usando a API de nível inferior do AWS SDK para Java.

1. Crie uma instância da classe `AmazonGlacierClient` (o cliente).

   Você precisa especificar uma região da AWS onde o arquivo que deseja excluir está armazenado. Todas as operações realizadas por você usando esse cliente se aplicam a essa regiãoAWS.
2. Forneça informações sobre a solicitação criando uma instância da classe `DeleteArchiveRequest`.

   Você precisa fornecer um ID de arquivo, um nome de cofre e o ID da conta. Se você não fornecer um ID da conta, o ID da conta associado às credenciais fornecidas por você para assinar a solicitação será pressuposto. Para obter mais informações, consulte Usando o AWS SDK para Java com o Amazon Glacier.
3. Execute o método `deleteArchive` fornecendo o objeto de solicitação como um parâmetro.

O trecho de código Java a seguir ilustra as etapas anteriores.

```
AmazonGlacierClient client;

DeleteArchiveRequest request = new DeleteArchiveRequest()
    .withVaultName("*** provide a vault name ***")
    .withArchiveId("*** provide an archive ID ***");

client.deleteArchive(request);
```

> **Note**
>
> Para obter informações sobre a API REST subjacente, consulte Delete Archive (DELETE archive).

Exemplo: excluir um arquivamento usando a AWS SDK para Java

O exemplo de código do Java a seguir usa o AWS SDK para Java para excluir um arquivo. Para instruções detalhadas sobre como executar esse exemplo, consulte Executar exemplos do Java para o Amazon Glacier usando o Eclipse. Você precisa atualizar o código, conforme mostrado, com um nome de cofre e o ID do arquivo que deseja excluir.

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);
        }
    }
}
```

Excluindo um arquivo no Amazon Glacier usando o AWS SDK para .NET

Tanto o alto quanto o baixo nível APIs fornecidos pelo Amazon SDK para .NET fornecem um método para excluir um arquivo.

Tópicos

- Excluindo um arquivo usando a API de alto nível do AWS SDK para .NET
- Excluindo um arquivo usando a API de baixo nível AWS SDK para .NET

Excluindo um arquivo usando a API de alto nível do AWS SDK para .NET

A classe `ArchiveTransferManager` da API de nível superior fornece o método `DeleteArchive` que você pode usar para excluir um arquivo.

Exemplo: excluir um arquivo usando a API de alto nível do AWS SDK para .NET

O exemplo de código C# a seguir usa a API de alto nível do AWS SDK para .NET para excluir um arquivo. Para step-by-step obter instruções sobre como executar esse exemplo, consulteExecutar exemplos de código. Você precisa atualizar o código conforme mostrado com o ID do arquivo que deseja excluir.

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();
    }
  }
}
```

Excluindo um arquivo usando a API de baixo nível AWS SDK para .NET

Veja a seguir as etapas para excluir um arquivo usando o AWS SDK para .NET.

1. Crie uma instância da classe `AmazonGlacierClient` (o cliente).

   Você precisa especificar uma AWS região na qual o arquivo que você deseja excluir está armazenado. Todas as operações que você executa usando esse cliente se aplicam a essa AWS região.

2. Forneça informações sobre a solicitação criando uma instância da classe `DeleteArchiveRequest`.

   Você precisa fornecer um ID de arquivo, um nome de cofre e o ID da conta. Se você não fornecer um ID da conta, o ID da conta associado às credenciais fornecidas por você para assinar a solicitação será pressuposto. Para obter mais informações, consulte Usando o AWS SDKs com o Amazon Glacier.

3. Execute o método `DeleteArchive` fornecendo o objeto de solicitação como um parâmetro.

Exemplo: excluir um arquivo usando a API de baixo nível do AWS SDK para .NET

O exemplo do C# a seguir ilustra as etapas anteriores. O exemplo usa a API de baixo nível do AWS SDK para .NET para excluir um arquivo.

> **Note**
>
> Para obter informações sobre a API REST subjacente, consulte Delete Archive (DELETE archive).

Para step-by-step obter instruções sobre como executar esse exemplo, consulteExecutar exemplos de código. Você precisa atualizar o código conforme mostrado com o ID do arquivo que deseja excluir.

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);
    }
  }
}
```

Excluir um arquivo do Amazon Glacier usando a API REST

Você pode usar a API Delete Archive para excluir um arquivo.

- Para obter mais informações sobre a API Delete Archive, consulte Delete Archive (DELETE archive).
- Para obter informações sobre como usar a API REST, consulte Referência de API para Amazon Glacier.

Excluindo um arquivo no Amazon Glacier usando o AWS Command Line Interface

Você pode excluir arquivos no Amazon Glacier (Amazon Glacier) usando AWS Command Line Interface o ().AWS CLI

Tópicos

- (Pré-requisito) Configurando o AWS CLI
- Exemplo: Excluindo um arquivo usando o AWS CLI

(Pré-requisito) Configurando o AWS CLI

1. Faça download e configure a AWS CLI. Para obter instruções, consulte os seguintes tópicos no Guia do usuário do AWS Command Line Interface :

   Instalando o AWS Command Line Interface

   Configurando o AWS Command Line Interface

2. Verifique sua AWS CLI configuração inserindo os seguintes comandos no prompt de comando. Esses comandos não fornecem as credenciais explicitamente, de modo que as credenciais do perfil padrão são usadas.

   - Tente usar o comando de ajuda.

     ```
     aws help
     ```

   - Para obter uma lista dos cofres do Amazon Glacier na conta configurada, use o comando `list-vaults`. *123456789012*Substitua pelo seu Conta da AWS ID.

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

   - Para ver os dados de configuração atuais do AWS CLI, use o `aws configure list` comando.

     ```
     aws configure list
     ```

Exemplo: Excluindo um arquivo usando o AWS CLI

1. Use o comando initiate-job para iniciar um trabalho de recuperação de inventário.

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

   Saída esperada:

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

```
}
```

2. Use o comando describe-job para verificar o status do trabalho de recuperação anterior.

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

Saída esperada:

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

3. Aguarde a conclusão do trabalho .

Você deve aguardar até a saída do trabalho estar pronta para download. Se você definiu uma configuração de notificação no cofre ou especificou um tópico do Amazon Simple Notification Service (Amazon SNS) ao iniciar o trabalho, o Amazon Glacier envia uma mensagem para o tópico depois de concluir o trabalho.

É possível definir a configuração de notificação para eventos específicos no cofre. Para obter mais informações, consulte Configurar notificações do cofre no Amazon Glacier. O Amazon Glacier enviará uma mensagem para o tópico do SNS especificado sempre que o evento determinado ocorrer.

4. Quando estiver concluído, use o comando get-job-output para fazer download do trabalho de recuperação para o arquivo `output.json`.

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

Esse comando produz um arquivo com os campos a seguir.

```
{
"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. Use o comando `delete-archive` para excluir cada arquivamento de um cofre até que não haja mais nenhum.

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

Usando o AWS SDKs com o Amazon Glacier

AWS permite SDKs que você desenvolva aplicativos para o Amazon Glacier. As bibliotecas de SDKs encapsulam a API do Amazon Glacier subjacente, simplificando as tarefas de programação. Por exemplo, para cada solicitação enviada ao Amazon Glacier, deve ser incluída uma assinatura a fim de autenticar as solicitações. Ao usar as bibliotecas do SDK, você precisa fornecer somente suas credenciais de AWS segurança em seu código, e as bibliotecas computam a assinatura necessária e a incluem na solicitação enviada ao Amazon Glacier. Eles AWS SDKs fornecem bibliotecas que mapeiam para a API REST subjacente e fornecem objetos que você pode usar para criar solicitações e processar respostas com facilidade.

Tópicos

- AWS Bibliotecas do SDK para Java e.NET
- Usando o Amazon Glacier com um SDK AWS
- Usando o AWS SDK para Java com o Amazon Glacier
- Como usar o AWS SDK para .NET com o Amazon Glacier

O AWS Command Line Interface (AWS CLI) é uma ferramenta unificada para gerenciar sua Serviços da AWS, incluindo o Amazon Glacier. Para obter informações sobre como baixar o AWS CLI, consulte AWS Command Line Interface. Para obter uma lista de comandos da CLI do Amazon Glacier, consulte Referência de comandos da AWS CLI.

AWS Bibliotecas do SDK para Java e.NET

O AWS SDKs for Java e.NET oferece bibliotecas de wrapper de alto e baixo nível.

Você pode encontrar exemplos de trabalho com o Amazon Glacier usando o AWS SDK para Java e o AWS SDK para .NET em todo este guia do desenvolvedor.

O que é API de nível inferior?

As bibliotecas wrapper de baixo nível são mapeadas de maneira próxima à API REST subjacente (Referência de API para Amazon Glacier) compatível com o Amazon Glacier. Para cada operação REST do Amazon Glacier, a API de nível inferior oferece um método correspondente, um objeto de solicitação para que sejam fornecidas informações solicitadas e um objeto de resposta para

que você processe a resposta do Amazon Glacier. As bibliotecas wrapper de nível inferior são a implementação mais completa das operações subjacentes do Amazon Glacier.

Para obter informações sobre essas bibliotecas do SDK, consulte Usando o AWS SDK para Java com o Amazon Glacier e Como usar o AWS SDK para .NET com o Amazon Glacier.

O que é API de nível superior?

Para simplificar ainda mais o desenvolvimento do aplicativo, essas bibliotecas oferecem uma abstração de nível superior para algumas das operações. Por exemplo:

- Fazer upload de um arquivo: para fazer upload de um arquivo usando API de nível inferior, além do nome de arquivo e o nome do cofre onde deseja salvar o arquivo, você precisa fornecer uma soma de verificação (SHA-256 tree hash) da carga útil. No entanto, a API de nível superior computa a soma de verificação para você.
- Fazer download de um arquivo ou inventário do cofre: para fazer download de um arquivo usando API de nível inferior, você primeiro inicia um trabalho, aguarda a conclusão e obtém a saída do trabalho. Escreva um código adicional para configurar um tópico do Amazon Simple Notification Service (Amazon SNS) para receber via Amazon Glacier uma notificação quando o trabalho for concluído. Você também precisa um mecanismo de sondagem para verificar se uma mensagem de conclusão do trabalho foi publicada no tópico. A API de nível superior oferece um método para fazer download de um arquivo que cuida de todas essas etapas. Você especifica somente um ID de arquivo e um caminho de pasta onde deseja salvar os dados obtidos por download.

Para obter informações sobre essas bibliotecas do SDK, consulte Usando o AWS SDK para Java com o Amazon Glacier e Como usar o AWS SDK para .NET com o Amazon Glacier.

Quando usar as APIs de nível superior e de nível inferior

Em geral, se a API de nível superior oferece métodos dos quais precisa para realizar uma operação, você deverá usá-la em função da simplicidade proporcionada. No entanto, se a API de nível superior não oferece a funcionalidade, você poderá usar a API de nível inferior. Além disso, a API de nível inferior permite o controle granular da operação, como a lógica de nova tentativa em caso de falha. Por exemplo, ao fazer upload de um arquivo, a API de nível superior usa o tamanho do arquivo para determinar se é necessário fazer upload do arquivo em uma única operação ou usar a API de multipart upload. A API também tem uma lógica de nova tentativa integrada em caso de falha no upload. No entanto, o aplicativo pode precisar de controle granular sobre essas decisões, quando você pode usar a API de nível inferior.

Usando o Amazon Glacier com um SDK AWS

AWS kits de desenvolvimento de software (SDKs) estão disponíveis para muitas linguagens de programação populares. Cada SDK fornece uma API, exemplos de código e documentação que permitem que os desenvolvedores criem facilmente aplicações em seu idioma de preferência.

Documentação do SDK	Exemplos de código
AWS SDK para C++	AWS SDK para C++ exemplos de código
AWS CLI	AWS CLI exemplos de código
AWS SDK para Go	AWS SDK para Go exemplos de código
AWS SDK para Java	AWS SDK para Java exemplos de código
AWS SDK para JavaScript	AWS SDK para JavaScript exemplos de código
AWS SDK para Kotlin	AWS SDK para Kotlin exemplos de código
AWS SDK para .NET	AWS SDK para .NET exemplos de código
AWS SDK para PHP	AWS SDK para PHP exemplos de código
Ferramentas da AWS para PowerShell	Ferramentas da AWS para PowerShell exemplos de código
AWS SDK para Python (Boto3)	AWS SDK para Python (Boto3) exemplos de código
AWS SDK para Ruby	AWS SDK para Ruby exemplos de código
AWS SDK para Rust	AWS SDK para Rust exemplos de código
AWS SDK para SAP ABAP	AWS SDK para SAP ABAP exemplos de código
AWS SDK para Swift	AWS SDK para Swift exemplos de código

Para obter exemplos específicos do Amazon Glacier, consulte Exemplos de código para o Amazon Glacier usando AWS SDKs.

> **Exemplo de disponibilidade**
>
> Não consegue encontrar o que precisa? Solicite um exemplo de código usando o link Fornecer feedback na parte inferior desta página.

Usando o AWS SDK para Java com o Amazon Glacier

O AWS SDK para Java fornece alto e baixo nível APIs para o Amazon Glacier (Amazon Glacier), conforme descrito em. Usando o AWS SDKs com o Amazon Glacier Para obter informações sobre o download do AWS SDK para Java, consulte Amazon SDK for Java.

> **Note**
>
> AWS SDK para Java Ele fornece clientes seguros para acessar o Amazon Glacier. De acordo com as melhores práticas, seus aplicativos devem criar um único cliente e reutilizá-lo entre os threads.

Tópicos

- Usar a API de nível inferior
- Usar a API de nível superior
- Executar exemplos do Java para o Amazon Glacier usando o Eclipse
- Definição do endpoint

Usar a API de nível inferior

A classe `AmazonGlacierClient` de nível inferior fornece todos os métodos mapeados para as operações REST subjacentes do Amazon Glacier (Referência de API para Amazon Glacier). Ao chamar qualquer um desses métodos, deve ser criado um objeto de solicitação correspondente e fornecer um objeto de resposta no qual o método possa retornar a resposta do Amazon Glacier à operação.

Por exemplo, a classe `AmazonGlacierClient` fornece o método `createVault` para criar um cofre. Esse método é mapeado para a operação REST Create Vault (consulte Create Vault (PUT vault)). Para usar esse método, crie instâncias do objeto `CreateVaultResult` que recebe a resposta do Amazon Glacier conforme mostrado no seguinte trecho de código do Java:

```
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);
```

Todos os exemplos de nível inferior no guia usam esse padrão.

> **Note**
>
> O segmento do código anterior especifica `AccountID` durante a criação da solicitação. No entanto, ao usar o AWS SDK para Java, o `AccountId` na solicitação é opcional e, portanto, todos os exemplos de baixo nível neste guia não definem esse valor. Esse `AccountId` é o Conta da AWS ID. Esse valor deve corresponder ao Conta da AWS ID associado às credenciais usadas para assinar a solicitação. Você pode especificar o Conta da AWS ID ou, opcionalmente, um '-'. Nesse caso, o Amazon Glacier usa Conta da AWS o ID associado às credenciais usadas para assinar a solicitação. Se você especificar o ID da conta, não inclua hifens nele. Ao usar AWS SDK para Java, se você não fornecer o ID da conta, a biblioteca definirá o ID da conta como '-'.

Usar a API de nível superior

Para simplificar ainda mais o desenvolvimento de seu aplicativo, o AWS SDK para Java fornece a `ArchiveTransferManager` classe que implementa uma abstração de alto nível para alguns dos métodos na API de baixo nível. Ele fornece métodos úteis, como os métodos `upload` e `download`, para operações de arquivo.

Por exemplo, o trecho de código do Java a seguir usa o método de nível superior `upload` para fazer upload de um arquivo.

```
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();
```

Observe que todas as operações realizadas se aplicam à AWS região especificada ao criar o `ArchiveTransferManager` objeto. Se você não especificar nenhuma AWS região, ela será AWS SDK para Java definida `us-east-1` como a AWS região padrão.

Todos os exemplos de nível superior neste guia usam esse padrão.

> **Note**
>
> A classe `ArchiveTransferManager` de nível superior pode ser construída com uma instância `AmazonGlacierClient` ou `AWSCredentials`.

Executar exemplos do Java para o Amazon Glacier usando o Eclipse

A maneira mais fácil de começar a usar os exemplos de código Java é instalar o AWS Toolkit for Eclipse mais recente. Para obter informações sobre como instalar ou atualizar para o toolkit mais recente, vá até http://aws.amazon.com/eclipse. As tarefas a seguir orientam você em meio à criação e aos testes dos exemplos de código do Java fornecidos nesta seção.

Processo geral de criação de exemplos de código Java

1	Crie um perfil de credenciais padrão para suas AWS credenciais, conforme descrito no AWS SDK para Java tópico Fornecendo AWS credenciais no Amazon SDK for Java.
2	Crie um novo projeto AWS Java no Eclipse. O projeto é pré-configurado com o AWS SDK para Java.
3	Copie o código da seção que você está lendo ao projeto.
4	Atualize o código fornecendo todos os dados necessários. Por exemplo, se estiver fazendo o upload de um arquivo, forneça o caminho do arquivo e o nome do bucket.

5	Execute o código. Verifique se o objeto é criado usando o Console de gerenciamento da AWS. Para obter mais informações sobre o Console de gerenciamento da AWS, acesse http://aws.amazon.com/console/.

Definição do endpoint

Por padrão, o AWS SDK para Java usa o endpoint`https://glacier.us-east-1.amazonaws.com`. Você pode definir o endpoint explicitamente conforme mostrado nos trechos de código do Java a seguir.

O trecho de código a seguir mostra como definir o endpoint para a região Oeste dos EUA (Oregon) (`us-west-2`) na API de nível inferior.

Example

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

O trecho de código a seguir mostra como definir o endpoint para a região oeste dos EUA (Oregon) na API de nível superior.

```
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);
```

Para obter uma lista de AWS regiões e endpoints compatíveis, consulteAcessar o Amazon Glacier.

Como usar o AWS SDK para .NET com o Amazon Glacier

A API do AWS SDK para .NET está disponível em `AWSSDK.dll`. Para obter informações sobre como fazer download do AWS SDK para .NET, acesse Bibliotecas de códigos de exemplo. Conforme

descrito em Usando o AWS SDKs com o Amazon Glacier, o AWS SDK para .NET fornece APIs de alto nível e de baixo nível.

> **Note**
>
> As APIs de nível inferior e superior fornecem clientes thread-safe para acessar o Amazon Glacier. De acordo com as melhores práticas, seus aplicativos devem criar um único cliente e reutilizá-lo entre os threads.

Tópicos

- Usar a API de nível inferior
- Usar a API de nível superior
- Executar exemplos de código
- Definição do endpoint

Usar a API de nível inferior

A classe `AmazonGlacierClient` de nível inferior fornece todos os métodos mapeados para as operações REST subjacentes do Amazon Glacier (Referência de API para Amazon Glacier). Ao chamar qualquer um desses métodos, deve ser criado um objeto de solicitação correspondente e fornecer um objeto de resposta no qual o método possa retornar uma resposta do Amazon Glacier à operação.

Por exemplo, a classe `AmazonGlacierClient` fornece o método `CreateVault` para criar um cofre. Esse método é mapeado para a operação REST Create Vault (consulte Create Vault (PUT vault)). Para usar esse método, crie instâncias das classes `CreateVaultRequest` e `CreateVaultResponse` para fornecer informações de solicitação e receber uma resposta do Amazon Glacier, conforme mostrado no seguinte trecho de código C#:

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

CreateVaultRequest request = new CreateVaultRequest()
{
  AccountId = "-",
```

```
  VaultName = "*** Provide vault name ***"
};

CreateVaultResponse response = client.CreateVault(request);
```

Todos os exemplos de nível inferior no guia usam esse padrão.

> **Note**
>
> O segmento do código anterior especifica `AccountId` durante a criação da solicitação. No entanto, durante o uso do AWS SDK para .NET, o `AccountId` na solicitação é opcional e, assim, todos os exemplos de baixo nível neste guia não definem esse valor. Esse `AccountId` é o ID da Conta da AWS. Esse valor deve corresponder ao ID de Conta da AWS associado às credenciais usadas para assinar a solicitação. Especifique o ID da Conta da AWS ou opcionalmente um '-', nesse caso o Amazon Glacier usa o ID da Conta da AWS associado às credenciais usadas para assinar a solicitação. Se você especificar o ID da conta, não inclua hifens nele. Ao usar o AWS SDK para .NET, se você não fornecer o ID da conta, a biblioteca definirá o ID da conta como '-'.

Usar a API de nível superior

Para simplificar ainda mais o desenvolvimento do aplicativo, o AWS SDK para .NET fornece a classe `ArchiveTransferManager` que implementa uma abstração de alto nível para alguns dos métodos na API de baixo nível. Ele fornece métodos úteis, como `Upload` e `Download`, para as operações de arquivo.

Por exemplo, o trecho de código do C# a seguir usa o método de nível superior `Upload` para fazer upload de um arquivo.

```
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;
```

Todas as operações realizadas por você se aplicam à AWS região especificada ao criar o `ArchiveTransferManager` objeto . Todos os exemplos de nível superior neste guia usam esse padrão.

> **Note**
>
> A classe `ArchiveTransferManager` de nível superior ainda precisa do cliente `AmazonGlacierClient` de nível inferior, que você pode passar explicitamente, ou o `ArchiveTransferManager` criará o cliente.

Executar exemplos de código

A maneira mais fácil de começar a usar os exemplos de código .NET é instalar o AWS SDK para .NET. Para mais informações, acesse Amazon SDK para .NET.

O procedimento a seguir descreve as etapas para você testar os exemplos de código fornecidos neste guia.

Processo geral de criação dos exemplos de código do .NET (usando o Visual Studio)

1	Crie um perfil de credenciais para suas credenciais da AWS conforme descrito no tópico Amazon SDK para .NET Como configurar credenciais da AWS.
2	Crie um novo projeto do Visual Studio usando o modelo AWS Projeto Vazio.
3	Substitua o código no arquivo do projeto, `Program.cs` , com o código da seção que você está lendo.
4	Execute o código. Verifique se o objeto é criado usando o Console de gerenciamento da AWS. Para obter mais informações sobre o Console de gerenciamento da AWS, acesse https://aws.amazon.com/console/.

Definição do endpoint

Por padrão, o AWS SDK para .NET define o endpoint para a região Oeste dos EUA (Oregon) (`https://glacier.us-west-2.amazonaws.com`). Defina o endpoint para AWS outras regiões conforme mostrado nos trechos do C# a seguir.

O trecho de código a seguir mostra como definir o endpoint para a região Oeste dos EUA (Oregon) (`us-west-2`) na API de nível inferior.

Example

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

O trecho de código a seguir mostra como definir o endpoint para a região Oeste dos EUA (Oregon) na API de nível superior.

```
var manager = new ArchiveTransferManager(Amazon.RegionEndpoint.USWest2);
```

Para obter uma lista atual de AWS regiões e endpoints compatíveis, consulte Acessar o Amazon Glacier.

Exemplos de código para o Amazon Glacier usando AWS SDKs

Os exemplos de código a seguir mostram como usar o Amazon Glacier com um kit de desenvolvimento AWS de software (SDK).

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar perfis de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cenários são exemplos de código que mostram como realizar tarefas específicas chamando várias funções dentro de um serviço ou combinadas com outros Serviços da AWS.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Glacier com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Exemplos de código

- Exemplos básicos do Amazon Glacier usando AWS SDKs
  - Olá, Amazon Glacier
  - Ações para o Amazon Glacier usando AWS SDKs
    - Use AddTagsToVault com um AWS SDK ou CLI
    - Use CreateVault com um AWS SDK ou CLI
    - Use DeleteArchive com um AWS SDK ou CLI
    - Use DeleteVault com um AWS SDK ou CLI
    - Use DeleteVaultNotifications com um AWS SDK ou CLI
    - Use DescribeJob com um AWS SDK ou CLI
    - Use DescribeVault com um AWS SDK ou CLI
    - Use GetJobOutput com um AWS SDK ou CLI
    - Use GetVaultNotifications com um AWS SDK ou CLI
    - Use InitiateJob com um AWS SDK ou CLI
    - Use ListJobs com um AWS SDK ou CLI
    - Use ListTagsForVault com um AWS SDK ou CLI
    - Use ListVaults com um AWS SDK ou CLI

    - Use SetVaultNotifications com um AWS SDK ou CLI
    - Use UploadArchive com um AWS SDK ou CLI
    - Use UploadMultipartPart com um AWS SDK ou CLI
- Cenários para o uso do Amazon Glacier AWS SDKs
  - Arquive um arquivo no Amazon Glacier, receba notificações e inicie um trabalho usando um SDK AWS
  - Obtenha o conteúdo de arquivamento do Amazon Glacier e exclua o arquivo usando um SDK AWS

Exemplos básicos do Amazon Glacier usando AWS SDKs

Os exemplos de código a seguir mostram como usar os conceitos básicos do Amazon Glacier AWS SDKs com.

Exemplos

- Olá, Amazon Glacier
- Ações para o Amazon Glacier usando AWS SDKs
  - Use AddTagsToVault com um AWS SDK ou CLI
  - Use CreateVault com um AWS SDK ou CLI
  - Use DeleteArchive com um AWS SDK ou CLI
  - Use DeleteVault com um AWS SDK ou CLI
  - Use DeleteVaultNotifications com um AWS SDK ou CLI
  - Use DescribeJob com um AWS SDK ou CLI
  - Use DescribeVault com um AWS SDK ou CLI
  - Use GetJobOutput com um AWS SDK ou CLI
  - Use GetVaultNotifications com um AWS SDK ou CLI
  - Use InitiateJob com um AWS SDK ou CLI
  - Use ListJobs com um AWS SDK ou CLI
  - Use ListTagsForVault com um AWS SDK ou CLI
  - Use ListVaults com um AWS SDK ou CLI
  - Use SetVaultNotifications com um AWS SDK ou CLI
  - Use UploadArchive com um AWS SDK ou CLI

- Use UploadMultipartPart com um AWS SDK ou CLI

Olá, Amazon Glacier

O exemplo de código a seguir mostra como começar a usar o Amazon Glacier.

.NET

SDK para .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
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}");
        }
```

```
    }
}
```

- Para obter detalhes da API, consulte ListVaultsa Referência AWS SDK para .NET da API.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Glacier com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Ações para o Amazon Glacier usando AWS SDKs

Os exemplos de código a seguir demonstram como realizar ações individuais do Amazon Glacier com. AWS SDKs Cada exemplo inclui um link para GitHub, onde você pode encontrar instruções para configurar e executar o código.

Esses trechos chamam a API do Amazon Glacier e são trechos de código de programas maiores que devem ser executados no contexto. É possível ver as ações em contexto em Cenários para o uso do Amazon Glacier AWS SDKs.

Os exemplos a seguir incluem apenas as ações mais utilizadas. Para obter uma lista completa, consulte a Referência da API do Amazon Glacier.

Exemplos

- Use AddTagsToVault com um AWS SDK ou CLI
- Use CreateVault com um AWS SDK ou CLI
- Use DeleteArchive com um AWS SDK ou CLI
- Use DeleteVault com um AWS SDK ou CLI
- Use DeleteVaultNotifications com um AWS SDK ou CLI
- Use DescribeJob com um AWS SDK ou CLI
- Use DescribeVault com um AWS SDK ou CLI
- Use GetJobOutput com um AWS SDK ou CLI
- Use GetVaultNotifications com um AWS SDK ou CLI
- Use InitiateJob com um AWS SDK ou CLI
- Use ListJobs com um AWS SDK ou CLI
- Use ListTagsForVault com um AWS SDK ou CLI

- Use ListVaults com um AWS SDK ou CLI
- Use SetVaultNotifications com um AWS SDK ou CLI
- Use UploadArchive com um AWS SDK ou CLI
- Use UploadMultipartPart com um AWS SDK ou CLI

Use **AddTagsToVault** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `AddTagsToVault`.

.NET

SDK para .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
   /// <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;
    }
```

- Para obter detalhes da API, consulte AddTagsToVaulta Referência AWS SDK para .NET da API.

CLI

AWS CLI

O seguinte comando adiciona duas tags ao cofre `my-vault`:

```
aws glacier add-tags-to-vault --account-id - --vault-name my-vault --
tags id=1234,date=july2015
```

O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

- Para obter detalhes da API, consulte AddTagsToVaultem Referência de AWS CLI Comandos.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Glacier com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`CreateVault`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `CreateVault`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Arquivar um arquivo, obter notificações e iniciar um trabalho

.NET

SDK para .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
/// <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;
}
```

- Para obter detalhes da API, consulte CreateVaulta Referência AWS SDK para .NET da API.

CLI

AWS CLI

O seguinte comando cria um cofre chamado `my-vault`:

```
aws glacier create-vault --vault-name my-vault --account-id -
```

O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

- Para obter detalhes da API, consulte CreateVaultem Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
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);
        }
    }
}
```

- Para obter detalhes da API, consulte CreateVaulta Referência AWS SDK for Java 2.x da API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no Repositório de exemplos de código da AWS.

Crie o cliente.

```
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 };
```

Crie o cofre.

```
// 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();
```

- Para obter mais informações, consulte o Guia do desenvolvedor do AWS SDK para JavaScript.
- Para obter detalhes da API, consulte CreateVaulta Referência AWS SDK para JavaScript da API.

SDK para JavaScript (v2)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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!");
  }
});
```

- Para obter mais informações, consulte o Guia do desenvolvedor do AWS SDK para JavaScript.
- Para obter detalhes da API, consulte CreateVaulta Referência AWS SDK para JavaScript da API.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: criar novo cofre para a conta do usuário. Como nenhum valor foi fornecido ao AccountId parâmetro -, os cmdlets usam o padrão “-” indicando a conta atual.

```
New-GLCVault -VaultName myvault
```

Saída:

```
/01234567812/vaults/myvault
```

- Para obter detalhes da API, consulte CreateVaultem Referência de Ferramentas da AWS para PowerShell cmdlet (V4).

Ferramentas para PowerShell V5

Exemplo 1: criar novo cofre para a conta do usuário. Como nenhum valor foi fornecido ao AccountId parâmetro -, os cmdlets usam o padrão "-" indicando a conta atual.

```
New-GLCVault -VaultName myvault
```

Saída:

```
/01234567812/vaults/myvault
```

- Para obter detalhes da API, consulte CreateVaultem Referência de Ferramentas da AWS para PowerShell cmdlet (V5).

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
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
```

- Para obter detalhes da API, consulte a CreateVaultReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Glacier com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`DeleteArchive`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `DeleteArchive`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Obter o conteúdo do arquivo e excluir o arquivo

CLI

AWS CLI

Como excluir um arquivo de um cofre

O exemplo de `delete-archive` a seguir remove o arquivamento especificado de `example_vault`.

```
aws glacier delete-archive \
    --account-id 111122223333  \
    --vault-name example_vault \
    --archive-id Sc0u9ZP8yaWkmh-XGlIvAVprtLhaLCGnNwNl5I5x9HqPIkX5mjc0DrId3Ln-
Gi_k2HzmlIDZUz117KSdVMdMXLuFWi9PJUitxWO73edQ43eTlMWkH0pd9zVSAuV_XXZBVhKhyGhJ7w
```

Este comando não produz saída.

- Para obter detalhes da API, consulte DeleteArchiveem Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
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);
        }
    }
}
```

- Para obter detalhes da API, consulte DeleteArchivea Referência AWS SDK for Java 2.x da API.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
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
```

- Para obter detalhes da API, consulte a DeleteArchiveReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Glacier com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`DeleteVault`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `DeleteVault`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Obter o conteúdo do arquivo e excluir o arquivo

CLI

AWS CLI

O seguinte comando cria um cofre chamado `my-vault`:

```
aws glacier delete-vault --vault-name my-vault --account-id -
```

Esse comando não produz nenhuma saída. O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

- Para obter detalhes da API, consulte DeleteVaultem Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
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);
        }
    }
}
```

- Para obter detalhes da API, consulte DeleteVaulta Referência AWS SDK for Java 2.x da API.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
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
```

- Para obter detalhes da API, consulte a DeleteVaultReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Glacier com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`DeleteVaultNotifications`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `DeleteVaultNotifications`.

CLI

AWS CLI

Para remover as notificações do SNS de um cofre

O comando de `delete-vault-notifications` a seguir remove as notificações enviadas pelo Amazon Simple Notification Service (Amazon SNS) do cofre especificado.

```
aws glacier delete-vault-notifications \
    --account-id 111122223333 \
    --vault-name example_vault
```

Este comando não produz saída.

- Para obter detalhes da API, consulte DeleteVaultNotificationsem Referência de AWS CLI Comandos.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
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
```

- Para obter detalhes da API, consulte a DeleteVaultNotificationsReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Glacier com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`DescribeJob`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `DescribeJob`.

CLI

AWS CLI

O comando a seguir recupera informações sobre um trabalho de recuperação de inventário em um cofre chamado `my-vault`:

```
aws glacier describe-job --account-id - --vault-name my-
vault --job-id zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
RO47Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61dsO4CnMW
```

Saída:

```
{
    "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"
}
```

O ID do trabalho pode ser encontrado na saída de `aws glacier initiate-job` e `aws glacier list-jobs`. O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

- Para obter detalhes da API, consulte DescribeJobem Referência de AWS CLI Comandos.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: exibir detalhes do trabalho especificado. Quando o trabalho é concluído com êxito, o cmdlet GCJob Read-Output pode ser usado para recuperar o conteúdo do trabalho (um arquivo ou uma lista de inventário) para o sistema de arquivos local.

```
Get-GLCJob -VaultName myvault -JobId "op1x...JSbthM"
```

Saída:

```
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
```

- Para obter detalhes da API, consulte DescribeJobem Referência de Ferramentas da AWS para PowerShell cmdlet (V4).

Ferramentas para PowerShell V5

Exemplo 1: exibir detalhes do trabalho especificado. Quando o trabalho é concluído com êxito, o cmdlet GCJob Read-Output pode ser usado para recuperar o conteúdo do trabalho (um arquivo ou uma lista de inventário) para o sistema de arquivos local.

```
Get-GLCJob -VaultName myvault -JobId "op1x...JSbthM"
```

Saída:

```
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
```

- Para obter detalhes da API, consulte DescribeJobem Referência de Ferramentas da AWS para PowerShell cmdlet (V5).

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
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
```

- Para obter detalhes da API, consulte a DescribeJobReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Glacier com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`DescribeVault`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `DescribeVault`.

.NET

SDK para .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
/// <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;
    }
```

- Para obter detalhes da API, consulte DescribeVaulta Referência AWS SDK para .NET da API.

CLI

AWS CLI

O seguinte comando recupera dados sobre o cofre `my-vault`:

```
aws glacier describe-vault --vault-name my-vault --account-id -
```

O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

- Para obter detalhes da API, consulte DescribeVaultem Referência de AWS CLI Comandos.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Glacier com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`GetJobOutput`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `GetJobOutput`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Obter o conteúdo do arquivo e excluir o arquivo

CLI

AWS CLI

O seguinte comando salva a saída de um trabalho de inventário do cofre em um arquivo no diretório atual chamado `output.json`:

```
aws glacier get-job-output --account-id - --vault-name my-
vault --job-id zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
RO47Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61dsO4CnMW output.json
```

O `job-id` está disponível na saída de `aws glacier list-jobs`. Observe que o nome do arquivo de saída é um argumento posicional que não é prefixado por um nome de opção. O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

Saída:

```
{
    "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
```

- Para obter detalhes da API, consulte GetJobOutputem Referência de AWS CLI Comandos.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: baixar o conteúdo do arquivo que foi agendado para recuperação na tarefa especificada e armazená-lo em um arquivo no disco. O download valida a soma de verificação para você, se houver uma disponível. Se desejado, toda a resposta, incluindo a soma de verificação, pode ser exibida especificando **`-Select '*'`**.

```
Read-GLCJobOutput -VaultName myvault -JobId "HSWjArc...Zq2XLiW" -FilePath "c:
\temp\blue.bin"
```

- Para obter detalhes da API, consulte GetJobOutputem Referência de Ferramentas da AWS para PowerShell cmdlet (V4).

Ferramentas para PowerShell V5

Exemplo 1: baixar o conteúdo do arquivo que foi agendado para recuperação na tarefa especificada e armazená-lo em um arquivo no disco. O download valida a soma de verificação para você, se houver uma disponível. Se desejado, toda a resposta, incluindo a soma de verificação, pode ser exibida especificando **`-Select '*'`**.

```
Read-GLCJobOutput -VaultName myvault -JobId "HSWjArc...Zq2XLiW" -FilePath "c:
\temp\blue.bin"
```

- Para obter detalhes da API, consulte GetJobOutputem Referência de Ferramentas da AWS para PowerShell cmdlet (V5).

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
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
```

- Para obter detalhes da API, consulte a GetJobOutputReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Glacier com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **GetVaultNotifications** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `GetVaultNotifications`.

CLI

AWS CLI

O comando a seguir recebe uma descrição da configuração de notificação de um cofre chamado `my-vault`:

```
aws glacier get-vault-notifications --account-id - --vault-name my-vault
```

Saída:

```
{
    "vaultNotificationConfig": {
        "Events": [
            "InventoryRetrievalCompleted",
            "ArchiveRetrievalCompleted"
        ],
        "SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-vault"
    }
}
```

Se nenhuma notificação tiver sido configurada para o cofre, será retornado um erro. O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

- Para obter detalhes da API, consulte GetVaultNotificationsem Referência de AWS CLI Comandos.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
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
```

- Para obter detalhes da API, consulte a GetVaultNotificationsReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Glacier com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`InitiateJob`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `InitiateJob`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Arquivar um arquivo, obter notificações e iniciar um trabalho

.NET

SDK para .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

Recupere um arquivo de um cofre. Este exemplo usa a ArchiveTransferManager classe. Para obter detalhes da API, consulte 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}%");
        }
    }
```

- Para obter detalhes da API, consulte InitiateJoba Referência AWS SDK para .NET da API.

CLI

AWS CLI

O comando a seguir inicia um trabalho para obter um inventário de cofre `my-vault`:

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

Saída:

```
{
    "location": "/0123456789012/vaults/my-vault/jobs/
zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
RO47Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61dsO4CnMW",
    "jobId": "zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
RO47Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61dsO4CnMW"
}
```

O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

O comando a seguir inicia um trabalho para recuperar um arquivo do cofre `my-vault`:

```
aws glacier initiate-job --account-id - --vault-name my-vault --job-
parameters file://job-archive-retrieval.json
```

`job-archive-retrieval.json` é um arquivo JSON na pasta local que especifica o tipo de trabalho, a ID do arquivo e alguns parâmetros opcionais:

```
{
  "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"
}
```

O arquivo IDs está disponível na saída de `aws glacier upload-archive` `aws glacier get-job-output` e.

Saída:

```
{
```

```
    "location": "/011685312445/vaults/mwunderl/jobs/l7IL5-
EkXyEY9Ws95fClzIbk2O5uLYaFdAYOi-
azsX_Z8V6NH4yERHzars8wTKYQMX6nBDI9cMNHzyZJO59-8N9aHWav",
    "jobId": "l7IL5-EkXy2O5uLYaFdAYOiEY9Ws95fClzIbk-
azsX_Z8V6NH4yERHzars8wTKYQMX6nBDI9cMNHzyZJO59-8N9aHWav"
}
```

Consulte Iniciar Tarefa na Referência de API do Amazon Glacier para ver detalhes sobre o formato dos parâmetros da tarefa.

- Para obter detalhes da API, consulte InitiateJobem Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

Recupere um inventário do cofre.

```
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);

        }
    }
}
```

- Para obter detalhes da API, consulte InitiateJoba Referência AWS SDK for Java 2.x da API.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: iniciar um trabalho para recuperar um arquivo do cofre especificado de propriedade do usuário. O status do trabalho pode ser verificado usando o GLCJob cmdlet Get-. Quando o trabalho é concluído com êxito, o cmdlet GCJob Read-Output pode ser usado para recuperar o conteúdo do arquivamento no sistema de arquivos local.

```
Start-GLCJob -VaultName myvault -JobType "archive-retrieval" -JobDescription
 "archive retrieval" -ArchiveId "o9O9j...TX-TpIhQJw"
```

Saída:

```
JobId            JobOutputPath Location
-----            ------------- --------
op1x...JSbthM                  /012345678912/vaults/test/jobs/
op1xe...I4HqCHkSJSbthM
```

- Para obter detalhes da API, consulte InitiateJobem Referência de Ferramentas da AWS para PowerShell cmdlet (V4).

Ferramentas para PowerShell V5

Exemplo 1: iniciar um trabalho para recuperar um arquivo do cofre especificado de propriedade do usuário. O status do trabalho pode ser verificado usando o GLCJob cmdlet Get-. Quando o trabalho é concluído com êxito, o cmdlet GCJob Read-Output pode ser usado para recuperar o conteúdo do arquivamento no sistema de arquivos local.

```
Start-GLCJob -VaultName myvault -JobType "archive-retrieval" -JobDescription
 "archive retrieval" -ArchiveId "o9O9j...TX-TpIhQJw"
```

Saída:

```
JobId            JobOutputPath Location
-----            ------------- --------
op1x...JSbthM                  /012345678912/vaults/test/jobs/
op1xe...I4HqCHkSJSbthM
```

- Para obter detalhes da API, consulte InitiateJobem Referência de Ferramentas da AWS para PowerShell cmdlet (V5).

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

Recupere um inventário do cofre.

```
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
```

Recupere um arquivo de um cofre.

```
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
```

- Para obter detalhes da API, consulte a InitiateJobReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Glacier com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`ListJobs`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `ListJobs`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto nos seguintes exemplos de código:

- Arquivar um arquivo, obter notificações e iniciar um trabalho
- Obter o conteúdo do arquivo e excluir o arquivo

.NET

SDK para .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
   /// <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;
    }
```

- Para obter detalhes da API, consulte ListJobsa Referência AWS SDK para .NET da API.

CLI

AWS CLI

O seguinte comando lista trabalhos em andamento e concluídos recentemente no cofre `my-vault`:

```
aws glacier list-jobs --account-id - --vault-name my-vault
```

Saída:

```
{
    "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""
        }
    ]
}
```

O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

- Para obter detalhes da API, consulte ListJobsem Referência de AWS CLI Comandos.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
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
```

- Para obter detalhes da API, consulte a ListJobsReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Glacier com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`ListTagsForVault`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `ListTagsForVault`.

.NET

SDK para .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
    /// <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;
    }
```

- Para obter detalhes da API, consulte ListTagsForVaulta Referência AWS SDK para .NET da API.

CLI

AWS CLI

O seguinte comando lista as tags aplicadas ao cofre `my-vault`:

```
aws glacier list-tags-for-vault --account-id - --vault-name my-vault
```

Saída:

```
{
    "Tags": {
        "date": "july2015",
        "id": "1234"
    }
}
```

O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

- Para obter detalhes da API, consulte ListTagsForVaultem Referência de AWS CLI Comandos.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Glacier com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`ListVaults`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `ListVaults`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Arquivar um arquivo, obter notificações e iniciar um trabalho

.NET

SDK para .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
/// <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;
}
```

- Para obter detalhes da API, consulte ListVaultsa Referência AWS SDK para .NET da API.

CLI

AWS CLI

O seguinte comando lista os cofres na conta e na região padrão:

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

Saída:

```
{
    "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"
        }
    ]
}
```

O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

- Para obter detalhes da API, consulte ListVaultsem Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
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);
        }
    }
}
```

- Para obter detalhes da API, consulte ListVaultsa Referência AWS SDK for Java 2.x da API.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
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
```

- Para obter detalhes da API, consulte a ListVaultsReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Glacier com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`SetVaultNotifications`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `SetVaultNotifications`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Arquivar um arquivo, obter notificações e iniciar um trabalho

CLI

AWS CLI

O seguinte comando configura as notificações do SNS para o cofre `my-vault`:

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

`notificationconfig.json` é um arquivo JSON na pasta atual que especifica um tópico do SNS e os eventos a serem publicados:

```
{
  "SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-vault",
```

```
  "Events": ["ArchiveRetrievalCompleted", "InventoryRetrievalCompleted"]
}
```

O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

- Para obter detalhes da API, consulte SetVaultNotificationsem Referência de AWS CLI Comandos.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
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
```

- Para obter detalhes da API, consulte a SetVaultNotificationsReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Glacier com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`UploadArchive`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `UploadArchive`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Arquivar um arquivo, obter notificações e iniciar um trabalho

.NET

SDK para .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
   /// <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;
       }
   }
```

- Para obter detalhes da API, consulte UploadArchivea Referência AWS SDK para .NET da API.

CLI

AWS CLI

O seguinte comando faz upload um arquivo na pasta atual com o nome `archive.zip` para um cofre chamado `my-vault`:

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

Saída:

```
{
    "archiveId": "kKB7ymWJVpPSwhGP6ycSOAekp9ZYe_--
zM_mw6k76ZFGEIWQX-ybtRDvc2VkPSDtfKmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hHO0bjbGehXTcApVud_wyDw",
    "checksum":
 "969fb39823836d81f0cc028195fcdbcbbe76cdde932d4646fa7de5f21e18aa67",
    "location": "/0123456789012/vaults/my-vault/archives/
kKB7ymWJVpPSwhGP6ycSOAekp9ZYe_--zM_mw6k76ZFGEIWQX-
ybtRDvc2VkPSDtfKmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hHO0bjbGehXTcApVud_wyDw"
}
```

O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

Para recuperar um arquivamento carregado, inicie um trabalho de recuperação com o comando aws glacier initiate-job.

- Para obter detalhes da API, consulte UploadArchiveem Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
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();
    }
}
```

- Para obter detalhes da API, consulte UploadArchivea Referência AWS SDK for Java 2.x da API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no Repositório de exemplos de código da AWS.

Crie o cliente.

```
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 };
```

Faça upload do arquivo.

```
// 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();
```

- Para obter mais informações, consulte o Guia do desenvolvedor do AWS SDK para JavaScript.
- Para obter detalhes da API, consulte UploadArchivea Referência AWS SDK para JavaScript da API.

SDK para JavaScript (v2)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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);
  }
});
```

- Para obter mais informações, consulte o Guia do desenvolvedor do AWS SDK para JavaScript.
- Para obter detalhes da API, consulte UploadArchivea Referência AWS SDK para JavaScript da API.

PowerShell

Ferramentas para PowerShell V4

Exemplo 1: carregar um único arquivo no cofre especificado, exibindo o ID do arquivo e a soma de verificação computada.

```
Write-GLCArchive -VaultName myvault -FilePath c:\temp\blue.bin
```

Saída:

```
FilePath                  ArchiveId             Checksum
--------                  ---------             --------
C:\temp\blue.bin          o9O9jUUs...TTX-TpIhQJw 79f3e...f4395b
```

Exemplo 2: carregar o conteúdo de uma hierarquia de pastas no cofre especificado na conta do usuário. Para cada arquivo carregado, o cmdlet emite o nome do arquivo, o ID do arquivo correspondente e a soma de verificação computada do arquivo.

```
Write-GLCArchive -VaultName myvault -FolderPath . -Recurse
```

Saída:

```
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
```

- Para obter detalhes da API, consulte UploadArchiveem Referência de Ferramentas da AWS para PowerShell cmdlet (V4).

Ferramentas para PowerShell V5

Exemplo 1: carregar um único arquivo no cofre especificado, exibindo o ID do arquivo e a soma de verificação computada.

```
Write-GLCArchive -VaultName myvault -FilePath c:\temp\blue.bin
```

Saída:

```
FilePath                    ArchiveId             Checksum
--------                    ---------             --------
C:\temp\blue.bin            o9O9jUUs...TTX-TpIhQJw 79f3e...f4395b
```

Exemplo 2: carregar o conteúdo de uma hierarquia de pastas no cofre especificado na conta do usuário. Para cada arquivo carregado, o cmdlet emite o nome do arquivo, o ID do arquivo correspondente e a soma de verificação computada do arquivo.

```
Write-GLCArchive -VaultName myvault -FolderPath . -Recurse
```

Saída:

```
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
```

- Para obter detalhes da API, consulte UploadArchiveem Referência de Ferramentas da AWS para PowerShell cmdlet (V5).

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
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
```

- Para obter detalhes da API, consulte a UploadArchiveReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Glacier com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`UploadMultipartPart`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `UploadMultipartPart`.

CLI

AWS CLI

O comando a seguir carrega a primeira parte de 1 MiB (1024 x 1024 bytes) de um arquivamento:

```
aws glacier upload-multipart-part --body part1 --range 'bytes
 0-1048575/*' --account-id - --vault-name my-vault --upload-
id 19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_lR7vgFuJV6NtcV5zpsJ
```

O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

O parâmetro de corpo segue um caminho para um arquivo da parte no sistema de arquivos local. O parâmetro de intervalo usa um intervalo de conteúdo HTTP indicando os bytes que a peça ocupa no arquivamento concluído. O ID de upload é retornado pelo comando `aws glacier initiate-multipart-upload` e também pode ser coletado usando `aws glacier list-multipart-uploads`.

Para obter mais informações sobre uploads de várias partes para o Amazon Glacier usando a AWS CLI, consulte Como usar o Amazon Glacier no Guia do usuário da CLI.AWS

- Para obter detalhes da API, consulte UploadMultipartPartem Referência de AWS CLI Comandos.

JavaScript

SDK para JavaScript (v2)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

Crie um carregamento fracionado com pedaços de 1 megabyte de um objeto 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);
        }
      });
    });
  }
});
```

- Para obter mais informações, consulte o Guia do desenvolvedor do AWS SDK para JavaScript.
- Para obter detalhes da API, consulte UploadMultipartParta Referência AWS SDK para JavaScript da API.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Glacier com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Cenários para o uso do Amazon Glacier AWS SDKs

Os exemplos de código a seguir mostram como implementar cenários comuns no Amazon Glacier com. AWS SDKs Esses casos mostram como realizar tarefas específicas chamando várias funções dentro do Amazon Glacier ou combinadas com outros Serviços da AWS. Cada cenário inclui um link para o código-fonte completo, onde podem ser encontradas instruções sobre como configurar e executar o código.

Os cenários têm como alvo um nível intermediário de experiência para ajudar você a compreender ações de serviço em contexto.

Exemplos

- Arquive um arquivo no Amazon Glacier, receba notificações e inicie um trabalho usando um SDK AWS
- Obtenha o conteúdo de arquivamento do Amazon Glacier e exclua o arquivo usando um SDK AWS

Arquive um arquivo no Amazon Glacier, receba notificações e inicie um trabalho usando um SDK AWS

O exemplo de código a seguir mostra como:

- Criar um cofre do Amazon Glacier.
- Configurar o cofre para publicar notificações em um tópico do Amazon SNS.
- Fazer upload de um arquivo para o cofre.
- Iniciar um trabalho de recuperação de arquivo.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

Crie uma classe que envolva operações do 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
```

Chame funções na classe wrapper para criar um cofre e fazer upload de um arquivo, depois configure o cofre para publicar notificações e iniciar um trabalho para recuperar o arquivo.

```
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.")
```

- Para ver detalhes da API, consulte os tópicos a seguir na Referência da API do SDK da AWS para Python (Boto3).
  - CreateVault
  - InitiateJob
  - ListJobs
  - ListVaults
  - SetVaultNotifications
  - UploadArchive

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Glacier com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Obtenha o conteúdo de arquivamento do Amazon Glacier e exclua o arquivo usando um SDK AWS

O exemplo de código a seguir mostra como:

- Listar trabalhos para um cofre do Amazon Glacier e obter o status do trabalho.
- Obter a saída de um trabalho de recuperação de arquivo concluído.
- Excluir um arquivo.
- Excluir um cofre.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

Crie uma classe que envolva operações do 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
```

Chame funções na classe wrapper para obter o conteúdo de arquivo de um trabalho concluído e, em seguida, exclua o arquivo.

```
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)
```

- Para ver detalhes da API, consulte os tópicos a seguir na Referência da API do SDK da AWS para Python (Boto3).
  - DeleteArchive
  - DeleteVault
  - GetJobOutput
  - ListJobs

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Glacier com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Segurança no Amazon Glacier

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de um data center e de uma arquitetura de rede criados para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O modelo de responsabilidade compartilhada descreve isto como segurança da nuvem e segurança na nuvem.

- Segurança da nuvem — AWS é responsável por proteger a infraestrutura que executa AWS os serviços no Nuvem AWS. AWS também fornece serviços que você pode usar com segurança. A eficácia da nossa segurança é regularmente testada e verificada por auditores de terceiros como parte dos Programas de conformidade da AWS. Para saber mais sobre os programas de conformidade que se aplicam ao Amazon Glacier, consulte Serviços da AWS no escopo pelo Programa de Conformidade.
- Segurança na nuvem — Sua responsabilidade é determinada pelo AWS serviço que você usa. Também existe a responsabilidade por outros fatores, incluindo a confidencialidade de dados, os requisitos da organização e as leis e regulamentos aplicáveis.

Esta documentação ajudará você a entender como aplicar o modelo de responsabilidade compartilhada ao usar o Amazon Glacier. Os tópicos a seguir mostram como configurar o Amazon Glacier para atender aos seus objetivos de segurança e conformidade. Você também aprenderá a usar outros AWS serviços que podem ajudá-lo a monitorar e proteger seus recursos do Amazon Glacier.

Tópicos

- Proteção de dados no Amazon Glacier
- Gerenciamento de identidade e acesso para o Amazon Glacier
- Conectar e monitorar no Amazon Glacier
- Validação de conformidade do Amazon Glacier
- Resiliência no Amazon Glacier
- Segurança da infraestrutura no Amazon Glacier

Proteção de dados no Amazon Glacier

O Amazon Glacier fornece armazenamento em nuvem altamente durável para arquivamento de dados e backup de longo prazo. O Amazon Glacier foi projetado para oferecer 99,999999999% de durabilidade e fornece recursos abrangentes de segurança e conformidade que podem ajudá-lo a atender as rigorosas exigências regulatórias. O Amazon Glacier armazena dados de forma redundante em várias zonas de disponibilidade (AZ) da AWS e em vários dispositivos dentro de cada AZ. Para aumentar a durabilidade, o Amazon Glacier armazena seus dados de maneira sincronizada em várias zonas de disponibilidade antes de confirmar um upload bem-sucedido.

Para obter mais informações sobre a infraestrutura de nuvem global da AWS, consulte Infraestrutura global.

Para fins de proteção de dados, recomendamos que você proteja as credenciais Conta da AWS e forneça a usuário individuais, grupos ou funções somente as permissões necessárias para cumprir suas responsabilidades no trabalho..

Se você precisar de módulos criptográficos validados pelo FIPS 140-2 ao acessar AWS por meio de uma interface de linha de comando ou uma API, use um endpoint FIPS. Para ter mais informações sobre endpoints do FIPS disponíveis, consulte Federal Information Processing Standard (FIPS) 140-2.

Tópicos

- Criptografia de dados
- Gerenciamento de chaves
- Privacidade do tráfego entre redes

Criptografia de dados

Proteção de dados protege os dados em trânsito (à medida que são transferidos de e para o Amazon Glacier) e em repouso (enquanto estão armazenados em datacenters da AWS). Você pode proteger os dados em trânsito que são carregados diretamente no Amazon Glacier usando Secure Sockets Layer (SSL) ou criptografia do lado do cliente.

Você também pode acessar o Amazon Glacier por meio do Amazon S3. O Amazon S3 oferece suporte à configuração do ciclo de vida em um bucket do Amazon S3, o que permite a você transitar objetos para a classe de armazenamento do Amazon Glacier para arquivamento. Os

dados em trânsito entre o Amazon S3 e o Amazon Glacier por meio de políticas de ciclo de vida são criptografados usando SSL.

Os dados em repouso armazenados no Amazon Glacier são criptografados automaticamente no lado do servidor usando o Advanced Encryption Standard de 256 bits (AES-256) com chaves mantidas pela AWS. Se preferir gerenciar suas próprias chaves, você também poderá usar a criptografia do lado do cliente antes de armazenar dados no Amazon Glacier. Para obter informações sobre o recurso de criptografia padrão do Amazon S3, consulte Criptografia padrão do Amazon S3 no Guia do Amazon Simple Storage Service.

Gerenciamento de chaves

A criptografia do lado do servidor trata da criptografia de dados em repouso, ou seja, o Amazon Glacier criptografa seus dados à medida que os grava em seus datacenters e os descriptografa para você quando você os acessa. Contanto que você autentique sua solicitação e tenha permissões de acesso, não há diferença na forma de acesso aos dados criptografados ou não criptografados.

Os dados em repouso no Amazon Glacier são criptografados automaticamente no lado do servidor usando AES-256, por meio de chaves mantidas pela AWS. Como proteção adicional, a AWS criptografa a chave com uma chave raiz que é alternada regularmente.

Privacidade do tráfego entre redes

O acesso ao Amazon Glacier pela rede acontece por meio de APIs publicadas pela AWS. Os clientes devem ser compatíveis com o Transport Layer Security (TLS) 1.2. Recomendamos usar o TLS 1.3 ou posterior. Os clientes também devem ter suporte a pacotes de criptografia com sigilo de encaminhamento perfeito (PFS) como Ephemeral Diffie-Hellman (DHE) ou Ephemeral Elliptic Curve Diffie-Hellman (ECDHE). A maioria dos sistemas modernos como Java 7 e versões posteriores oferece suporte a esses modos. Além disso, devem ser assinadas solicitações usando um ID de chave de acesso e uma chave de acesso secreta associadas a uma entidade principal do IAM ou pode usar AWS Security Token ServiceAWS STSpara gerar credenciais de segurança temporárias visando assinar solicitações.

Endpoints da VPC

Um endpoint de nuvem privada virtual (VPC) permite que você conecte seu VPC de forma privada a serviços suportados AWS e serviços de endpoint da VPC alimentados por PrivateLink AWS sem a necessidade de um gateway da Internet, dispositivo NAT, conexão VPN ou conexão AWS Direct Connect. Embora o Amazon Glacier não ofereça suporte a endpoints da VPC diretamente, você

pode aproveitar os endpoints da VPC do Amazon Simple Storage Service (Amazon S3) se acessar o Amazon Glacier como um nível de armazenamento integrado ao Amazon S3.

Para obter mais informações sobre a configuração do ciclo de vida do Amazon S3 e a transição de objetos para a classe de armazenamento do Amazon Glacier, consulte Gerenciamento do ciclo de vida de objetos e Transição de objetos no Guia do usuário do Amazon Simple Storage Service. Para obter mais informações sobre o endpoints da VPC, consulte Endpoints da VPC no Manual do usuário da Amazon VPC.

Gerenciamento de identidade e acesso para o Amazon Glacier

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. Os administradores do IAM controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) a usar os recursos do Amazon Glacier. O IAM é um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

Tópicos

- Público
- Autenticação com identidades
- Gerenciar o acesso usando políticas
- Como funciona o Amazon Glacier com o IAM
- Exemplos de políticas baseadas em identidade para o Amazon Glacier
- Exemplos de política baseada em recursos para o Amazon Glacier
- Solução de problemas de identidade e acesso do Amazon Glacier
- Referência de permissões da API

Público

A forma como você usa AWS Identity and Access Management (IAM) difere com base na sua função:

- Usuário do serviço: solicite permissões ao seu administrador se você não conseguir acessar os atributos (consulte Solução de problemas de identidade e acesso do Amazon Glacier).

- Administrador do serviço: determine o acesso do usuário e envie solicitações de permissão (consulte Como funciona o Amazon Glacier com o IAM)
- Administrador do IAM: escreva políticas para gerenciar o acesso (consulte Exemplos de políticas baseadas em identidade para o Amazon Glacier)

Autenticação com identidades

A autenticação é a forma como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado como usuário do IAM ou assumindo uma função do IAM. Usuário raiz da conta da AWS

Você pode fazer login como uma identidade federada usando credenciais de uma fonte de identidade como Centro de Identidade do AWS IAM (IAM Identity Center), autenticação de login único ou credenciais. Google/Facebook Para ter mais informações sobre como fazer login, consulte Como fazer login em sua Conta da AWS no Guia do usuário do Início de Sessão da AWS .

Para acesso programático, AWS fornece um SDK e uma CLI para assinar solicitações criptograficamente. Para ter mais informações, consulte AWS Signature Version 4 para solicitações de API no Guia do usuário do IAM.

Conta da AWS usuário root

Ao criar um Conta da AWS, você começa com uma identidade de login chamada usuário Conta da AWS raiz que tem acesso completo a todos Serviços da AWS os recursos. É altamente recomendável não usar o usuário-raiz em tarefas diárias. Consulte as tarefas que exigem credenciais de usuário-raiz em Tarefas que exigem credenciais de usuário-raiz no Guia do usuário do IAM.

Identidade federada

Como prática recomendada, exija que os usuários humanos usem a federação com um provedor de identidade para acessar Serviços da AWS usando credenciais temporárias.

Uma identidade federada é um usuário do seu diretório corporativo, provedor de identidade da web ou Directory Service que acessa Serviços da AWS usando credenciais de uma fonte de identidade. As identidades federadas assumem funções que oferecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, recomendamos Centro de Identidade do AWS IAM. Para saber mais, consulte O que é o IAM Identity Center? no Guia do usuário do Centro de Identidade do AWS IAM .

Usuários e grupos do IAM

Um usuário do IAM é uma identidade com permissões específicas para uma única pessoa ou aplicação. É recomendável usar credenciais temporárias, em vez de usuários do IAM com credenciais de longo prazo. Para obter mais informações, consulte Exigir que usuários humanos usem a federação com um provedor de identidade para acessar AWS usando credenciais temporárias no Guia do usuário do IAM.

Um grupo do IAM especifica um conjunto de usuários do IAM e facilita o gerenciamento de permissões para grandes conjuntos de usuários. Para ter mais informações, consulte Casos de uso de usuários do IAM no Guia do usuário do IAM.

Perfis do IAM

Uma perfil do IAM é uma identidade com permissões específicas que oferece credenciais temporárias. Você pode assumir uma função mudando de um usuário para uma função do IAM (console) ou chamando uma operação de AWS API AWS CLI ou. Para saber mais, consulte Métodos para assumir um perfil no Manual do usuário do IAM.

Os perfis do IAM são úteis para acesso de usuário federado, permissões de usuário do IAM temporárias, acesso entre contas, acesso entre serviços e aplicações em execução no Amazon EC2. Consulte mais informações em Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Gerenciar o acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política define permissões quando associada a uma identidade ou recurso. AWS avalia essas políticas quando um diretor faz uma solicitação. A maioria das políticas é armazenada AWS como documentos JSON. Para ter mais informações sobre documentos de política JSON, consulte Visão geral das políticas JSON no Guia do usuário do IAM.

Por meio de políticas, os administradores especificam quem tem acesso a que, definindo qual entidade principal pode realizar ações em quais recursos e sob quais condições.

Por padrão, usuários e perfis não têm permissões. Um administrador do IAM cria políticas do IAM e as adiciona aos perfis, os quais os usuários podem então assumir. As políticas do IAM definem permissões, independentemente do método usado para realizar a operação.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissão JSON que você anexa a uma identidade (usuário, grupo ou perfil). Essas políticas controlam quais ações as identidades podem realizar, em quais recursos e sob quais condições. Para saber como criar uma política baseada em identidade, consulte Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente no Guia do Usuário do IAM.

As políticas baseadas em identidade podem ser políticas em linha (incorporadas diretamente em uma única identidade) ou políticas gerenciadas (políticas autônomas anexadas a várias identidades). Para saber como escolher entre uma política gerenciada e políticas em linha, consulte Escolher entre políticas gerenciadas e políticas em linha no Guia do usuário do IAM.

Políticas baseadas em recursos

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. Entre os exemplos estão políticas de confiança de perfil do IAM e políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. É necessário especificar uma entidade principal em uma política baseada em recursos.

Políticas baseadas em recursos são políticas em linha localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas do IAM em uma política baseada em recursos.

Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais que podem definir o máximo de permissões concedidas por tipos de políticas mais comuns:

- Limites de permissões: definem o número máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM. Para saber mais sobre limites de permissões, consulte Limites de permissões para identidades do IAM no Guia do usuário do IAM.
- Políticas de controle de serviço (SCPs) — Especifique as permissões máximas para uma organização ou unidade organizacional em AWS Organizations. Para saber mais, consulte Políticas de controle de serviço no Guia do usuário do AWS Organizations .
- Políticas de controle de recursos (RCPs) — Defina o máximo de permissões disponíveis para recursos em suas contas. Para obter mais informações, consulte Políticas de controle de recursos (RCPs) no Guia AWS Organizations do usuário.

- Políticas de sessão: políticas avançadas transmitidas como um parâmetro durante a criação de uma sessão temporária para um perfil ou um usuário federado. Para saber mais, consulte Políticas de sessão no Guia do usuário do IAM.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de políticas estão envolvidos, consulte Lógica de avaliação de políticas no Guia do usuário do IAM.

Como funciona o Amazon Glacier com o IAM

Antes de usar o IAM para gerenciar o acesso ao Amazon Glacier, saiba quais atributos do IAM estão disponíveis para uso.

Atributos do IAM que você pode usar com o Amazon Glacier

Recurso do IAM	Suporte ao Amazon Glacier
Políticas baseadas em identidade	Sim
Políticas baseadas em atributos	Sim
Ações de políticas	Sim
Recursos de políticas	Sim
Chaves de condição de política (específicas do serviço)	Sim
ACLs	Não
ABAC (tags em políticas)	Não
Credenciais temporárias	Sim
Permissões de entidade principal	Não

Recurso do IAM	Suporte ao Amazon Glacier
Perfis de serviço	Não
Funções vinculadas ao serviço	Não

Para ter uma visão de alto nível de como o Amazon Glacier e AWS outros serviços funcionam com a maioria dos recursos do IAM, AWS consulte os serviços que funcionam com o IAM no Guia do usuário do IAM.

Usar políticas baseadas em identidade para o Amazon Glacier

Compatível com políticas baseadas em identidade: sim

As políticas baseadas em identidade são documentos de políticas de permissões JSON que podem ser anexados a uma identidade, como usuário do IAM, grupo de usuários ou perfil. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente no Guia do Usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações e recursos permitidos ou negados, assim como as condições sob as quais as ações são permitidas ou negadas. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte Referência de elemento de política JSON do IAM no Guia do usuário do IAM.

Exemplos de políticas baseadas em identidade para o Amazon Glacier

Consulte exemplos de políticas baseadas em identidade do Amazon Glacier em Exemplos de políticas baseadas em identidade para o Amazon Glacier.

Políticas baseadas em recursos no Amazon Glacier

Compatível com políticas baseadas em recursos: sim

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico.

Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificado pode executar nesse atributo e em que condições. É necessário especificar uma entidade principal em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Para permitir o acesso entre contas, é possível especificar uma conta inteira ou as entidades do IAM em outra conta como a entidade principal em uma política baseada em recursos. Consulte mais informações em Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

O serviço Amazon Glacier oferece suporte a apenas um tipo de política baseada em recursos chamada política de cofre, vinculada a um cofre. Essa política define quais diretores podem realizar ações no cofre.

Você pode usar políticas de cofre do Amazon Glacier para gerenciar permissões das seguintes maneiras:

- Gerencie permissões de usuário na conta em uma única política de cofre, em vez de políticas de usuário individuais.
- Gerencie permissões entre contas como uma alternativa a usar funções do &IAM.

Exemplos de políticas baseadas em recursos no Amazon Glacier

Para ver exemplos de políticas baseadas em recursos do Amazon Glacier, consulte Exemplos de política baseada em recursos para o Amazon Glacier.

Ações de políticas para o Amazon Glacier

Compatível com ações de políticas: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Action` de uma política JSON descreve as ações que podem ser usadas para permitir ou negar acesso em uma política. Incluem ações em uma política para conceder permissões para executar a operação associada.

Para ver uma lista das ações do Amazon Glacier, consulte Ações definidas pelo Amazon Glacier na Referência de autorização do serviço.

As ações de políticas no Amazon Glacier usam o seguinte prefixo antes da ação:

```
glacier
```

Para especificar várias ações em uma única instrução, separe-as com vírgulas.

```
"Action": [
            "glacier:CreateVault",
            "glacier:DescribeVault",
            "glacier:ListVaults"
         ]
```

Você também pode especificar várias ações usando caracteres-curinga (*). Por exemplo, para especificar todas as ações que começam com a palavra `Describe`, inclua a seguinte ação:

```
"Action": "glacier:GetVault*"
```

Consulte exemplos de políticas baseadas em identidade do Amazon Glacier em Exemplos de políticas baseadas em identidade para o Amazon Glacier.

Recursos de políticas para o Amazon Glacier

Compatível com recursos de políticas: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento de política JSON `Resource` especifica o objeto ou os objetos aos quais a ação se aplica. Como prática recomendada, especifique um recurso usando seu nome do recurso da Amazon (ARN). Para ações que não oferecem compatibilidade com permissões em nível de recurso, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

```
"Resource": "*"
```

Para ver uma lista dos tipos de recursos do Amazon Glacier e seus ARNs, consulte Recursos definidos pelo Amazon Glacier na Referência de autorização de serviço. Para saber com quais ações você pode especificar o ARN de cada recurso, consulte Ações definidas pelo Amazon Glacier.

No Amazon Glacier, o recurso primário é um cofre. O Amazon Glacier oferece suporte a políticas somente no nível do cofre. Ou seja, em uma política do IAM, o `Resource` valor que você especifica pode ser um cofre específico ou um conjunto de cofres em uma região específica. AWS O Amazon Glacier não dá suporte a permissões no nível de arquivo.

Para todas as ações do Amazon Glacier, o `Resource` especifica o cofre para o qual você deseja conceder as permissões. Esses recursos têm nomes de recursos exclusivos da Amazon (ARNs) associados a eles, conforme mostrado na tabela a seguir, e você pode usar um caractere curinga (*) no ARN para combinar nomes de cofres que começam com o mesmo prefixo.

O Amazon Glacier fornece um conjunto de operações para trabalhar com recursos dele. Para obter informações sobre as operações disponíveis, consulte Referência de API para Amazon Glacier.

Algumas ações da API do Amazon Glacier oferecem suporte a vários recursos. Por exemplo, `glacier:AddTagsToVault` acessa examplevault1 e examplevault2, portanto, uma entidade principal deve ter permissões para acessar os dois recursos. Para especificar vários recursos em uma única instrução, separe-os ARNs com vírgulas.

```
"Resource": [
""arn:aws:glacier:us-west-2:123456789012:vaults/examplevault1",",
""arn:aws:glacier:us-west-2:123456789012:vaults/examplevault2","
]
```

Chaves de condição de política do Amazon Glacier

Compatível com chaves de condição de política específicas de serviço: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Condition` especifica quando as instruções são executadas com base em critérios definidos. É possível criar expressões condicionais que usem agentes de condição, como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação. Para ver todas as chaves de condição AWS globais, consulte as chaves de contexto de condição AWS global no Guia do usuário do IAM.

Para ver uma lista de chaves de condição do Amazon Glacier, consulte Chaves de condição do Amazon Glacier na Referência de autorização do serviço. Para saber com quais ações e recursos é possível usar a chave de condição, consulte Ações definidas pelo Amazon Glacier.

Para obter exemplos de como usar chaves de condição específicas do Glacier , consulte Políticas do Vault Lock.

ACLs no Amazon Glacier

Suportes ACLs: Não

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

ABAC com o Amazon Glacier

Oferece compatibilidade com ABAC (tags em políticas): não

O controle de acesso por atributo (ABAC) é uma estratégia de autorização que define permissões com base em atributos chamados de tags. Você pode anexar tags a entidades e AWS recursos do IAM e, em seguida, criar políticas ABAC para permitir operações quando a tag do diretor corresponder à tag no recurso.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no elemento de condição de uma política usando as `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* ou chaves de condição `aws:TagKeys`.

Se um serviço for compatível com as três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço for compatível com as três chaves de condição somente para alguns tipos de recursos, o valor será Parcial

Para saber mais sobre o ABAC, consulte Definir permissões com autorização do ABAC no Guia do usuário do IAM. Para visualizar um tutorial com etapas para configurar o ABAC, consulte Usar controle de acesso por atributo (ABAC) no Guia do usuário do IAM.

Usar credenciais temporárias com o Amazon Glacier

Compatível com credenciais temporárias: sim

As credenciais temporárias fornecem acesso de curto prazo aos AWS recursos e são criadas automaticamente quando você usa a federação ou troca de funções. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para ter mais informações, consulte Credenciais de segurança temporárias no IAM e Serviços da Serviços da AWS que funcionam com o IAM no Guia do usuário do IAM.

Permissões de entidade principal entre serviços para o Amazon Glacier

Compatível com o recurso de encaminhamento de sessões de acesso (FAS): Não

As sessões de acesso direto (FAS) usam as permissões do principal chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) de fazer solicitações aos serviços posteriores. Para obter detalhes da política ao fazer solicitações de FAS, consulte Sessões de acesso direto.

Perfis de serviço para o Amazon Glacier

Compatível com perfis de serviço: não

O perfil de serviço é um perfil do IAM que um serviço assume para executar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para saber mais, consulte Criar um perfil para delegar permissões a um AWS service (Serviço da AWS) no Guia do Usuário do IAM.

> **Warning**
>
> A alteração das permissões de um perfil de serviço pode interromper a funcionalidade do Amazon Glacier. Só edite os perfis de serviço quando o Amazon Glacier orientar a fazê-lo.

Funções vinculadas ao serviço para o Amazon Glacier

Compatível com perfis vinculados ao serviço: Não

Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode assumir o perfil de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não editar as permissões para perfis vinculados ao serviço.

Para obter detalhes sobre como criar ou gerenciar perfis vinculados a serviços, consulte Serviços da AWS que funcionam com o IAM. Encontre um serviço na tabela que inclua um `Yes` na coluna Perfil vinculado ao serviço. Escolha o link Sim para visualizar a documentação do perfil vinculado a serviço desse serviço.

Exemplos de políticas baseadas em identidade para o Amazon Glacier

Por padrão, usuários e funções não têm permissão para criar ou modificar recursos do Amazon Glacier. Para conceder permissão aos usuários para executar ações nos recursos que eles precisam, um administrador do IAM pode criar políticas do IAM.

Para aprender a criar uma política baseada em identidade do IAM ao usar esses documentos de política em JSON de exemplo, consulte Criar políticas do IAM (console) no Guia do usuário do IAM.

Para obter detalhes sobre ações e tipos de recursos definidos pelo Amazon Glacier, incluindo o formato de cada um dos ARNs tipos de recursos, consulte Ações, recursos e chaves de condição para o Amazon Glacier na Referência de autorização de serviço.

Veja a seguir um exemplo de política que concede permissões para três ações relacionadas ao Amazon Glacier Vault (`glacier:CreateVault`, `glacier:DescribeVault` e`glacier:ListVaults`) em um recurso, usando o Amazon Resource Name (ARN) que identifica todos os cofres na região. `us-west-2` AWS ARNs identifique AWS recursos de forma exclusiva. Para obter mais informações sobre o ARNs uso com o Amazon Glacier, consulte. Recursos de políticas para o 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/*"
      }
```

```
        ]
    }
```

A política concede permissões para criar, listar e obter descrições de cofres na região `us-west-2`. O caractere curinga (*) ao final do ARN do significa que essa instrução pode corresponder a qualquer nome de cofre.

> **Important**
>
> Ao conceder permissões para criar um cofre usando a operação `glacier:CreateVault`, você deve especificar um caractere curinga (*) porque você não saberá o nome do cofre até criar o cofre.

Tópicos

- Práticas recomendadas de política
- Usando o console do Amazon Glacier
- Permitir que os usuários visualizem suas próprias permissões
- Exemplos de política gerenciada pelo cliente

Práticas recomendadas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir recursos do Amazon Glacier em sua conta. Essas ações podem incorrer em custos para sua Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e avance para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso. Para saber mais, consulte Políticas gerenciadas pela AWS ou Políticas gerenciadas pela AWS para funções de trabalho no Guia do usuário do IAM.
- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em recursos específicos sob condições específicas, também

conhecidas como permissões de privilégio mínimo. Para saber mais sobre como usar o IAM para aplicar permissões, consulte Políticas e permissões no IAM no Guia do usuário do IAM.

- Use condições nas políticas do IAM para restringir ainda mais o acesso: é possível adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, é possível escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como CloudFormation. Para saber mais, consulte Elementos da política JSON do IAM: condição no Guia do usuário do IAM.
- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de cem verificações de política e recomendações práticas para ajudar a criar políticas seguras e funcionais. Para saber mais, consulte Validação de políticas do IAM Access Analyzer no Guia do Usuário do IAM.
- Exigir autenticação multifator (MFA) — Se você tiver um cenário que exija usuários do IAM ou um usuário root, ative Conta da AWS a MFA para obter segurança adicional. Para exigir MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para saber mais, consulte Configuração de acesso à API protegido por MFA no Guia do Usuário do IAM.

Para saber mais sobre as práticas recomendadas do IAM, consulte Práticas recomendadas de segurança no IAM no Guia do usuário do IAM.

Usando o console do Amazon Glacier

Para acessar o console do Amazon Glacier, você deve ter um conjunto mínimo de permissões. Com essas permissões, é possível listar e visualizar detalhes sobre os recursos do Amazon Glacier em sua Conta da AWS. Caso crie uma política baseada em identidade mais restritiva que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis) com essa política.

Você não precisa permitir permissões mínimas do console para usuários que estão fazendo chamadas somente para a API AWS CLI ou para a AWS API. Em vez disso, permita o acesso somente a ações que correspondam à operação de API que estiverem tentando executar.

O console do Amazon Glacier fornece um ambiente integrado para você criar e gerenciar cofres do Amazon Glacier. No mínimo, as identidades do IAM criadas devem receber permissões para a ação

`glacier:ListVaults` de visualizar o console do Amazon Glacier, conforme mostrado no exemplo a seguir.

JSON

```
{
      "Version":"2012-10-17",
      "Statement": [
         {
            "Action": [
            "glacier:ListVaults"
            ],
            "Effect": "Allow",
            "Resource": "*"
         }
      ]
   }
```

AWS aborda muitos casos de uso comuns fornecendo políticas autônomas do IAM que são criadas e administradas pela AWS. As políticas gerenciadas concedem permissões necessárias para casos de uso comuns, de maneira que você possa evitar a necessidade de investigar quais permissões são necessárias. Para obter mais informações, consulte Políticas gerenciadas pela AWS no Guia do usuário do IAM.

As seguintes políticas AWS gerenciadas, que você pode associar aos usuários em sua conta, são específicas do Amazon Glacier:

- AmazonGlacierReadOnlyAccess— Concede acesso somente de leitura ao Amazon Glacier por meio do. Console de gerenciamento da AWS
- AmazonGlacierFullAccess— Concede acesso total ao Amazon Glacier por meio do. Console de gerenciamento da AWS

Você também pode criar suas próprias políticas do IAM personalizadas e conceder permissões para ações e recursos de API do Amazon Glacier. Você pode anexar essas políticas personalizadas a perfis personalizados do IAM que você cria para seus cofres do Amazon Glacier.

Ambas as políticas AWS gerenciadas do Amazon Glacier discutidas na próxima seção concedem permissões para. `glacier:ListVaults`

Para obter informações, consulte Adicionar permissões a um usuário no Guia do usuário do IAM.

Permitir que os usuários visualizem suas próprias permissões

Este exemplo mostra como criar uma política que permita que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando a API AWS CLI ou 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": "*"
        }
    ]
```

```
}
```

Exemplos de política gerenciada pelo cliente

Nesta seção, você encontrará exemplos de políticas de usuário que concedem permissões para diversas ações do Amazon Glacier. Essas políticas funcionam quando você usa a API REST do Amazon Glacier, o Amazon, o ou SDKs, se aplicável AWS CLI, o console de gerenciamento do Amazon Glacier.

> **Note**
>
> Todos os exemplos usam a região Oeste dos EUA (Oregon) (`us-west-2`) e contêm uma conta fictícia. IDs

Exemplos

- Exemplo 1: permitir que um usuário faça download de arquivos de um cofre
- Exemplo 2: permitir que um usuário crie um cofre e configure notificações
- Exemplo 3: permitir que um usuário faça upload para um cofre específico
- Exemplo 4: conceder a um usuário permissões completas em um cofre específico

Exemplo 1: permitir que um usuário faça download de arquivos de um cofre

Para fazer download de um arquivo, você primeiro inicia um trabalho a fim de recuperar o arquivo. Depois que o trabalho de recuperação for concluído, você poderá fazer download dos dados. A política de exemplo a seguir concede permissões para a ação `glacier:InitiateJob` a fim de iniciar um trabalho (que permite ao usuário recuperar um arquivo ou um inventário de cofre do cofre) e permissões para a ação `glacier:GetJobOutput` a fim de fazer download dos dados recuperados. A política também concede permissões para realizar a ação `glacier:DescribeJob`, de maneira que o usuário possa obter o status do trabalho. Para obter mais informações, consulte Initiate Job (trabalhos POST).

A política concede essas permissões em um cofre chamado `examplevault`. Você pode obter o ARN do cofre por meio do console do Amazon Glacier ou, de maneira programática, chamando as ações de API Describe Vault (GET vault) ou List Vaults (GET vaults).

Exemplo 2: permitir que um usuário crie um cofre e configure notificações

A política de exemplo a seguir concede permissões para criar um cofre na região EUA-oeste-2, conforme especificado no elemento `Resource`, e configurar notificações. Para obter mais informações sobre como trabalhar com notificações, consulte Configurar notificações do cofre no Amazon Glacier. A política também concede permissões para listar cofres na AWS região e obter uma descrição específica do cofre.

> **Important**
>
> Ao conceder permissões para criar um cofre usando a operação `glacier:CreateVault`, você deve especificar um caractere curinga (*) no valor `Resource` porque você não saberá o nome do cofre até criar o cofre.

Exemplo 3: permitir que um usuário faça upload para um cofre específico

A política de exemplo a seguir concede permissões a fim de fazer upload de arquivos para um determinado cofre na região EUA-oeste-2. Essas permissões permitem que um usuário faça upload de um arquivo ao mesmo tempo usando a operação da API Upload Archive (POST archive) ou em partes usando a operação da API Initiate Multipart Upload (POST multipart-uploads).

Exemplo 4: conceder a um usuário permissões completas em um cofre específico

A política de exemplo a seguir concede permissões a todas as ações do Amazon Glacier em um cofre chamado `examplevault`.

Exemplos de política baseada em recursos para o Amazon Glacier

Um cofre do Amazon Glacier pode ter uma política de acesso ao cofre e uma política do Vault Lock associada. Uma política de acesso ao cofre do Amazon Glacier é uma política baseada em recursos que você pode usar para gerenciar permissões para o seu cofre. Uma política do Vault Lock é uma política de acesso ao cofre que pode ser bloqueada. Depois que você bloqueia uma política do Vault Lock, ela não pode ser alterada. Você pode usar a política do Vault Lock para impor controles de conformidade.

Tópicos

- Políticas de acesso ao cofre
- Políticas do Vault Lock

Políticas de acesso ao cofre

Uma política de acesso ao cofre do Amazon Glacier é uma política baseada em recursos que você pode usar para gerenciar permissões para o seu cofre.

Você pode criar uma política de acesso ao cofre para cada cofre a fim de gerenciar permissões. Você pode modificar permissões em uma política de acesso ao cofre a qualquer momento. O Amazon Glacier também dá suporte a uma política do Vault Lock em cada cofre que, depois de bloqueado por você, não poderá ser alterada. Para obter mais informações sobre como trabalhar com políticas do Vault Lock, consulte Políticas do Vault Lock.

Exemplos

- Exemplo 1: conceder permissões entre contas para ações do Amazon Glacier específicas
- Exemplo 2: conceder permissões entre contas para operações de exclusão MFA

Exemplo 1: conceder permissões entre contas para ações do Amazon Glacier específicas

O exemplo de política a seguir concede permissões entre contas a duas Contas da AWS para um conjunto de operações do Amazon Glacier em um cofre chamado `examplevault`.

> **Note**
>
> A conta que detém o cofre é cobrada por todos os custos associados ao cofre. Todas as solicitações, transferências de dados e custos de recuperação feitos por contas externas permitidas são cobrados na conta que detém o cofre.

Exemplo 2: conceder permissões entre contas para operações de exclusão MFA

Você pode usar a Autenticação Multifator (MFA) para proteger os recursos do Amazon Glacier. Para oferecer um nível extra de segurança, o MFA exige que os usuários provem a posse física de um dispositivo MFA fornecendo um código MFA válido. Para obter mais informações sobre como

configurar acesso MFA, consulte Configurar acesso à API protegido por MFA no Guia do Usuário do IAM.

O exemplo de política concede a um Conta da AWS usuário com credenciais temporárias permissão para excluir arquivos de um cofre chamado examplevault, desde que a solicitação seja autenticada com um dispositivo de MFA. A política usa a chave de condição `aws:MultiFactorAuthPresent` para especificar esse requisito adicional. Para obter mais informações, consulte Chaves disponíveis para condições no Guia do Usuário DO IAM.

Políticas do Vault Lock

Um cofre do Amazon Glacier pode ter uma política de acesso ao cofre baseada em recursos e uma política do Vault Lock anexada a ele. Uma política do Vault Lock é uma política de acesso ao cofre que você pode bloquear. Usar uma política do Vault Lock pode ajudar você a impor requisitos regulatórios e de conformidade. O Amazon Glacier oferece um conjunto de operações da API para gerenciar as políticas do Vault Lock, consulte Bloquear um cofre usando a API do Amazon Glacier.

Como exemplo de uma política do Vault Lock, suponhamos que você precise manter arquivos por um ano antes de excluí-los. Para implementar esse requisito, você pode criar uma política do Vault Lock que negue a usuários permissões para excluir um arquivo até que o arquivo exista por um ano. Você pode testar essa política antes de bloqueá-la. Depois de bloquear a política, ela se tornará imutável. Para obter mais informações sobre o processo de bloqueio, consulte Políticas do Vault Lock. Se quiser gerenciar outras permissões de usuário que possam ser alteradas, você poderá usar a política de acesso ao cofre (consulte Políticas de acesso ao cofre).

Você pode usar a API do Amazon Glacier SDKs AWS CLI, a Amazon ou o console do Amazon Glacier para criar e gerenciar políticas do Vault Lock. Para obter uma lista de ações do Amazon Glacier permitidas para políticas baseadas em recursos do cofre, consulte Referência de permissões da API.

Exemplos

- Exemplo 1: negar permissões de exclusão para arquivos com menos de 365 dias
- Exemplo 2: negar permissões de exclusão com base em uma tag

Exemplo 1: negar permissões de exclusão para arquivos com menos de 365 dias

Suponhamos que você tenha um requisito regulatório para reter arquivos por até um ano antes de excluí-los. Você pode impor esse requisito implementando a política do Vault Lock a seguir. A política negará a ação `glacier:DeleteArchive` no cofre examplevault se o arquivo que estiver sendo

excluído tiver menos de um ano. A política usa a chave de condição `ArchiveAgeInDays` específica do Amazon Glacier para impor o requisito de retenção de um ano.

Exemplo 2: negar permissões de exclusão com base em uma tag

Suponhamos que você tenha uma regra de retenção baseada em tempo de que um arquivo possa ser excluído caso tenha menos de um ano. Ao mesmo tempo, suponhamos que você precise impor uma retenção legal sobre os arquivos a fim de evitar a exclusão ou a modificação por uma duração indefinida durante uma investigação legal. Nesse caso, a retenção legal tem precedência sobre a regra de retenção baseada em tempo especificada na política do Vault Lock.

Para impor essas duas regras, a seguinte política de exemplo tem duas instruções:

- A primeira instrução nega permissões de exclusão a todos, bloqueando o cofre. Esse bloqueio é realizado usando a tag `LegalHold`.
- A segunda instrução concede permissões de exclusão quando o arquivo tem menos de 365 dias. Mas, mesmo quando arquivos tiverem menos de 365 dias, ninguém poderá excluí-los quando a condição da primeira instrução for atendida.

Solução de problemas de identidade e acesso do Amazon Glacier

Use as informações a seguir para ajudar a diagnosticar e corrigir problemas comuns que você possa encontrar ao trabalhar com o Amazon Glacier e o IAM.

Tópicos

- Não tenho autorização para realizar uma ação no Amazon Glacier
- Não estou autorizado a realizar iam: PassRole
- Quero permitir que pessoas de fora da minha tenham acesso Conta da AWS aos meus recursos do Amazon Glacier

Não tenho autorização para realizar uma ação no Amazon Glacier

Se você receber uma mensagem de erro informando que não tem autorização para executar uma ação, suas políticas deverão ser atualizadas para permitir que você realize a ação.

O erro do exemplo a seguir ocorre quando o usuário do IAM `mateojackson` tenta usar o console para visualizar detalhes sobre um atributo *`my-example-widget`* fictício, mas não tem as permissões `glacier:`*`GetWidget`* fictícias.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 glacier:GetWidget on resource: my-example-widget
```

Nesse caso, a política do usuário `mateojackson` deve ser atualizada para permitir o acesso ao recurso *`my-example-widget`* usando a ação `glacier:GetWidget`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Não estou autorizado a realizar iam: PassRole

Caso receba uma mensagem de erro informando que você não tem autorização para executar a ação `iam:PassRole`, as políticas deverão ser atualizadas para permitir a transmissão de um perfil ao Amazon Glacier.

Alguns Serviços da AWS permitem que você passe uma função existente para esse serviço em vez de criar uma nova função de serviço ou uma função vinculada ao serviço. Para fazer isso, é preciso ter permissões para passar o perfil para o serviço.

O exemplo de erro a seguir ocorre quando um usuário do IAM chamado `marymajor` tenta usar o console para executar uma ação no Amazon Glacier. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Quero permitir que pessoas de fora da minha tenham acesso Conta da AWS aos meus recursos do Amazon Glacier

É possível criar um perfil que os usuários de outras contas ou pessoas fora da organização podem usar para acessar seus recursos. É possível especificar quem é confiável para assumir o perfil. Para serviços que oferecem suporte a políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se o Amazon Glacier é compatível com esses atributos, consulte Como funciona o Amazon Glacier com o IAM.
- Para saber como fornecer acesso aos seus recursos em todos os Contas da AWS que você possui, consulte Como fornecer acesso a um usuário do IAM em outro Conta da AWS que você possui no Guia do usuário do IAM.
- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte Como fornecer acesso Contas da AWS a terceiros no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte Conceder acesso a usuários autenticados externamente (federação de identidades) no Guia do usuário do IAM.
- Para saber a diferença entre perfis e políticas baseadas em recurso para acesso entre contas, consulte Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Referência de permissões da API

Ao configurar Como funciona o Amazon Glacier com o IAM e escrever uma política de permissões que pode ser anexada a uma identidade do IAM; (políticas baseadas em identidade) ou um recurso (políticas baseadas em recurso), você pode usar a tabela a seguir como referência. A inclui cada operação da API do Amazon Glacier, as ações correspondentes para as quais você pode conceder permissões para realizar a ação e o AWS recurso para o qual você pode conceder as permissões.

Você especifica as ações no elemento `Action` da política, além do valor do recurso no elemento `Resource` da política. Além disso, você pode usar o elemento `Condition` da linguagem da política do &IAM para especificar quando uma política deve entrar em vigor.

Para especificar uma ação, use o prefixo `glacier:` seguido do nome da operação da API (por exemplo, `glacier:CreateVault`). Para a maioria das ações do Amazon Glacier, `Resource` é o cofre para o qual você deseja conceder as permissões. Você especifica um cofre como o valor `Resource` usando o ARN do cofre. Para expressar condições, você usa chaves de condição predefinidas. Para obter mais informações, consulte Políticas baseadas em recursos no Amazon Glacier.

A tabela a seguir lista as ações que podem ser usadas com políticas baseadas em identidade e recurso.

> **Note**
>
> Algumas ações podem ser usadas somente com políticas baseadas em identidade. Essas ações são marcadas por um asterisco vermelho (*) depois do nome da operação da API na primeira coluna.

API do Amazon Glacier e permissões necessárias para ações

[Abort Multipart Upload (DELETE uploadID)]

Permissões obrigatórias (ações de API): `glacier:AbortMultipartUpload`

Recursos: `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/vault-name`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/example*`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/*`

Chaves de condição do Amazon Glacier:

[Abort Vault Lock (DELETE lock-policy)]

Permissões obrigatórias (ações de API): `glacier:AbortVaultLock`

Recursos:

Chaves de condição do Amazon Glacier:

[Add Tags To Vault (POST tags add)]

Permissões obrigatórias (ações de API):`glacier:AddTagsToVault`

Recursos: `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/vault-name`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/example*`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/*`

Chaves de condição do Amazon Glacier: `glacier:ResourceTag/`*`TagKey`*

[Complete Multipart Upload (POST uploadID)]

Permissões obrigatórias (ações de API):`glacier:CompleteMultipartUpload`

Recursos: `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/vault-name`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/example*`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/*`

Chaves de condição do Amazon Glacier: `glacier:ResourceTag/`*`TagKey`*

Complete Vault Lock (POST lockId)

Permissões obrigatórias (ações de API):`glacier:CompleteVaultLock`

Recursos:

Chaves de condição do Amazon Glacier: `glacier:ResourceTag/`*`TagKey`*

Create Vault (PUT vault) *

Permissões obrigatórias (ações de API):`glacier:CreateVault`

Recursos:

Chaves de condição do Amazon Glacier:

Delete Archive (DELETE archive)

Permissões obrigatórias (ações de API):`glacier:DeleteArchive`

Recursos: `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/vault-name`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/example*`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/*`

Chaves de condição do Amazon Glacier: `glacier:ArchiveAgeInDays`, `glacier:ResourceTag/`*`TagKey`*

Delete Vault (DELETE vault)

Permissões obrigatórias (ações de API):`glacier:DeleteVault`

Recursos: `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/vault-name`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/example*`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/*`

Chaves de condição do Amazon Glacier: `glacier:ResourceTag/`*`TagKey`*

Política de acesso de exclusão do cofre (DELETE access-policy)

Permissões obrigatórias (ações de API):`glacier:DeleteVaultAccessPolicy`

Recursos: `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/vault-name`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/example*`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/*`

Chaves de condição do Amazon Glacier: `glacier:ResourceTag/`*`TagKey`*

Delete Vault Notifications (DELETE notification-configuration)

Permissões obrigatórias (ações de API):`glacier:DeleteVaultNotifications`

Recursos: `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/vault-name`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/example*`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/*`

Chaves de condição do Amazon Glacier: `glacier:ResourceTag/`*`TagKey`*

Trabalho de descrição (GET JobID)

Permissões obrigatórias (ações de API):`glacier:DescribeJob`

Recursos: `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/vault-name`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/example*`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/*`

Chaves de condição do Amazon Glacier:

Describe Vault (GET vault)

Permissões obrigatórias (ações de API):`glacier:DescribeVault`

Recursos: `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/vault-name`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/example*`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/*`

Chaves de condição do Amazon Glacier:

Política de recuperação para obter dados (GET policy) *

Permissões obrigatórias (ações de API):`glacier:GetDataRetrievalPolicy`

Recursos: `arn:aws:glacier:`*`region`*`:`*`account-id`*`:policies/retrieval-limit-policy`

Chaves de condição do Amazon Glacier:

Get Job Output (GET output)

Permissões obrigatórias (ações de API):`glacier:GetJobOutput`

Recursos: `arn:aws:glacier:region:account-id:vaults/vault-name`, `arn:aws:glacier:region:account-id:vaults/example*`, `arn:aws:glacier:region:account-id:vaults/*`

Chaves de condição do Amazon Glacier:

Get Vault Access Policy (GET access-policy)

Permissões obrigatórias (ações de API):`glacier:GetVaultAccessPolicy`

Recursos: `arn:aws:glacier:region:account-id:vaults/vault-name`, `arn:aws:glacier:region:account-id:vaults/example*`, `arn:aws:glacier:region:account-id:vaults/*`

Chaves de condição do Amazon Glacier:

Get Vault Lock (GET lock-policy)

Permissões obrigatórias (ações de API):`glacier:GetVaultLock`

Recursos: `arn:aws:glacier:region:account-id:vaults/vault-name`, `arn:aws:glacier:region:account-id:vaults/example*`, `arn:aws:glacier:region:account-id:vaults/*`

Chaves de condição do Amazon Glacier:

Get Vault Notifications (GET notification-configuration)

Permissões obrigatórias (ações de API):`glacier:GetVaultNotifications`

Recursos: `arn:aws:glacier:region:account-id:vaults/vault-name`, `arn:aws:glacier:region:account-id:vaults/example*`, `arn:aws:glacier:region:account-id:vaults/*`

Chaves de condição do Amazon Glacier:

Initiate Job (trabalhos POST)

Permissões obrigatórias (ações de API):`glacier:InitiateJob`

Recursos: `arn:aws:glacier:region:account-id:vaults/vault-name`, `arn:aws:glacier:region:account-id:vaults/example*`, `arn:aws:glacier:region:account-id:vaults/*`

Chaves de condição do Amazon Glacier: `glacier:ArchiveAgeInDays`, `glacier:ResourceTag/`*`TagKey`*

Initiate Multipart Upload (POST multipart-uploads)

Permissões obrigatórias (ações de API):`glacier:InitiateMultipartUpload`

Recursos: `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/vault-name`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/example*`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/*`

Chaves de condição do Amazon Glacier: `glacier:ResourceTag/`*`TagKey`*

Initiate Vault Lock (POST lock-policy)

Permissões obrigatórias (ações de API):`glacier:InitiateVaultLock`

Recursos:

Chaves de condição do Amazon Glacier: `glacier:ResourceTag/`*`TagKey`*

List Jobs (GET jobs)

Permissões obrigatórias (ações de API):`glacier:ListJobs`

Recursos: `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/vault-name`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/example*`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/*`

Chaves de condição do Amazon Glacier:

List Multipart Uploads (GET multipart-uploads)

Permissões obrigatórias (ações de API):`glacier:ListMultipartUploads`

Recursos: `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/vault-name`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/example*`, `arn:aws:glacier:`*`region`*`:`*`account-id`*`:vaults/*`

Chaves de condição do Amazon Glacier:

List Parts (GET uploadID)

Permissões obrigatórias (ações de API):`glacier:ListParts`

Recursos: `arn:aws:glacier:region:account-id:vaults/vault-name`, `arn:aws:glacier:region:account-id:vaults/example*`, `arn:aws:glacier:region:account-id:vaults/*`

Chaves de condição do Amazon Glacier:

Listar tags para cofre (GET tags)

Permissões obrigatórias (ações de API):`glacier:ListTagsForVault`

Recursos: `arn:aws:glacier:region:account-id:vaults/vault-name`, `arn:aws:glacier:region:account-id:vaults/example*`, `arn:aws:glacier:region:account-id:vaults/*`

Chaves de condição do Amazon Glacier:

List Vaults (GET vaults)

Permissões obrigatórias (ações de API):`glacier:ListVaults`

Recursos:

Chaves de condição do Amazon Glacier:

Remove Tags From Vault (POST tags remove)

Permissões obrigatórias (ações de API):`glacier:RemoveTagsFromVault`

Recursos: `arn:aws:glacier:region:account-id:vaults/vault-name`, `arn:aws:glacier:region:account-id:vaults/example*`, `arn:aws:glacier:region:account-id:vaults/*`

Chaves de condição do Amazon Glacier: `glacier:ResourceTag/TagKey`

Set Data Retrieval Policy (PUT policy) *

Permissões obrigatórias (ações de API):`glacier:SetDataRetrievalPolicy`

Recursos:`arn:aws:glacier:region:account-id:policies/retrieval-limit-policy`

Chaves de condição do Amazon Glacier:

Set Vault Access Policy (PUT access-policy)

Permissões obrigatórias (ações de API):`glacier:SetVaultAccessPolicy`

Recursos: `arn:aws:glacier:region:account-id:vaults/vault-name`, `arn:aws:glacier:region:account-id:vaults/example*`, `arn:aws:glacier:region:account-id:vaults/*`

Chaves de condição do Amazon Glacier: `glacier:ResourceTag/TagKey`

Definir configuração de notificação de cofre (PUT notification-configuration)

Permissões obrigatórias (ações de API):`glacier:SetVaultNotifications`

Recursos: `arn:aws:glacier:region:account-id:vaults/vault-name`, `arn:aws:glacier:region:account-id:vaults/example*`, `arn:aws:glacier:region:account-id:vaults/*`

Chaves de condição do Amazon Glacier: `glacier:ResourceTag/TagKey`

Upload Archive (POST archive)

Permissões obrigatórias (ações de API):`glacier:UploadArchive`

Recursos: `arn:aws:glacier:region:account-id:vaults/vault-name`, `arn:aws:glacier:region:account-id:vaults/example*`, `arn:aws:glacier:region:account-id:vaults/*`

Chaves de condição do Amazon Glacier: `glacier:ResourceTag/TagKey`

Upload Part (PUT uploadID)

Permissões obrigatórias (ações de API):`glacier:UploadMultipartPart`

Recursos: `arn:aws:glacier:region:account-id:vaults/vault-name`, `arn:aws:glacier:region:account-id:vaults/example*`, `arn:aws:glacier:region:account-id:vaults/*`

Chaves de condição do Amazon Glacier: `glacier:ResourceTag/TagKey`

Conectar e monitorar no Amazon Glacier

O monitoramento é uma parte importante para manter a confiabilidade, a disponibilidade e a performance do Amazon Glacier e das soluções da AWS. Você deve coletar dados de monitoramento de todas as partes de sua solução da AWS, de forma a identificar e depurar a origem da falha, caso ela ocorra. A AWS fornece as seguintes ferramentas para monitorar seus recursos do Amazon Glacier e responder aos possíveis incidentes:

Alarmes do Amazon CloudWatch

Ao usar o Amazon Glacier por meio do Amazon S3, é possível usar alarmes do Amazon CloudWatch para monitorar uma única métrica ao longo de um período que você especificará. Se a métrica ultrapassar um limite especificado, uma notificação será enviada para um tópico do Amazon SNS ou para uma política do AWS Auto Scaling. Os alarmes do CloudWatch não invocam ações só porque estão em um determinado estado. O estado deve ter sido alterado e mantido por uma quantidade especificada de períodos. Para obter mais informações, consulte Monitorar métricas com o Amazon CloudWatch.

AWS CloudTrailLogs do

O CloudTrail fornece um registro de ações executadas por um usuário, uma função ou um serviço da AWS no Amazon Glacier. O CloudTrail captura todas as chamadas de API para o Amazon Glacier, como eventos, incluindo as chamadas do console do Amazon Glacier e de chamadas de código para as APIs do Amazon Glacier. Para obter mais informações, consulte Registro de chamadas de API do Amazon Glacier com AWS CloudTrail.

AWS Trusted Advisor

O Trusted Advisor conta com as práticas recomendadas aprendidas com o atendimento a centenas de milhões de clientes da AWS. O Trusted Advisor inspeciona seu ambiente da AWS e faz recomendações quando há oportunidades para economizar dinheiro, melhorar a performance e a disponibilidade do sistema e ajuda a corrigir falhas de segurança. Todos os clientes da AWS têm acesso a cinco verificações do Trusted Advisor. Os clientes com um plano de suporte Business ou Enterprise podem ver todas as verificações do Trusted Advisor.

Consulte mais informações em AWS Trusted Advisor no Guia de Usuário Suporte.

Validação de conformidade do Amazon Glacier

Auditores terceiros avaliam a segurança e a conformidade do Amazon Glacier como parte de vários programas de conformidade da AWS, incluindo os seguintes:

- Controles do Sistema e da Organização (CSO)
- Padrão de segurança de dados do setor de cartão de pagamento (PCI DSS – Payment Card Industry Data Security Standard)
- Federal Risk and Authorization Management Program (FedRAMP)
- Health Insurance Portability and Accountability Act (HIPAA)

A AWS fornece uma lista atualizada com frequência de serviços da AWS no escopo de programas de conformidade específicos em Serviços da AWS no escopo do programa de conformidade.

Os relatórios de auditoria de terceiros estão disponíveis para download por meio do AWS Artifact. Para obter mais informações, consulte Download de relatórios no AWS Artifact no Guia do Usuário do AWS Artifact.

Para obter mais informações sobre programas de conformidade da AWS, consulte Programas de conformidade da AWS.

Sua responsabilidade de conformidade ao usar o Amazon Glacier é determinada pela confidencialidade de seus dados, pelas metas de conformidade da sua empresa e pelas regulamentações e leis aplicáveis. Caso o uso do Amazon Glacier esteja sujeito à conformidade com padrões como HIPAA, PCI ou FedRAMP, a AWS fornecerá os recursos para ajudar:

- O Vault Lock do Amazon Glacier permite implantar e aplicar facilmente controles de conformidade para cofres individuais do Amazon Glacier com uma política de bloqueio do cofre. Você pode especificar controles como "Write Once Read Many" (WORM – Uma gravação e muitas leituras) em uma política de bloqueio do cofre e bloquear a política de edições futuras. Depois que a política for bloqueada, ela não poderá mais ser alterada. As políticas de bloqueio de cofre podem ajudar você a estar em conformidade com estruturas regulatórias, como SEC17a-4 e HIPAA.
- Guias de início rápido de segurança e conformidade abordam considerações de arquitetura e etapas para implantação de ambientes de linha de base focados em conformidade e segurança na AWS.
- O whitepaper Architecting for HIPAA Security and Compliance descreve como as empresas usam a AWS para ajudá-las a atender aos requisitos da HIPAA.

- O AWS Well-Architected Tool (AWS WA Tool) é um serviço que fornece um processo consistente para avaliar e medir a arquitetura usando as práticas recomendadas da AWS. O WA Tool da AWS fornece recomendações para tornar suas cargas de trabalho mais confiáveis, seguras, eficientes e econômicas.
- Recursos de conformidade da AWS fornecem vários guias e manuais que podem se aplicar ao seu setor e local.
- AWS Config O pode ajudar você a avaliar até que ponto as configurações de recursos estão em conformidade com práticas internas e diretrizes e regulamentos do setor.
- O Security Hub da AWS fornece uma visão abrangente do seu estado de segurança na AWS que ajuda você a verificar a conformidade com os padrões e as melhores práticas do setor de segurança.

Resiliência no Amazon Glacier

A infraestrutura global AWS é criada com base em regiões e zonas de disponibilidade. AWS As regiões fornecem várias zonas de disponibilidade separadas e isoladas fisicamente, que são conectadas com baixa latência, alta taxa de transferência e redes altamente redundantes. As zonas de disponibilidade oferecem a você uma forma eficiente para criar e operar aplicativos e bancos de dados. Elas são mais altamente disponíveis, tolerantes a falhas e escaláveis que infraestruturas com único data center ou infraestruturas com vários datacenters tradicionais. O Amazon Glacier armazena dados de forma redundante em vários dispositivos abrangendo um mínimo de três zonas de disponibilidade. Para aumentar a durabilidade, o Amazon Glacier armazena seus dados de maneira sincronizada em várias zonas de disponibilidade antes de confirmar um upload bem-sucedido.

Para obter mais informações sobre regiões e zonas de disponibilidade da AWS, consulte Infraestrutura global da AWS.

Segurança da infraestrutura no Amazon Glacier

Como serviço gerenciado, o Amazon Glacier é protegido pelos procedimentos de segurança da rede global da AWS descritos em Amazon Web Services: Visão Geral dos Processos de Segurança.

O acesso ao Amazon Glacier pela rede acontece por meio de APIs publicadas pela AWS. Os clientes devem ser compatíveis com o Transport Layer Security (TLS) 1.2. Recomendamos usar o TLS 1.3 ou posterior. Os clientes também devem ter suporte a pacotes de criptografia com sigilo de encaminhamento perfeito (PFS) como Ephemeral Diffie-Hellman (DHE) ou Elliptic Curve Diffie-Hellman Encaminhamento (ECDHE). A maioria dos sistemas modernos como Java 7 e versões posteriores oferece suporte a esses modos. Além disso, as solicitações devem ser assinadas usando um ID de chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM, ou pode ser usada AWS Security Token ServiceAWS STS para gerar credenciais de segurança temporárias para assinar solicitações.

Endpoints da VPC

Um endpoint de nuvem privada virtual (VPC) permite que você conecte seu VPC de forma privada a serviços suportados AWS e serviços de endpoint da VPC alimentados por PrivateLink AWS sem a necessidade de um gateway da Internet, dispositivo NAT, conexão VPN ou conexão AWS Direct Connect. Embora o Amazon Glacier não ofereça suporte a endpoints da VPC diretamente, você pode aproveitar os endpoints da VPC do Amazon S3 se acessar o Amazon Glacier como um nível de armazenamento integrado ao Amazon S3.

Para obter mais informações sobre a configuração do ciclo de vida do Amazon S3 e a transição de objetos para a classe de armazenamento do Amazon Glacier, consulte Gerenciamento do ciclo de vida de objetos e Transição de objetos no Guia do usuário do Amazon Simple Storage Service. Para obter mais informações sobre o endpoints da VPC, consulte Endpoints da VPC no Manual do usuário da Amazon VPC.

Políticas de recuperação de dados do Amazon Glacier

Com as políticas de recuperação de dados do Amazon Glacier, você pode definir facilmente cotas de recuperação de dados e gerenciar as atividades de recuperação de dados em cada uma delas. Conta da AWS Região da AWS Para obter mais informações sobre cobranças pela recuperação de dados do Amazon Glacier, consulte Definição de preço do Amazon Glacier.

> **Important**
>
> Uma política de recuperação de dados se aplica somente a recuperações Padrão e gerencia solicitações de recuperação feitas diretamente ao Amazon Glacier.
> Para obter mais informações sobre as classes de armazenamento do Amazon Glacier, consulte Classes de armazenamento para arquivamento de objetos e Transição de objetos no Guia do usuário do Amazon Simple Storage Service.

Tópicos

- Escolher uma política de recuperação de dados do Amazon Glacier
- Usar o console do Amazon Glacier para configurar uma política de recuperação de dados
- Usar a API do Amazon Glacier para configurar uma política de recuperação de dados

Escolher uma política de recuperação de dados do Amazon Glacier

É possível escolher entre três tipos de política de recuperação de dados do Amazon Glacier: Sem limite de recuperação, Somente nível gratuito e Taxa máxima de recuperação.

Sem limite de recuperação é a política de recuperação de dados padrão que é usada para recuperações. Se você usar a política Sem limite de recuperação, nenhuma cota de recuperação será definida, e todas as solicitações válidas de recuperação de dados serão aceitas.

Ao usar uma política somente de nível gratuito, você pode manter suas recuperações dentro do limite diário de nível AWS gratuito e não incorrer em nenhum custo de recuperação de dados. Se você quiser recuperar mais dados do que o seu limite de nível AWS gratuito, você pode usar uma política de taxa máxima de recuperação para definir uma bytes-per-hour cota de taxa de recuperação. A política de Taxa Máxima de Recuperação garante que a taxa máxima de recuperação de todos os

trabalhos de recuperação em sua conta Região da AWS não exceda a bytes-per-hour cota que você definiu.

Com as políticas Somente nível gratuito e Taxa máxima de recuperação, as solicitações de recuperação de dados que excedam as cotas de recuperação não são aceitas. Se você usar uma política Somente nível gratuito, o Amazon Glacier vai rejeitar de maneira síncrona solicitações de recuperação que excederiam a franquia de nível gratuito da AWS . Se você usa uma política de taxa máxima de recuperação, o Amazon Glacier rejeita solicitações de recuperação que fazem com que a taxa máxima de recuperação dos trabalhos em andamento bytes-per-hour exceda a cota definida pela política. Essas políticas ajudam a simplificar o gerenciamento de custos da recuperação de dados.

Estes são alguns fatos úteis sobre políticas de recuperação de dados:

- As configurações da recuperação de dados não alteram o período de 3 a 5 horas utilizado para recuperar dados do Amazon Glacier usando recuperações padrão.
- Definir uma nova política de recuperação de dados não afeta os trabalhos de recuperação aceitos anteriormente já em andamento.
- Se uma solicitação do trabalho de recuperação é rejeitada por causa de uma política de recuperação de dados, você não é cobrado pelo trabalho ou pela solicitação.
- Você pode definir uma política de recuperação de dados para cada um Região da AWS, que governará todas as atividades de recuperação de dados na sua conta Região da AWS . Uma política de recuperação de dados é específica para um determinado setor Região da AWS porque os custos de recuperação de dados variam de acordo com o outro. Regiões da AWS Para obter mais informações, consulte Definição de preço do Amazon Glacier.

Política Somente nível gratuito

Você pode definir uma política de recuperação de dados como somente de nível gratuito para garantir que suas recuperações sempre permaneçam dentro do limite de nível AWS gratuito, para que você não incorra em cobranças de recuperação de dados. Se uma solicitação de recuperação é rejeitada, você recebe uma mensagem de erro informando que a solicitação foi negada pela política de recuperação de dados atual.

Você pode definir a política de recuperação de dados como Somente nível gratuito por região. Depois que a política for definida, você não poderá recuperar mais dados em um dia do que o limite

diário proporcional de recuperação do nível gratuito da AWS para essa Região da AWS. Você também não incorre em taxas de recuperação de dados.

Você também poderá alternar para uma política Somente nível gratuito depois de ter incorrido em cobranças pela recuperação de dados em um mês. Neste caso, a política Somente nível gratuito entra em vigor para novas solicitações de recuperação, mas não afeta as solicitações anteriores. Você será cobrado pelos encargos incorridos anteriormente.

Política Max Retrieval Rate

Você pode definir sua política de recuperação de dados como Taxa máxima de recuperação para controlar a taxa máxima de recuperação especificando uma cota de recuperação de dados que tenha um máximo. bytes-per-hour Quando você define a política de recuperação de dados como Taxa máxima de recuperação, uma nova solicitação de recuperação é rejeitada se fizer com que a taxa máxima de recuperação dos trabalhos em andamento exceda a bytes-per-hour cota especificada pela política. Se uma solicitação de trabalho é rejeitada, você recebe uma mensagem de erro informando que a solicitação foi negada pela política de recuperação de dados atual.

Definir sua política de recuperação de dados de acordo com a política de Taxa Máxima de Recuperação pode afetar o quanto da sua franquia de Nível AWS Gratuito você pode usar em um dia. Por exemplo, suponhamos que você defina Max Retrieval Rate como 1 MB por hora. Isso é menor do que a taxa da apólice de nível AWS gratuito. Para garantir que você faça bom uso do subsídio diário de nível AWS gratuito, primeiro defina sua política como somente de nível gratuito e, em seguida, alterne para a política de taxa máxima de recuperação posteriormente, se necessário. Para obter mais informações sobre como seu subsídio de recuperação é calculado, acesse Amazon Glacier. FAQs

Política Sem limite de recuperação

Se a política de recuperação de dados estiver definida como Sem limite de recuperação, todas as solicitações de recuperação de dados válidos serão aceitas, e os custos de recuperação de dados vão variar com base no uso.

Usar o console do Amazon Glacier para configurar uma política de recuperação de dados

Usar o console do Amazon Glacier para criar uma política de recuperação de dados

1. Faça login Console de gerenciamento da AWS e abra o console do Amazon Glacier em casahttps://console.aws.amazon.com/glacier/.
2. Em Selecionar uma região, escolha uma no Região da AWS menu suspenso. Você pode configurar uma política de recuperação de dados para cada um Região da AWS.
3. No painel de navegação à esquerda, escolha Configurações de recuperação de dados.
4. Escolha Editar. A página Editar políticas de recuperação de dados é exibida.
5. Em Políticas de recuperação de dados, escolha uma política.

   Você pode selecionar uma das três políticas de recuperação de dados: Sem limite de recuperação, Somente nível gratuito ou Especificar uma taxa máxima de recuperação.

   - Se você escolher Sem limite de recuperação, todas as solicitações de recuperação de dados válidas serão aceitas.
   - Se você escolher somente o nível gratuito, as solicitações de recuperação de dados que excederem o nível AWS gratuito não serão aceitas.
   - Se você escolher Especificar uma taxa máxima de recuperação, as solicitações de recuperação de dados serão rejeitadas se fizerem com que a taxa máxima de recuperação dos trabalhos em andamento exceda a taxa máxima de recuperação especificada. Você deve especificar um valor de gigabytes (GB) por hora na caixa GB/hora em Taxa máxima de recuperação. Ao inserir um valor para GB/hora, o console calcula um custo estimado para você.
6. Escolha Salvar alterações.

Usar a API do Amazon Glacier para configurar uma política de recuperação de dados

Você pode visualizar e definir uma política de recuperação de dados usando a API REST do Amazon Glacier ou usando o. AWS SDKs

Usar a API REST do Amazon Glacier para configurar uma política de recuperação de dados

Você pode visualizar e definir uma política de recuperação de dados usando a API REST do Amazon Glacier. Você pode visualizar uma política de recuperação de dados existente usando a operação Política de recuperação para obter dados (GET policy). Você define uma política de recuperação de dados usando a operação Set Data Retrieval Policy (PUT policy).

Ao usar a operação de política PUT, selecione o tipo da política de recuperação de dados definindo o valor do campo `Strategy` JSON como `BytesPerHour`, `FreeTier` ou `None`. `BytesPerHour` é equivalente a escolher Especificar uma taxa máxima de recuperação no console, `FreeTier` a escolher Somente nível gratuito e `None` a selecionar Sem política de recuperação.

Quando você usar a operação Initiate Job (trabalhos POST) para iniciar um trabalho de recuperação de dados que exceder a taxa de recuperação máxima definida na política de recuperação de dados, a operação `Initiate Job` é interrompida e lança uma exceção.

Usando o AWS SDKs para configurar uma política de recuperação de dados

AWS permite SDKs que você desenvolva aplicativos para o Amazon Glacier. Eles SDKs fornecem bibliotecas que são mapeadas para a API REST subjacente e fornecem objetos que permitem criar solicitações e processar respostas com facilidade. Para obter mais informações, consulte Usando o AWS SDKs com o Amazon Glacier.

Marcação de recursos do Amazon Glacier

Uma tag é um rótulo atribuído a um recurso . Cada tag consiste em uma chave e um valor, ambos definidos por você. Você pode atribuir as tags definidas aos recursos do cofre do Amazon Glacier. O uso de tags é uma forma simples, mas eficiente, de gerenciar os recursos da AWS e organizar os dados, incluindo dados de faturamento.

Tópicos

- Conceitos básicos da marcação
- Restrições de tag
- Monitoramento de custos com marcação
- Gerenciar controle de acesso com marcação
- Seções relacionadas

Conceitos básicos da marcação

Você usa o console do Amazon Glacier, AWS Command Line Interface (AWS CLI), ou a API do Amazon Glacier para concluir as seguintes tarefas:

- Adicionar tags a um cofre
- Listar as tags de um cofre
- Remover tags de um cofre

Para obter informações sobre como adicionar, listar e remover tags, consulte Marcar seus Cofres do Amazon Glacier.

Você pode usar tags para categorizar os cofres. Por exemplo, você pode categorizar cofres por finalidade, proprietário ou ambiente. Como você define a chave e o valor para cada marca, é possível criar um conjunto de categorias personalizado para atender às suas necessidades específicas. Por exemplo, convém definir um conjunto de tags que ajude a acompanhar cofres por proprietário e finalidade para o cofre. Estes são alguns exemplos de tags:

- Proprietário: nome
- Finalidade: arquivos de vídeo

- Ambiente: produção

Restrições de tag

As restrições de tag básicas são as seguintes:

- O número máximo de tags de um recurso (cofre) é 50.
- As chaves e os valores de tags diferenciam maiúsculas de minúsculas.

As restrições da chave de tag são as seguintes:

- Dentro de um conjunto de tags de um cofre, cada chave de tag deve ser exclusiva. Se uma tag for adicionada com uma chave que já estiver em uso, a nova tag substituirá o par de chave-valor existente.
- Não é possível iniciar uma chave de tag com `aws:`, pois esse prefixo é reservado para uso pela AWS. A AWS cria tags que começam com esse prefixo em seu nome, mas você não pode editá-las ou excluí-las.
- As chaves de tag devem ter entre 1 e 128 caracteres Unicode.
- As chaves de tag devem consistir nos seguintes caracteres: letras Unicode, dígitos, espaços e os seguintes caracteres especiais: `_ . / = + - @`.

As restrições de valor da tag são as seguintes:

- Os valores de tag devem ter entre 0 e 255 caracteres Unicode.
- Os valores das tags podem estar em branco. Caso contrário, elas devem conter os seguintes caracteres: letras Unicode, dígitos, espaço em branco e qualquer um dos seguintes caracteres especiais: `_ . / = + - @`.

Monitoramento de custos com marcação

É possível usar tags para categorizar e monitorar seus custos da AWS. Quando você aplica tags a qualquer recurso AWS, inclusive cofres, o AWS relatório de alocação de custos inclui o uso e os custos agregados por tags. É possível aplicar tags que representem categorias de negócios (como centros de custos, nomes de aplicativo e proprietários) para organizar os custos entre vários

serviços. Para obter mais informações, consulte Usar tags de alocação de custos para relatórios de faturamento personalizados no Manual do usuário do AWS Billing.

Gerenciar controle de acesso com marcação

Você pode usar tags como uma condição em uma declaração de política de acesso. Por exemplo, você pode configurar uma tag hold legal e incluí-la como uma condição em uma política de retenção de dados que afirma que "arquivar a exclusão de todos será negado se o valor da tag hold legal estiver definido como ". `True`.” Você pode implantar a política de retenção de dados e definir a tag hold legal como `False` em condições normais. Se os dados precisarem ser colocados em espera para auxiliar uma investigação, você poderá ativar facilmente a retenção legal definindo o valor de tag como `True` e removendo a retenção de uma maneira semelhante depois. Para obter mais informações, consulte Controlar o acesso usando tags no Guia do usuário do IAM.

Seções relacionadas

- Marcar seus Cofres do Amazon Glacier

Registro de chamadas de API do Amazon Glacier com AWS CloudTrail

O Amazon Glacier (Amazon Glacier) está AWS CloudTrail integrado a um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS serviço no Amazon Glacier. CloudTrail captura todas as chamadas de API para o Amazon Glacier como eventos, incluindo chamadas do console do Amazon Glacier e de chamadas de código para o Amazon Glacier. APIs Se você criar uma trilha, poderá permitir a entrega contínua de CloudTrail eventos para um bucket do Amazon S3, incluindo eventos para o Amazon Glacier. Se não configurar uma trilha, você ainda poderá visualizar os eventos mais recentes no console do CloudTrail em Event history. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita ao Amazon Glacier, o endereço IP a partir do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para saber mais sobre isso CloudTrail, consulte o Guia AWS CloudTrail do usuário.

Informações sobre o Amazon Glacier em CloudTrail

CloudTrail é ativado no seu Conta da AWS quando você cria a conta. Quando a atividade ocorre no Amazon Glacier, essa atividade é registrada em um CloudTrail evento junto com outros eventos de AWS serviço no histórico de eventos. Você pode visualizar, pesquisar e baixar eventos recentes no seu Conta da AWS. Para obter mais informações, consulte Visualização de eventos com histórico de CloudTrail eventos.

Para obter um registro contínuo dos eventos em seu Conta da AWS, incluindo eventos para o Amazon Glacier, crie uma trilha. Uma trilha permite CloudTrail entregar arquivos de log para um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, a trilha se aplica a todas as AWS regiões. A trilha registra eventos de todas as AWS regiões na AWS partição e entrega os arquivos de log ao bucket do Amazon S3 que você especificar. Além disso, você pode configurar outros AWS serviços para analisar e agir com base nos dados de eventos coletados nos CloudTrail registros. Para obter mais informações, consulte:

- Visão geral da criação de uma trilha
- CloudTrail Serviços e integrações compatíveis
- Configurando notificações do Amazon SNS para CloudTrail

- Recebendo arquivos de CloudTrail log de várias regiões e recebendo arquivos de CloudTrail log de várias contas

Todas as ações do Amazon Glacier são registradas CloudTrail e documentadas no. Referência de API para Amazon Glacier Por exemplo, chamadas para as List Vaults (GET vaults) ações Create Vault (PUT vault)Delete Vault (DELETE vault), e geram entradas nos arquivos de CloudTrail log.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar o seguinte:

- Se a solicitação foi feita com credenciais de usuário raiz ou com outras credenciais.
- Se a solicitação foi feita com credenciais de segurança temporárias de um perfil ou de um usuário federado.
- Se a solicitação foi feita por outro AWS serviço.

Para obter mais informações, consulte Elemento userIdentity do CloudTrail .

Noções básicas de entradas de arquivos de log do Amazon Glacier

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log para um bucket do Amazon S3 que você especificar. CloudTrail os arquivos de log contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros da solicitação e assim por diante. CloudTrail os arquivos de log não são um rastreamento de pilha ordenado das chamadas públicas de API, portanto, eles não aparecem em nenhuma ordem específica.

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra as Describe Vault (GET vault) ações Create Vault (PUT vault) Delete Vault (DELETE vault)List Vaults (GET vaults),, e.


```
{
    "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"
            }
        }
    ]
}
```

Referência de API para Amazon Glacier

O Amazon Glacier dá suporte a um conjunto de operações – mais especificamente, um conjunto de chamadas de API RESTful – que permitem interagir com o serviço.

Você pode usar qualquer biblioteca de programação capaz de enviar solicitações HTTP objetivando enviar as solicitações REST ao Amazon Glacier. Ao enviar uma solicitação REST, o Amazon Glacier exige que você autentique toda solicitação com uma assinatura. Além disso, ao fazer o upload de um arquivo, você também deve computar a soma de verificação da carga útil e incluí-la na solicitação. Para obter mais informações, consulte Solicitações de assinatura.

Se ocorrer algum erro, você precisará saber o que o Amazon Glacier enviará em uma resposta de erro para que seja possível processá-lo. Esta seção fornece todas essas informações, além de documentar as operações REST, de maneira que você possa fazer chamadas de API REST diretamente.

Você pode usar as chamadas de API REST diretamente ou usar os SDKs da Amazon que fornecem bibliotecas wrapper para simplificar a tarefa de codificação. Essas bibliotecas assinam cada solicitação enviada por você e calculam a soma de verificação da carga útil na solicitação. Por isso, usar os SDKs da Amazon simplifica a tarefa de codificação. Este guia do desenvolvedor apresenta exemplos funcionais de operações do Amazon Glacier básicas usando o AWS SDK para Java e o .NET. Para obter mais informações, consulte, Usando o AWS SDKs com o Amazon Glacier.

Tópicos

- Cabeçalhos de solicitação comuns
- Cabeçalhos de resposta comuns
- Solicitações de assinatura
- Computar somas de verificação
- Respostas de erro
- Operações de cofre
- Operações de arquivo
- Operações de multipart upload
- Operações de trabalho
- Tipos de dados usados em operações de trabalho

- Operações de recuperação de dados

Cabeçalhos de solicitação comuns

Entre as solicitações REST do Amazon Glacier estão cabeçalhos que contêm informações básicas sobre a solicitação. A tabela a seguir descreve os cabeçalhos que podem ser usados por todas as solicitações REST do Amazon Glacier.

Nome do cabeçalho	Descrição	Obrigatório
`Authorization`	O cabeçalho obrigatório para assinar solicitações. O Amazon Glacier requer o Signature versão 4. Para obter mais informações, consulte Solicitações de assinatura. Tipo: string	Sim
`Content-Length`	O tamanho do corpo da solicitação (sem os cabeçalhos). Tipo: string Condição: obrigatório somente para a API Upload Archive (POST archive).	Condicional
`Date`	A data que pode ser usada para criar a assinatura contida no cabeçalho `Authorization` . Se o cabeçalho `Date` precisar ser usado na assinatura, ele deverá ser especificado no formato básico ISO 8601. Neste caso, o cabeçalho `x-amz-date` não é necessário. Observe que quando `x-amz-date` está presente, ele sempre substitui o valor do cabeçalho `Date`. Se o cabeçalho Data não for usado na assinatura, ele poderá ser um dos formatos de data completa especificados por RFC 2616, seção 3.3. Por exemplo, a seguinte data/hora `Wed, 10 Feb`	Condicional

Nome do cabeçalho	Descrição	Obrigatório
	`2017 12:00:00 GMT` é um cabeçalho de data/hora válido para usar com o Amazon Glacier. Se você estiver usando o cabeçalho `Date` para assinatura, ele deverá estar no formato `YYYYMMDD'T'HHMMSS'Z'` básico ISO 8601. Tipo: string Condição: se `Date` for especificado, mas não estiver em formato básico ISO 8601, você também deverá incluir o cabeçalho `x-amz-date` . Se `Date` for especificado em formato básico ISO 8601, isso será suficiente para assinar solicitações e você não precisar do cabeçalho `x-amz-date` . Para obter mais informações, consulte Tratamento de datas no Signature versão 4 no Glossário da Amazon Web Services.	
`Host`	Este cabeçalho especifica o endpoint de serviço para o qual você envia as solicitações. O valor deve ser da forma “`glacier.`*`region`*`.amazonaws.com` ” em que *`region`* (região) é substituída por uma designação de região da AWS, como `us-west-2` . Tipo: string	Sim

Nome do cabeçalho	Descrição	Obrigatório
`x-amz-content-sha256`	A soma de verificação SHA256 computada de toda a carga útil cujo upload é feito com Upload Archive (POST archive) ou Upload Part (PUT uploadID) . Esse cabeçalho não é igual ao cabeçalho `x-amz-sha256-tree-hash` , embora, para algumas pequenas cargas úteis, os valores sejam os mesmos. Quando `x-amz-content-sha256` é obrigatório, `x-amz-content-sha256` e `x-amz-sha256-tree-hash` devem ser especificados. Tipo: string Condição: obrigatório para API de streaming, Upload Archive (POST archive) e Upload Part (PUT uploadID).	Condicional
`x-amz-date`	A data usada para criar a assinatura no cabeçalho de autorização. O formato deve ser ISO 8601 básico no formato `YYYYMMDD'T'HHMMSS'Z'` . Por exemplo, a data/hora `20170210T120000Z` a seguir é um `x-amz-date` válido a ser usado com o Amazon Glacier. Tipo: string Condição: `x-amz-date` é opcional para todas as solicitações; ele pode ser usado para substituir a data usada em solicitações de assinatura. Se o cabeçalho `Date` for especificado no formato básico ISO 8601, `x-amz-date` não será necessário. Quando `x-amz-date` está presente, ele sempre substitui o valor do cabeçalho `Date`. Para obter mais informações, consulte Tratamento de datas no Signature versão 4 no Glossário da Amazon Web Services.	Condicional

Nome do cabeçalho	Descrição	Obrigatório
`x-amz-glacier-version`	A versão da API do Amazon Glacier a ser usada. A versão atual é `2012-06-01` . Tipo: string	Sim
`x-amz-sha256-tree-hash`	A soma de verificação do hash de árvore SHA256 para um arquivo cujo upload foi feito (Upload Archive (POST archive)) ou uma parte do arquivo (Upload Part (PUT uploadID)). Para obter mais informações sobre como calcular essa soma de verificação, consulte Computar somas de verificação. Tipo: string Padrão: nenhum Condição: obrigatório para Upload Archive (POST archive) e Upload Part (PUT uploadID).	Condicional

Cabeçalhos de resposta comuns

A tabela a seguir descreve os cabeçalhos de resposta comuns à maioria das respostas de API.

Nome	Descrição
`Content-Length`	O tamanho em bytes do corpo da resposta. Tipo: string
`Date`	A data e a hora em que o Amazon Glacier respondeu, por exemplo, `Wed, 10 Feb 2017 12:00:00 GMT`. O formato da data deve ser um dos formatos de data completos especificados por RFC 2616, seção 3.3. Observe que a `Date` retornada pode variar um pouco em relação a outras datas. Dessa forma, por exemplo, a data retornada de uma solicitação

Nome	Descrição
	Upload Archive (POST archive) talvez não corresponda à data mostrada para o arquivo em uma lista de inventários para o cofre. Tipo: string
`x-amzn-RequestId`	Um valor criado pelo Amazon Glacier que identifica com exclusividade a solicitação. Se você tiver um problema com o Amazon Glacier, a AWS poderá usar esse valor para solucioná-lo. É recomendável registrar em log esses valores. Tipo: string
`x-amz-sha256-tree-hash`	A soma de verificação do hash de árvore SHA256 do arquivo ou o corpo do inventário. Para obter mais informações sobre como calcular essa soma de verificação, consulte Computar somas de verificação. Tipo: string

Solicitações de assinatura

No Amazon Glacier toda solicitação enviada é autenticada com uma assinatura. Para assinar uma solicitação, calcule uma assinatura digital usando a função de hash criptográfico. Hash criptográfico é uma função que retorna um valor de hash exclusivo com base na entrada. A entrada da função de hash inclui o texto da solicitação e a chave de acesso secreta. A função de hash retorna um valor de hash que você inclui na solicitação como sua assinatura. A assinatura é parte do cabeçalho `Authorization` de sua solicitação.

Depois de receber a solicitação, o Amazon Glacier recalculará a assinatura usando a mesma função de hash e a entrada que você usou para assinar a solicitação. Quando a assinatura resultante corresponde à assinatura na solicitação, o Amazon Glacier processa a solicitação. Do contrário, a solicitação é rejeitada.

O Amazon Glacier oferece suporte à autenticação usando o Signature versão 4 da AWS. O processo para calcular uma assinatura pode ser dividido em três tarefas:

- Tarefa 1: Criar uma solicitação canônica

Reorganize sua solicitação HTTP em um formato canônico. É necessário usar uma forma canônica, pois o Amazon Glacier usa a mesma forma canônica quando recalcula uma assinatura para comparar com a que você enviou.

- Tarefa 2: Criar uma string para assinar

  Crie uma string que será usada como um dos valores de entrada para sua função hash criptográfica. A string, chamada string-to-sign, é uma concatenação do nome do algoritmo hash, da data da solicitação, de uma string do escopo da credencial e da solicitação canonizada da tarefa anterior. A cadeia de caracteres do escopo da credencial em si é uma concatenação de informações de data, AWS região e serviço.

- Tarefa 3: Criar uma assinatura

  Crie uma assinatura para sua solicitação usando uma função hash criptográfica que aceita duas strings de entrada: sua string para assinar e uma chave derivada. A chave derivada é calculada começando com sua chave de acesso secreta e usando a string do escopo da credencial para criar uma série de códigos de autenticação de mensagens baseados em hash ()HMACs. Observe que a função hash usada nessa etapa de assinatura não é o algoritmo de hash de árvore usado no Amazon Glacier que carrega dados. APIs

Tópicos

- Cálculo de assinatura de exemplo
- Calcular assinaturas para as operações de streaming

Cálculo de assinatura de exemplo

O exemplo a seguir mostra os detalhes da criação de uma assinatura para Create Vault (PUT vault). Esse exemplo pode ser usado como referência para verificar o método de cálculo da assinatura. Para obter mais informações, consulte Solicitações de AWS API de assinatura no Guia do usuário do IAM.

O exemplo supõe o seguinte:

- O timestamp da solicitação é `Fri, 25 May 2012 00:24:53 GMT`.
- O endpoint é a região Leste dos EUA (Norte da Virgínia) `us-east-1`.

A sintaxe de solicitação geral (incluindo o corpo JSON) é:

```
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
```

O formato canônico da solicitação calculada para Tarefa 1: Crie uma solicitação canônica é:

```
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
```

A última linha da solicitação canônica é o hash do corpo da solicitação. Além disso, observe a terceira linha vazia na solicitação canônica. Isso porque não há parâmetros de consulta para essa API.

A string para assinar para a Tarefa 2: Criar uma string para assinar é:

```
AWS4-HMAC-SHA256
20120525T002453Z
20120525/us-east-1/glacier/aws4_request
5f1da1a2d0feb614dd03d71e87928b8e449ac87614479332aced3a701f916743
```

A primeira linha da string-to-sign é o algoritmo, a segunda linha é a time stamp, a terceira linha é o escopo da credencial e a última linha é o hash da solicitação canônica da Tarefa 1: Crie uma solicitação canônica. O nome do serviço a usar no escopo da credencial é `glacier`.

Para a Tarefa 3: Criar uma assinatura, a chave derivada pode ser representada como:

```
derived key = HMAC(HMAC(HMAC(HMAC("AWS4" + YourSecretAccessKey,"20120525"),"us-
east-1"),"glacier"),"aws4_request")
```

Se a chave de acesso secreta, `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`, for usada, a assinatura calculada será:

```
3ce5b2f2fffac9262b4da9256f8d086b4aaf42eba5f111c21681a65a127b7c2a
```

A etapa final é construir o cabeçalho `Authorization`. Para a chave de acesso de demonstração `AKIAIOSFODNN7EXAMPLE`, o cabeçalho (com quebras de linha adicionadas por motivo de legibilidade) é:

```
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20120525/us-east-1/
glacier/aws4_request,
SignedHeaders=host;x-amz-date;x-amz-glacier-version,
Signature=3ce5b2f2fffac9262b4da9256f8d086b4aaf42eba5f111c21681a65a127b7c2a
```

Calcular assinaturas para as operações de streaming

Upload Archive (POST archive) e Upload Part (PUT uploadID) são operações de streaming que exigem que você inclua um cabeçalho adicional `x-amz-content-sha256` ao assinar e enviar a solicitação. As etapas de assinatura das operações de streaming são exatamente as mesmas de outras operações, com a adição do cabeçalho de streaming.

O cálculo do cabeçalho de streaming `x-amz-content-sha256` é baseado no SHA256 hash de todo o conteúdo (carga útil) que deve ser carregado. Observe que esse cálculo é diferente do hash (Computar somas de verificação) da SHA256 árvore. Além dos casos triviais, o valor de hash SHA 256 dos dados da carga útil será diferente do hash da SHA256 árvore dos dados da carga útil.

Se os dados da carga útil forem especificados como uma matriz de bytes, você poderá usar o seguinte trecho de código Java para calcular o 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();
}
```

Da mesma forma, em C#, você pode calcular o SHA256 hash dos dados da carga, conforme mostrado no trecho de código a seguir.

```
public static byte[] CalculateSHA256Hash(byte[] payload)
{
    SHA256 sha256 = System.Security.Cryptography.SHA256.Create();
    byte[] hash = sha256.ComputeHash(payload);

    return hash;
}
```

Cálculo da assinatura de exemplo para Streaming API

O exemplo a seguir mostra os detalhes da criação de uma assinatura para Upload Archive (POST archive) um dos dois streaming APIs no Amazon Glacier. O exemplo supõe o seguinte:

- O timestamp da solicitação é `Mon, 07 May 2012 00:00:00 GMT`.
- O endpoint é a região Leste dos EUA (Norte da Virgínia), eu-east-1.
- O conteúdo da carga útil é uma string "Bem-vindo ao Amazon Glacier."

A sintaxe da solicitação geral (inclusive o corpo JSON) é mostrada no exemplo abaixo. O cabeçalho `x-amz-content-sha256` está incluído. Neste exemplo simplificado, `x-amz-sha256-tree-hash` e `x-amz-content-sha256` têm o mesmo valor. No entanto, para uploads de arquivos maiores que 1 MB, este não é o caso.

```
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
```

A forma canônica da solicitação calculada para Tarefa 1: Criar uma solicitação canônica é mostrada abaixo. O cabeçalho de streaming `x-amz-content-sha256` é incluído com o valor. Isso significa que você deve ler a carga útil e calcular o SHA256 hash primeiro e depois calcular a assinatura.

```
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
```

O restante do cálculo da assinatura segue as etapas descritas em Cálculo de assinatura de exemplo. O cabeçalho `Authorization` que usa a chave de acesso secreta `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY` e a chave de acesso `AKIAIOSFODNN7EXAMPLE` é mostrado abaixo (com quebras de linha adicionadas para fins de legibilidade):

```
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
```

Computar somas de verificação

Ao fazer upload de um arquivo, você deve incluir os cabeçalhos `x-amz-sha256-tree-hash` e `x-amz-content-sha256`. O cabeçalho `x-amz-sha256-tree-hash` é uma soma de verificação da carga útil no corpo de solicitação. Este tópico descreve como calcular o cabeçalho `x-amz-sha256-tree-hash`. O cabeçalho `x-amz-content-sha256` é um hash de toda a carga útil e é obrigatório para a autorização. Para obter mais informações, consulte Cálculo da assinatura de exemplo para Streaming API.

A carga útil da solicitação pode ser:

- Todo o arquivo— Ao carregar um arquivo em uma única solicitação usando a Upload Archive API, você envia todo o arquivo no corpo da solicitação. Nesse caso, você deve incluir a soma de verificação de todo o arquivo.
- Parte do arquivo— Ao carregar um arquivo em partes usando a multipart upload API, você envia somente uma parte do arquivo no corpo da solicitação. Nesse caso, você inclui a soma de verificação da parte do arquivo. E, depois de fazer upload de todas as partes, você enviará uma solicitação Complete Multipart Upload, que deve incluir a soma de verificação de todo o arquivo.

A soma de verificação da carga útil é um hash de árvore SHA-256. Ele se chama hash de árvore porque, no processo de computação da soma de verificação, você computa uma árvore de valores de hash SHA-256. O valor do hash na raiz é a soma de verificação de todo o arquivo.

> **Note**
>
> Esta seção descreve uma maneira de computar o hash de árvore SHA-256. Porém, você pode usar qualquer procedimento, desde que ele produza o mesmo resultado.

Você computa o hash de árvore SHA-256 da seguinte maneira:

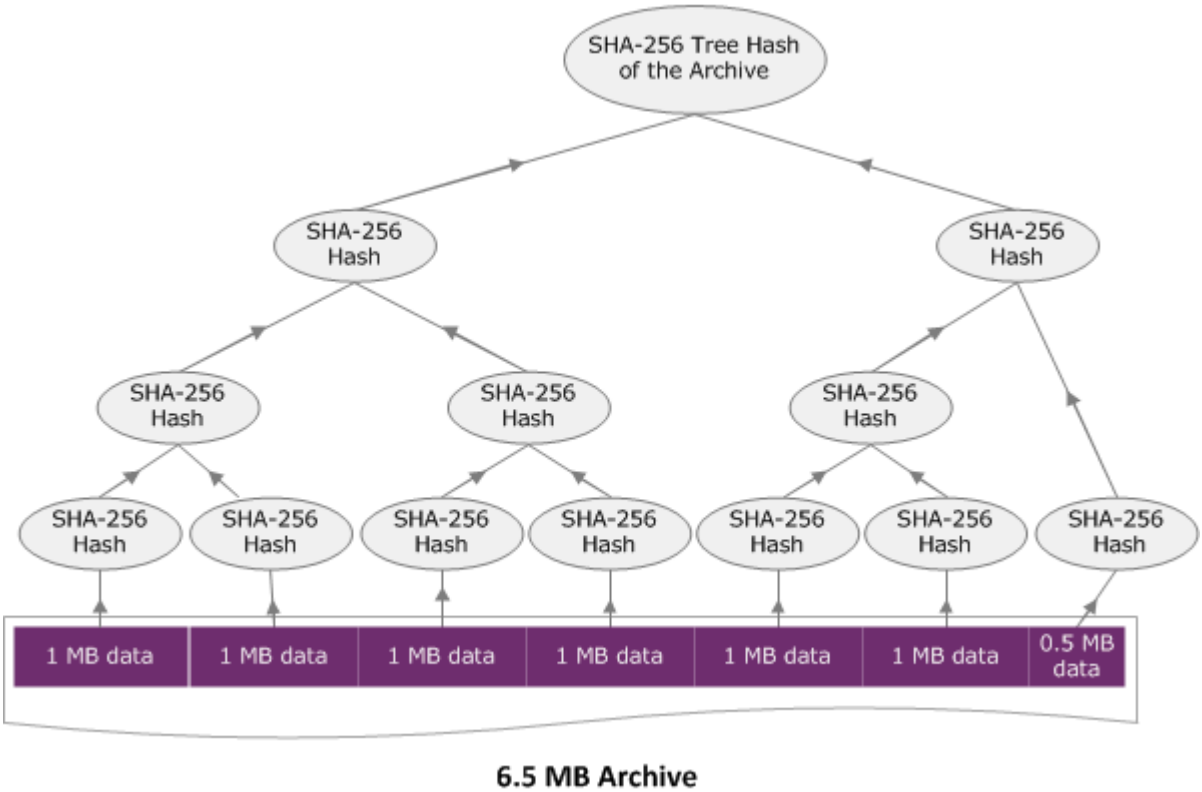
1. Para cada bloco de 1 MB de dados de carga útil, compute o hash SHA-256. O último bloco de dados pode ser menor que 1 MB. Por exemplo, se estiver fazendo upload de um arquivo de 3,2 MB, você computará os valores de hash SHA-256 para cada um dos três primeiros blocos de 1 MB de dados e, em seguida, computará o hash SHA-256 do 0,2 MB de dados restantes. Esses valores de hash formam os nós folha da árvore.
2. Compile o próximo nível da árvore.
   a. Concatene dois valores de hash do nó filho consecutivos e compute o hash SHA-256 dos valores de hash concatenados. Essa concatenação e a geração do hash SHA-256 produzem um nó pai para os dois nós filho.
   b. Quando restar somente um nó filho, promova esse valor de hash para o próximo nível na árvore.
3. Repita a etapa 2 até a árvore resultante ter uma raiz. A raiz da árvore fornece um hash de todo o arquivo, e uma raiz da subárvore apropriada fornece o hash para a parte em um multipart upload.

Tópicos

- Exemplo do hash de árvore 1: fazer upload de um arquivo em uma única solicitação
- Exemplo do hash de árvore 2: fazer upload de um arquivo usando um multipart upload
- Computar o hash de árvore de um arquivo
- Receber somas de verificação durante o download de dados

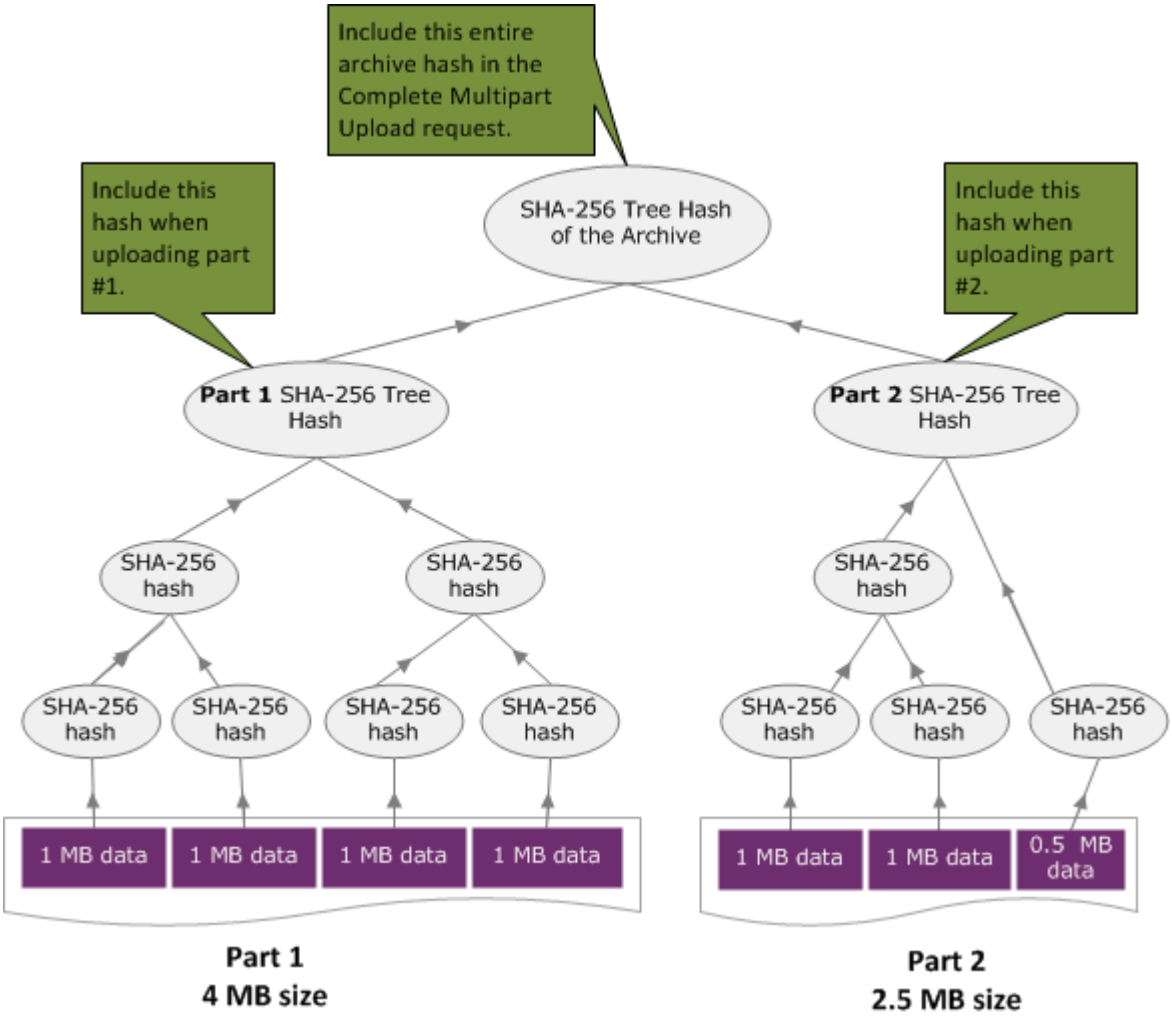
Exemplo do hash de árvore 1: fazer upload de um arquivo em uma única solicitação

Quando você faz upload de um arquivo em uma única solicitação usando a Upload Archive API (consulte Upload Archive (POST archive)), a carga útil da solicitação inclui todo o arquivo. Dessa forma, você deve incluir o hash de árvore de todo o arquivo no cabeçalho de solicitação `x-amz-sha256-tree-hash`. Suponhamos que você queira fazer upload de um arquivo de 6,5 MB. O diagrama a seguir ilustra o processo de criação do hash SHA-256 do arquivo. Você lê o arquivo e computa o hash SHA-256 de cada bloco de 1 MB. Você também computa o hash do 0,5 MB de dados restante e, em seguida, compila a árvore conforme descrito no procedimento anterior.



Exemplo do hash de árvore 2: fazer upload de um arquivo usando um multipart upload

O processo de computar o hash de árvore durante o upload de um arquivo usando-se multipart upload é o mesmo do upload do arquivo em uma única solicitação. A única diferença é que, em um multipart upload, você faz upload somente de uma parte do arquivo em cada solicitação (usando a API Upload Part (PUT uploadID)) e, assim, fornece a soma de verificação somente da parte no cabeçalho da solicitação `x-amz-sha256-tree-hash`. No entanto, depois de fazer upload de todas as partes, você deverá enviar a solicitação Complete Multipart Upload (consulte Complete Multipart Upload (POST uploadID)) com um hash de árvore de todo o arquivo no cabeçalho da solicitação `x-amz-sha256-tree-hash`.



Computar o hash de árvore de um arquivo

Os algoritmos mostrados aqui são selecionados para fins de demonstração. Você pode otimizar o código conforme necessário para o cenário de implementação. Se estiver usando um SDK da Amazon para programar em função do Amazon Glacier, o cálculo do hash de árvore será feito para você, e bastará fornecer a referência do arquivo.

Example 1: exemplo do Java

O exemplo a seguir mostra como calcular o hash da SHA256 árvore de um arquivo usando Java. Você pode executar esse exemplo fornecendo um local de arquivo como um argumento ou usar o método `TreeHashExample.computeSHA256TreeHash` diretamente do código.

```
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: exemplo do C# .NET

O exemplo a seguir mostra como calcular o hash da SHA256 árvore de um arquivo. Você pode executar esse exemplo fornecendo um local de arquivo como um argumento.

```
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;
        }
    }
}
```

Receber somas de verificação durante o download de dados

Ao recuperar um arquivo usando a Initiate Job API (consulte Initiate Job (trabalhos POST)), você pode especificar um intervalo para recuperação do arquivo. Da mesma maneira, ao fazer download dos dados usando a Get Job Output API (consulte Get Job Output (GET output)), você pode especificar um intervalo de dados para download. Existem duas características desses intervalos que precisam ser compreendidas quando você está recuperando e fazendo download dos dados do arquivo. O intervalo a ser recuperado precisa estar alinhado em megabytes ao arquivo. O intervalo de recuperação e o intervalo de download devem estar alinhados ao hash de árvore para receber valores da soma de verificação quando você faz download dos seus dados. A definição desses dois tipos de alinhamentos de intervalo é a seguinte:

- Megabyte alinhado - Um intervalo [StartByte, EndBytes] é um megabyte (1024*1024) alinhado quando StartBytesé divisível por 1 MB e EndBytesmais 1 é divisível por 1 MB ou é igual ao final do arquivo especificado (tamanho do byte do arquivo menos 1). Um intervalo usado na Initiate Job API, se especificado, é deverá estar alinhado a megabytes.
- Alinhado com hash de árvore - Um intervalo [StartBytes, EndBytes] é um hash de árvore alinhado em relação a um arquivo se e somente se a raiz do hash de árvore construído sobre o intervalo for equivalente a um nó no hash de árvore de todo o arquivo. O intervalo de recuperação e o intervalo de download devem estar alinhados ao hash de árvore para receber valores da soma de verificação dos dados cujo download você fez. Para obter um exemplo de intervalos e a relação com o hash de árvore do arquivo, consulte Exemplo de hash de árvore: recuperar um intervalo de arquivo que esteja alinhado ao hash de árvore.

  Um intervalo alinhado ao hash de árvore também está alinhado a megabytes. No entanto, um intervalo alinhado a megabytes não necessariamente está alinhado ao hash de árvore.

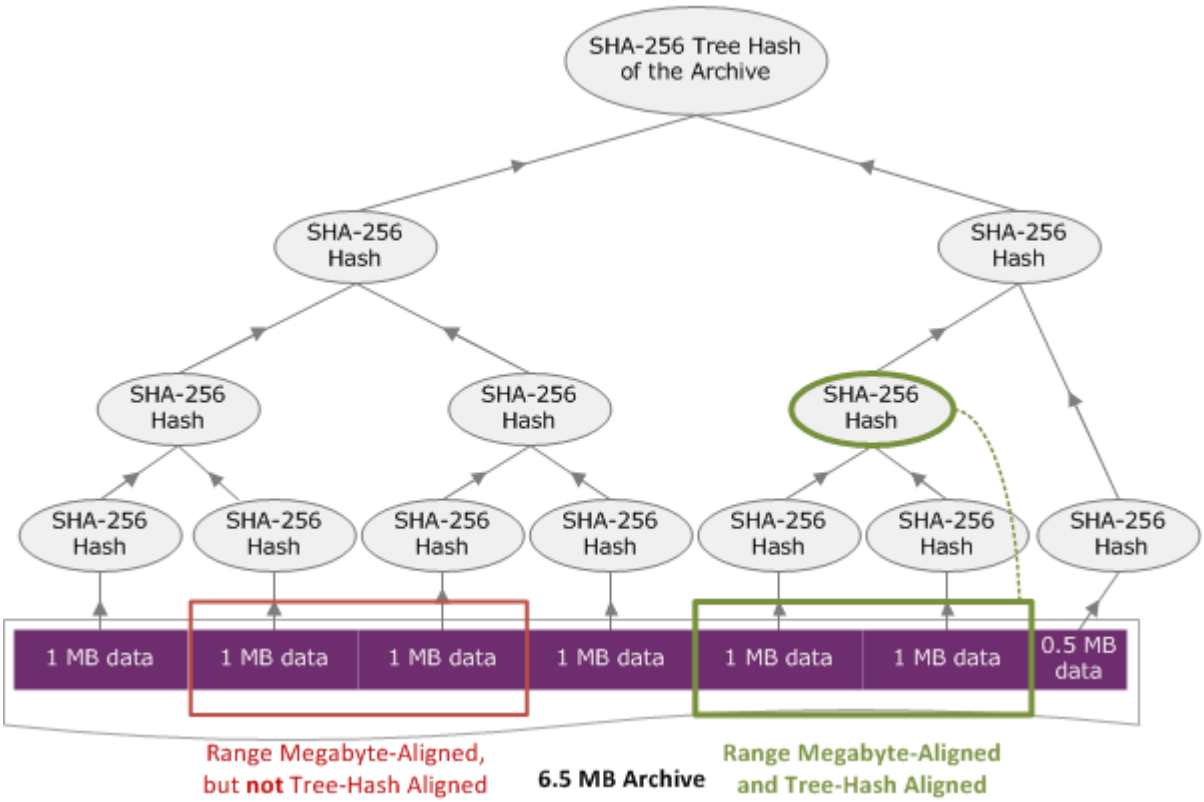
Os seguintes casos descrevem quando você recebe um valor da soma de verificação ao fazer download dos dados do arquivo:

- Se você não especificar um intervalo a ser recuperado na solicitação Initiate Job e fizer download de todo o arquivo na solicitação Get Job.
- Se você não especificar um intervalo a ser recuperado na solicitação Initiate Job e especificar um intervalo alinhado ao hash de árvore para download na solicitação Get Job.
- Se você especificar um intervalo alinhado ao hash de árvore a ser recuperado na solicitação Initiate Job e fizer download de todo o intervalo na solicitação Get Job.
- Se você especificar um intervalo alinhado ao hash de árvore a ser recuperado na solicitação Initiate Job e especificar um intervalo alinhado ao hash de árvore para download na solicitação Get Job.

Se especificar um intervalo a ser recuperado na solicitação Initiate Job que não esteja alinhado ao hash de árvore, você ainda poderá obter os dados do arquivo, mas nenhum valor de soma de verificação retornará quando fizer download dos dados na solicitação Get Job.

Exemplo de hash de árvore: recuperar um intervalo de arquivo que esteja alinhado ao hash de árvore

Suponhamos que você tenha um arquivo de 6,5 MB no cofre e queira recuperar 2 MB do arquivo. A maneira como você especifica o intervalo de 2 MB na solicitação Initiate Job determina se recebe valores de soma de verificação de dados quando faz download dos dados. O diagrama a seguir ilustra dois intervalos de 2 MB para o arquivo de 6,5 MB cujo download você pode fazer. Ambos os intervalos estão alinhados a megabytes, mas somente um está alinhado ao hash de árvore.



Especificação do intervalo alinhado ao hash de árvore

Esta seção fornece a especificação exata para o que constitui um intervalo alinhado ao hash de árvore. Os intervalos alinhados ao hash de árvore são importantes quando você está fazendo download de uma parte de um arquivo e especifica o intervalo de dados a serem recuperados, além do intervalo de download dos dados recuperados. Se ambos os intervalos estiverem alinhados ao hash de árvore, você receberá dados da soma de verificação ao fazer download dos dados.

Um intervalo [A, B] estará alinhado ao hash de árvore em relação a um arquivo somente se um novo hash de árvore for compilado [A, B], a raiz do hash de árvore desse intervalo for equivalente a um nó no hash de árvore de todo o arquivo. Você pode ver isso mostrado no diagrama em Exemplo de hash de árvore: recuperar um intervalo de arquivo que esteja alinhado ao hash de árvore. Nesta seção, fornecemos a especificação do alinhamento do hash de árvore.

Considere [P, Q) como a consulta de intervalo de um arquivo de N megabytes (MB) e que P e Q são múltiplos de um MB. O intervalo real inclusivo é [P MB, Q MB – 1 byte], mas por uma questão de simplicidade, mostramos esse intervalo como [P, Q). Assim, com essas considerações,

- Se P for um número ímpar, haverá somente um intervalo alinhado ao hash de árvore possível – ou seja, [P, P +1 MB).

- Se P for um número par e k for um número máximo em que P pode ser gravado como 2k * X, haverá, no máximo, k intervalos alinhados ao hash de árvore de intervalo começando com P. X é um número inteiro maior que 0. Os intervalos alinhados ao hash de árvore estão nas seguintes categorias:
  - Para cada i, em que (0 <= i <= k) e onde $P + 2^i < N$ e $[P, Q + 2^i)$ é um intervalo alinhado ao hash de árvore.
  - P = 0 é o caso especial em que A = 2[lgN]*0

Respostas de erro

No caso de um erro, a API retorna uma das seguintes exceções:

Código	Description	Código de status HTTP	Tipo
`AccessDeniedException`	Retornado se houve uma tentativa de acessar um recurso não permitido por uma política AWS Identity and Access Management (IAM) ou se o Conta da AWS ID incorreto foi usado no URI da solicitação. Para obter mais informações, consulte Gerenciamento de identidade e acesso para o Amazon Glacier.	`403 Forbidden`	Cliente
`BadRequest`	Retornado se a solicitação não puder ser processada.	`400 Bad Request`	Cliente
`ExpiredTokenException`	Retornado se o token de segurança usado na solicitação tiver expirado.	`403 Forbidden`	Cliente
`InsufficientCapacityException`	Retornado se houver capacidade insuficiente para processar a solicitação expressa. Esse erro se aplica somente a recuperaç	`503 Service Unavailable`	Servidor

Código	Description	Código de status HTTP	Tipo
	ões expressas, e não a recuperaç ões padrão ou em massa.		
`InvalidParameterValueException`	Retornado se um parâmetro da solicitação estiver especificado incorretamente.	`400 Bad Request`	Cliente
`InvalidSignatureException`	Retornado se a assinatura da solicitação for inválida.	`403 Forbidden`	Cliente
`LimitExceededException`	Retornado se a solicitação resultar em um dos limites a seguir excedido, um limite de cofre, um limite de tags ou o limite da capacidade provision ada.	`400 Bad Request`	Cliente
`MissingAuthenticationTokenException`	Retornado se dados da autentica ção não forem encontrados para a solicitação.	`400 Bad Request`	Cliente
`MissingParameterValueException`	Retornado se um cabeçalho obrigatório ou um parâmetro não for encontrado na solicitação.	`400 Bad Request`	Cliente
`PolicyEnforcedException`	Retornado se um trabalho de recuperação excederá o limite da taxa de recuperação da política de dados atual. Para obter mais informações sobre políticas de recuperação de dados, consulte Políticas de recuperação de dados do Amazon Glacier.	`400 Bad Request`	Cliente

Código	Description	Código de status HTTP	Tipo
`ResourceNotFoundException`	Retornado se o recurso especificado, como um cofre, um ID de upload ou um ID de trabalho, não existir.	`404 Not Found`	Cliente
`RequestTimeoutException`	Retornado em caso de upload de um arquivo e o Amazon Glacier expire enquanto recebe o upload.	`408 Request Timeout`	Cliente
`SerializationException`	Retornado se o corpo da solicitação for inválido. Em caso de inclusão de uma carga útil JSON, verifique se ela está bem formada.	`400 Bad Request`	Cliente
`ServiceUnavailableException`	Retornado se o serviço não puder concluir a solicitação.	`500 Internal Server Error`	Servidor
`ThrottlingException`	Retornado se você precisar reduzir a taxa de solicitações para o Amazon Glacier.	`400 Bad Request`	Cliente
`UnrecognizedClientException`	Retornado se o ID de chave de acesso ou o token de segurança for inválido.	`400 Bad Request`	Cliente

Vários Amazon Glacier APIs retornam a mesma exceção, mas com mensagens de exceção diferentes para ajudá-lo a solucionar o erro específico encontrado.

O Amazon Glacier retorna informações do erro no corpo da resposta. Os exemplos a seguir mostram algumas das respostas de erro.

Exemplo 1: descrever solicitação de trabalho com um ID de trabalho não existente

Suponhamos que você envie uma solicitação Trabalho de descrição (GET JobID) para um trabalho não existente. Ou seja, você especifica um ID de trabalho não existente.

```
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
```

Em resposta, o Amazon Glacier retorna a resposta de erro a seguir.

```
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"
  }
```

Em que:

Código

Uma das exceções em geral.

Tipo: string

Message

Uma descrição genérica da condição do erro específica da API que retorna o erro.

Tipo: string

Tipo

A origem do erro. O campo pode ter um dos seguintes valores: `Client`, `Server` ou `Unknown`.

Tipo: string

Observe o seguinte na resposta anterior:

- Para a resposta de erro, o Amazon Glacier retorna valores de código de status `4xx` e `5xx`. Neste exemplo, o código de status é `404 Not Found`.
- O `Content-Type` do valor de cabeçalho `application/json` indica JSON no corpo
- O JSON no corpo fornece as informações de erro.

Na solicitação anterior, em vez de um ID de trabalho inválido, suponhamos que você especifique um cofre não existente. A resposta retorna uma mensagem diferente.

```
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"
}
```

Exemplo 2: solicitação List Jobs com um valor inválido para o parâmetro da solicitação

Neste exemplo, você envia uma solicitação List Jobs (GET jobs) para recuperar trabalhos de cofre com um `statuscode` específico e fornece um valor `statuscode finished`, em vez dos valores aceitáveis `InProgress`, `Succeeded` ou `Failed`.

```
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
```

O Amazon Glacier retorna a `InvalidParameterValueException` com uma mensagem apropriada.

```
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"
}
```

Operações de cofre

Estas são as operações de cofre disponíveis para serem usadas no Amazon Glacier.

Tópicos

- Abort Vault Lock (DELETE lock-policy)
- Add Tags To Vault (POST tags add)
- Create Vault (PUT vault)
- Complete Vault Lock (POST lockId)
- Delete Vault (DELETE vault)
- Política de acesso de exclusão do cofre (DELETE access-policy)
- Delete Vault Notifications (DELETE notification-configuration)
- Describe Vault (GET vault)
- Get Vault Access Policy (GET access-policy)
- Get Vault Lock (GET lock-policy)
- Get Vault Notifications (GET notification-configuration)
- Initiate Vault Lock (POST lock-policy)

- Listar tags para cofre (GET tags)
- List Vaults (GET vaults)
- Remove Tags From Vault (POST tags remove)
- Set Vault Access Policy (PUT access-policy)
- Definir configuração de notificação de cofre (PUT notification-configuration)

Abort Vault Lock (DELETE lock-policy)

Descrição

Esta operação interrompe o processo de bloqueio do cofre se o bloqueio do cofre não estiver no estado `Locked`. Se o bloqueio do cofre estiver no estado `Locked` quando essa operação for solicitada, a operação retornará um erro `AccessDeniedException`. Interromper o processo de bloqueio do cofre remove a política de bloqueio do cofre especificado.

Um bloqueio de cofre é colocado no estado `InProgress` chamando Initiate Vault Lock (POST lock-policy). Um bloqueio de cofre é colocado no estado `Locked` chamando Complete Vault Lock (POST lockId). Você pode obter o estado de um bloqueio de cofre chamando Get Vault Lock (GET lock-policy). Para obter mais informações sobre o processo de bloqueio de cofre, consulte Vault Lock do Amazon Glacier. Para obter mais informações sobre políticas de bloqueio de cofre, consulte Políticas do Vault Lock.

Essa operação é idempotente. Você poderá chamar essa operação com êxito várias vezes, se o bloqueio de cofre estiver no estado `InProgress` ou se não houver política associada ao cofre.

Solicitações

Para excluir a política de bloqueio de cofre, envie uma solicitação `DELETE` HTTP para o URI do sub-recurso `lock-policy` do cofre.

Sintaxe

```
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**
>
> O valor de `AccountId` é o ID da Conta da AWS. Esse valor deve corresponder ao ID de Conta da AWS associado às credenciais usadas para assinar a solicitação. Você pode especificar um ID da Conta da AWS ou, de opcionalmente, um '-' (hífen) único. Nesse caso, o Amazon Glacier usa o ID de Conta da AWS associado às credenciais usadas para assinar a solicitação. Se você especificar o ID da conta, não inclua nenhum hífen ('-') no ID.

Parâmetros da solicitação

Essa operação não usa parâmetros de solicitação.

Cabeçalhos de solicitação

Essa operação usa somente os cabeçalhos de solicitação que são comuns a todas as operações. Para informações sobre os cabeçalhos de solicitação comuns, consulte Cabeçalhos de solicitação comuns.

Corpo da Solicitação

Essa operação não tem um corpo de solicitação.

Respostas

Se a política for excluída com êxito, o Amazon Glacier retornará uma resposta `HTTP 204 No Content`.

Sintaxe

```
HTTP/1.1 204 No Content
x-amzn-RequestId: x-amzn-RequestId
Date: Date
```

Cabeçalhos de resposta

Essa operação usa somente os cabeçalhos de resposta que são comuns a maioria das respostas. Para informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Corpo da resposta

Essa operação não retorna um corpo de resposta.

Erros

Para obter informações sobre exceções e mensagens de erro do Amazon Glacier, consulte Respostas de erro.

Exemplos

O exemplo a seguir demonstra como interromper o processo de bloqueio do cofre.

Exemplo de solicitação

Neste exemplo, uma solicitação `DELETE` é enviada para o sub-recurso `lock-policy` do cofre chamado **`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
```

Exemplo de resposta

Se a política for excluída com êxito, o Amazon Glacier retornará uma resposta `HTTP 204 No Content` conforme mostrado no exemplo a seguir.

```
HTTP/1.1 204 No Content
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:00:00 GMT
```

Seções relacionadas

- Complete Vault Lock (POST lockId)
- Get Vault Lock (GET lock-policy)

- Initiate Vault Lock (POST lock-policy)

Consulte também

Para obter mais informações sobre como usar essa API em um dos Amazon SDKs específicos de linguagem, consulte o seguinte:

- AWS Command Line Interface

Add Tags To Vault (POST tags add)

Esta operação adiciona as tags especificadas a um cofre. Cada tag é composta de uma chave e de um valor. Cada cofre pode ter até 50 tags. Se a solicitação excedesse o limite de tags do cofre, a operação lançaria o erro `LimitExceededException`.

Se uma tag já existir no cofre em uma chave especificada, o valor da chave existente será substituído. Para obter mais informações sobre tags, consulte Marcação de recursos do Amazon Glacier.

Sintaxe da Solicitação

Para adicionar tags a um cofre, envie uma solicitação HTTP POST para o URI das tags conforme mostrado no exemplo da sintaxe a seguir.

```
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**
>
> O valor de `AccountId` é o ID da Conta da AWS. Esse valor deve corresponder ao ID de Conta da AWS associado às credenciais usadas para assinar a solicitação. Você pode especificar um ID da Conta da AWS ou, de opcionalmente, um '-' (hífen) único. Nesse caso, o Amazon Glacier usa o ID de Conta da AWS associado às credenciais usadas para assinar a solicitação. Se você especificar o ID da conta, não inclua nenhum hífen ('-') no ID.

Parâmetros da solicitação

Nome	Descrição	Obrigatório
`operation=add`	Um parâmetro de string de consulta única `operation` com um valor de `add` para diferenciá-lo de Remove Tags From Vault (POST tags remove).	Sim

Cabeçalhos de solicitação

Essa operação usa somente os cabeçalhos de solicitação que são comuns a todas as operações. Para informações sobre os cabeçalhos de solicitação comuns, consulte Cabeçalhos de solicitação comuns.

Corpo da Solicitação

O corpo da solicitação contém os campos JSON a seguir.

Tags da

As tags a serem adicionadas ao cofre. Cada tag é composta de uma chave e de um valor. O valor pode ser uma string vazia.

Tipo: mapa de string para string

Restrições de tamanho: tamanho mínimo de 1. Tamanho máximo 10.

Obrigatório: Sim

Respostas

Se a solicitação de operação for bem-sucedida, o serviço retornará uma resposta `204 No Content` HTTP.

Sintaxe

```
HTTP/1.1 204 No Content
x-amzn-RequestId: x-amzn-RequestId
Date: Date
```

Cabeçalhos de resposta

Essa operação usa somente os cabeçalhos de resposta que são comuns a maioria das respostas. Para informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Corpo da resposta

Essa operação não retorna um corpo de resposta.

Erros

Para obter informações sobre exceções e mensagens de erro do Amazon Glacier, consulte Respostas de erro.

Exemplos

Exemplo de solicitação

O exemplo a seguir envia uma solicitação HTTP POST com as tags a serem adicionadas ao cofre.

```
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"
    }
}
```

Exemplo de resposta

Se a solicitação for bem-sucedida, o Amazon Glacier retornará um `HTTP 204 No Content` conforme mostrado no exemplo a seguir.

```
HTTP/1.1 204 No Content
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:02:00 GMT
```

Seções relacionadas

- Listar tags para cofre (GET tags)
- Remove Tags From Vault (POST tags remove)

Consulte também

Para obter mais informações sobre como usar essa API em um dos Amazon SDKs específicos de linguagem, consulte o seguinte:

- AWS Command Line Interface

Create Vault (PUT vault)

Descrição

Esta operação cria um novo cofre com o nome especificado. O nome do cofre deve ser exclusivo dentro de uma região da AWS para uma Conta da AWS. Você pode criar até 1.000 cofres por conta. Para obter mais informações sobre como criar mais cofres, acesse a página de detalhes de produto do Amazon Glacier.

Você deve usar as diretrizes a seguir ao nomear um cofre.

- Os nomes podem ter de 1 a 255 caracteres.
- Os caracteres válidos são a-z, A-Z, 0-9, '_' (sublinhado), '-' (hífen) e '.' (ponto).

Essa operação é idempotente; você pode enviar a mesma solicitação várias vezes, e ela não terá mais efeito depois da primeira vez em que o Amazon Glacier criar o cofre especificado.

Solicitações

Sintaxe

Para criar um cofre, envie uma solicitação PUT HTTP para o URI do cofre a ser criado.

```
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**
>
> O valor de `AccountId` é o ID da Conta da AWS. Esse valor deve corresponder ao ID de Conta da AWS associado às credenciais usadas para assinar a solicitação. Você pode especificar um ID da Conta da AWS ou, de opcionalmente, um '-' (hífen) único. Nesse caso, o Amazon Glacier usa o ID de Conta da AWS associado às credenciais usadas para assinar a solicitação. Se você especificar o ID da conta, não inclua nenhum hífen ('-') no ID.

Parâmetros da solicitação

Essa operação não usa parâmetros de solicitação.

Cabeçalhos de solicitação

Essa operação usa somente os cabeçalhos de solicitação que são comuns a todas as operações. Para informações sobre os cabeçalhos de solicitação comuns, consulte Cabeçalhos de solicitação comuns.

Corpo da Solicitação

O corpo da solicitação dessa operação deve estar vazio (0 byte).

Respostas

Sintaxe

```
HTTP/1.1 201 Created
x-amzn-RequestId: x-amzn-RequestId
Date: Date
Location: Location
```

Cabeçalhos de resposta

Uma resposta bem-sucedida inclui os seguintes cabeçalhos de resposta, além dos cabeçalhos de resposta que são comuns a todas as operações. Para mais informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Nome	Descrição
`Location`	O caminho do URI relativo do cofre que foi criado. Tipo: string

Corpo da resposta

Essa operação não retorna um corpo de resposta.

Erros

Para obter informações sobre exceções e mensagens de erro do Amazon Glacier, consulte Respostas de erro.

Exemplos

Exemplo de solicitação

O exemplo a seguir envia uma solicitação PUT HTTP para criar um cofre chamado `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
```

Exemplo de resposta

O Amazon Glacier cria o cofre e retorna o caminho do URI relativo do cofre no cabeçalho `Location`. O ID da conta sempre é exibido no cabeçalho `Location`, independentemente do ID da conta ou de um hífen ('-') ter sido especificado na solicitação.

```
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
```

Seções relacionadas

- List Vaults (GET vaults)
- Delete Vault (DELETE vault)
- Gerenciamento de identidade e acesso para o Amazon Glacier

Consulte também

Para obter mais informações sobre como usar essa API em um dos Amazon SDKs específicos de linguagem, consulte o seguinte:

- AWS Command Line Interface

Complete Vault Lock (POST lockId)

Descrição

Esta operação conclui o processo de bloqueio de cofre fazendo a transição do bloqueio de cofre do estado `InProgress` para o estado `Locked`, o que torna a política de bloqueio de cofre inalterável.

Um bloqueio de cofre é colocado no estado `InProgress` chamando Initiate Vault Lock (POST lock-policy). Você pode obter o estado do bloqueio de cofre chamando Get Vault Lock (GET lock-policy). Para obter mais informações sobre o processo de bloqueio de cofre, consulte Vault Lock do Amazon Glacier.

Essa operação é idempotente. Essa solicitação sempre será bem-sucedida se o bloqueio de cofre estiver no estado `Locked` e o ID de bloqueio fornecido corresponder ao ID de bloqueio usado originalmente para bloquear o cofre.

Se um ID de bloqueio inválido for passado na solicitação quando o bloqueio cofre estiver no estado `Locked`, a operação retornará um erro `AccessDeniedException`. Se um ID de bloqueio inválido for passado na solicitação quando o bloqueio cofre estiver no estado `InProgress`, a operação lançará um erro `InvalidParameter`.

Solicitações

Para concluir o processo de bloqueio de cofre, envie uma solicitação HTTP `POST` para o URI do sub-recurso `lock-policy` do cofre com um ID de bloqueio válido.

Sintaxe

```
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**
>
> O valor de `AccountId` é o ID da Conta da AWS. Esse valor deve corresponder ao ID de Conta da AWS associado às credenciais usadas para assinar a solicitação. Você pode especificar um ID da Conta da AWS ou, de opcionalmente, um '-' (hífen) único. Nesse caso, o Amazon Glacier usa o ID de Conta da AWS associado às credenciais usadas para assinar a solicitação. Se você especificar o ID da conta, não inclua nenhum hífen ('-') no ID.

O `lockId` valor é o ID de bloqueio obtido por meio de uma solicitação Initiate Vault Lock (POST lock-policy).

Parâmetros da solicitação

Cabeçalhos de solicitação

Essa operação usa somente os cabeçalhos de solicitação que são comuns a todas as operações. Para informações sobre os cabeçalhos de solicitação comuns, consulte Cabeçalhos de solicitação comuns.

Corpo da Solicitação

Essa operação não tem um corpo de solicitação.

Respostas

Se a solicitação de operação for bem-sucedida, o serviço retornará uma resposta `204 No Content` HTTP.

Sintaxe

```
HTTP/1.1 204 No Content
x-amzn-RequestId: x-amzn-RequestId
Date: Date
```

Cabeçalhos de resposta

Essa operação usa somente os cabeçalhos de resposta que são comuns a maioria das respostas. Para informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Corpo da resposta

Essa operação não retorna um corpo de resposta.

Erros

Para obter informações sobre exceções e mensagens de erro do Amazon Glacier, consulte Respostas de erro.

Exemplos

Exemplo de solicitação

O exemplo a seguir envia uma solicitação HTTP POST com o ID de bloqueio para concluir o processo de bloqueio do cofre.

```
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
```

Exemplo de resposta

Se a solicitação for bem-sucedida, o Amazon Glacier retornará uma resposta `HTTP 204 No Content`, conforme mostrado no exemplo a seguir.

```
HTTP/1.1 204 No Content
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:02:00 GMT
```

Seções relacionadas

- Abort Vault Lock (DELETE lock-policy)
- Get Vault Lock (GET lock-policy)
- Initiate Vault Lock (POST lock-policy)

Consulte também

Para obter mais informações sobre como usar essa API em um dos Amazon SDKs específicos de linguagem, consulte o seguinte:

- AWS Command Line Interface

Delete Vault (DELETE vault)

Descrição

Essa operação exclui um cofre. O Amazon Glacier só excluirá um cofre se não houver arquivos nele de acordo com o último inventário e não houver gravações no cofre desde o inventário mais recente. Se qualquer uma dessas condições não for atendida, a exclusão do cofre falhará (ou seja, o cofre não será removido), e o Amazon Glacier retornará um erro.

Você pode usar a operação Describe Vault (GET vault) que fornece informações do cofre, inclusive o número de arquivos no cofre. No entanto, as informações se baseiam no inventário de cofre gerado pelo Amazon Glacier mais recentemente.

Essa operação é idempotente.

> **Note**
>
> Quando você exclui um cofre, a política de acesso ao cofre anexada ao cofre também é excluída. Para obter mais informações sobre políticas de acesso ao cofre, consulte Políticas de acesso ao cofre.

Solicitações

Para excluir um cofre, envie uma solicitação `DELETE` para o URI do recurso de cofre.

Sintaxe

```
DELETE /AccountId/vaults/VaultName HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: Date
Authorization: SignatureValue
x-amz-glacier-version: 2012-06-01
```

> **Note**
>
> O valor de `AccountId` é o ID de Conta da AWS da conta proprietária do cofre. Você pode especificar um ID da Conta da AWS ou, de opcionalmente, um '-' (hífen) único. Nesse caso,

o Amazon Glacier usa o ID da Conta da AWS de associado às credenciais usadas para assinar a solicitação. Se você usar um ID de conta, não inclua nenhum hífen ('-') no ID.

Parâmetros da solicitação

Essa operação não usa parâmetros de solicitação.

Cabeçalhos de solicitação

Essa operação usa somente os cabeçalhos de solicitação que são comuns a todas as operações. Para informações sobre os cabeçalhos de solicitação comuns, consulte Cabeçalhos de solicitação comuns.

Corpo da Solicitação

Essa operação não tem um corpo de solicitação.

Respostas

Sintaxe

```
HTTP/1.1 204 No Content
x-amzn-RequestId: x-amzn-RequestId
Date: Date
```

Cabeçalhos de resposta

Essa operação usa somente os cabeçalhos de resposta que são comuns a maioria das respostas. Para informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Corpo da resposta

Essa operação não retorna um corpo de resposta.

Erros

Para obter informações sobre exceções e mensagens de erro do Amazon Glacier, consulte Respostas de erro.

Exemplos

Exemplo de solicitação

O exemplo a seguir exclui um cofre chamado `examplevault`. A solicitação de exemplo é uma solicitação `DELETE` para o URI do recurso (o cofre) para exclusão.

```
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
```

Exemplo de resposta

```
HTTP/1.1 204 No Content
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:02:00 GMT
```

Seções relacionadas

- Create Vault (PUT vault)
- List Vaults (GET vaults)
- Initiate Job (trabalhos POST)
- Gerenciamento de identidade e acesso para o Amazon Glacier

Consulte também

Para obter mais informações sobre como usar essa API em um dos Amazon SDKs específicos de linguagem, consulte o seguinte:

- AWS Command Line Interface

Política de acesso de exclusão do cofre (DELETE access-policy)

Description

Essa operação exclui a política de acesso associada ao cofre especificado. A operação acaba sendo consistente, ou seja, pode levar algum tempo até o Amazon Glacier remover completamente a política de acesso, e você ainda poderá ver o efeito da política por um curto período depois de enviar a solicitação de exclusão.

Essa operação é idempotente. Você poderá invocar a exclusão várias vezes, mesmo se não houver política associada ao cofre. Para obter mais informações sobre políticas de acesso ao cofre, consulte Políticas de acesso ao cofre.

Solicitações

Para excluir a política de acesso ao cofre atual, envie uma solicitação `DELETE` HTTP para o URI do sub-recurso `access-policy` do cofre.

Sintaxe

```
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**
>
> O `AccountId` valor é o Conta da AWS ID da conta proprietária do cofre. Você pode especificar um Conta da AWS ID ou, opcionalmente, um único '-' (hífen). Nesse caso, o Amazon Glacier usa Conta da AWS o ID associado às credenciais usadas para assinar a solicitação. Se você usar um ID de conta, não inclua nenhum hífen ('-') no ID.

Parâmetros da solicitação

Essa operação não usa parâmetros de solicitação.

Cabeçalhos de solicitação

Essa operação usa somente os cabeçalhos de solicitação que são comuns a todas as operações. Para informações sobre os cabeçalhos de solicitação comuns, consulte Cabeçalhos de solicitação comuns.

Corpo da Solicitação

Essa operação não tem um corpo de solicitação.

Respostas

Em resposta, o Amazon Glacier retornará `204 No Content`, se a política for excluída com êxito.

Sintaxe

```
HTTP/1.1 204 No Content
x-amzn-RequestId: x-amzn-RequestId
Date: Date
```

Cabeçalhos de resposta

Essa operação usa somente os cabeçalhos de resposta que são comuns a maioria das respostas. Para informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Corpo da resposta

Essa operação não retorna um corpo de resposta.

Erros

Para obter informações sobre exceções e mensagens de erro do Amazon Glacier, consulte Respostas de erro.

Exemplos

O exemplo a seguir demonstra como excluir uma política de acesso ao cofre.

Exemplo de solicitação

Neste exemplo, uma solicitação `DELETE` é enviada para o sub-recurso `access-policy` do cofre chamado **`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
```

Exemplo de resposta

Em resposta, se a política for excluída com êxito, o Amazon Glacier retornará um `204 No Content` conforme mostrado no exemplo a seguir.

```
HTTP/1.1 204 No Content
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:00:00 GMT
```

Seções relacionadas

- Get Vault Access Policy (GET access-policy)
- Set Vault Access Policy (PUT access-policy)

Consulte também

Para obter mais informações sobre o uso dessa API em uma das linguagens específicas da Amazon SDKs, consulte o seguinte:

- AWS Command Line Interface

Delete Vault Notifications (DELETE notification-configuration)

Description

Essa operação exclui a configuração de notificação definida para um cofre Definir configuração de notificação de cofre (PUT notification-configuration). A operação acaba sendo consistente, ou

seja, pode levar algum tempo até o Amazon Glacier desabilitar completamente as notificações, e você ainda receberá algumas notificações por um curto período depois de enviar a solicitação de exclusão.

Solicitações

Para excluir a configuração de notificação do cofre, envie uma solicitação `DELETE` para o sub-recurso `notification-configuration` do cofre.

Sintaxe

```
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**
>
> O `AccountId` valor é o Conta da AWS ID da conta proprietária do cofre. Você pode especificar um Conta da AWS ID ou, opcionalmente, um único '-' (hífen). Nesse caso, o Amazon Glacier usa Conta da AWS o ID associado às credenciais usadas para assinar a solicitação. Se você usar um ID de conta, não inclua nenhum hífen ('-') no ID.

Parâmetros da solicitação

Essa operação não usa parâmetros de solicitação.

Cabeçalhos de solicitação

Essa operação usa somente os cabeçalhos de solicitação que são comuns a todas as operações. Para informações sobre os cabeçalhos de solicitação comuns, consulte Cabeçalhos de solicitação comuns.

Corpo da Solicitação

Essa operação não tem um corpo de solicitação.

Respostas

Sintaxe

```
HTTP/1.1 204 No Content
x-amzn-RequestId: x-amzn-RequestId
Date: Date
```

Cabeçalhos de resposta

Essa operação usa somente os cabeçalhos de resposta que são comuns a maioria das respostas. Para informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Corpo da resposta

Essa operação não retorna um corpo de resposta.

Erros

Para obter informações sobre exceções e mensagens de erro do Amazon Glacier, consulte Respostas de erro.

Exemplos

O exemplo a seguir demonstra como remover a configuração de notificação de um cofre.

Exemplo de solicitação

Neste exemplo, uma solicitação `DELETE` é enviada para o sub-recurso `notification-configuration` do cofre chamado `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
```

Exemplo de resposta

```
HTTP/1.1 204 No Content
```

```
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:00:00 GMT
```

Seções relacionadas

- Get Vault Notifications (GET notification-configuration)
- Definir configuração de notificação de cofre (PUT notification-configuration)
- Gerenciamento de identidade e acesso para o Amazon Glacier

Consulte também

Para obter mais informações sobre o uso dessa API em uma das linguagens específicas da Amazon SDKs, consulte o seguinte:

- AWS Command Line Interface

Describe Vault (GET vault)

Descrição

Essa operação retorna informações sobre um cofre, inclusive o Amazon Resource Name (ARN – Nome de recurso da Amazon), a data na qual o cofre foi criado, o número de arquivos contidos no cofre e o tamanho total de todos os arquivos no cofre. O número de arquivos e o tamanho total são desde o inventário de cofre gerado mais recentemente pelo Amazon Glacier (consulte Trabalhar com cofres no Amazon Glacier). O Amazon Glacier gera inventários de cofre aproximadamente diários. Isso significa que, se você adicionar ou remover um arquivo de um cofre e enviar imediatamente uma solicitação Describe Vault, a resposta poderá não refletir as alterações.

Solicitações

Para obter informações sobre um cofre, envie uma solicitação `GET` para o URI do recurso de cofre específico.

Sintaxe

```
GET /AccountId/vaults/VaultName HTTP/1.1
Host: glacier.Region.amazonaws.com
```

```
Date: Date
Authorization: SignatureValue
x-amz-glacier-version: 2012-06-01
```

> **Note**
>
> O valor de `AccountId` é o ID de Conta da AWS da conta proprietária do cofre. Você pode especificar um ID da Conta da AWS ou, de opcionalmente, um '-' (hífen) único. Nesse caso, o Amazon Glacier usa o ID de Conta da AWS associado às credenciais usadas para assinar a solicitação. Se você usar um ID de conta, não inclua nenhum hífen ('-') no ID.

Parâmetros da solicitação

Essa operação não usa parâmetros de solicitação.

Cabeçalhos de solicitação

Essa operação usa somente os cabeçalhos de solicitação que são comuns a todas as operações. Para informações sobre os cabeçalhos de solicitação comuns, consulte Cabeçalhos de solicitação comuns.

Corpo da Solicitação

Essa operação não tem um corpo de solicitação.

Respostas

Sintaxe

```
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
}
```

Cabeçalhos de resposta

Essa operação usa somente os cabeçalhos de resposta que são comuns a maioria das respostas. Para informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Corpo da resposta

O corpo da resposta contém os seguintes campos JSON.

CreationDate

A data UTC quando o cofre foi criado.

Tipo: uma representação de string no formato de data ISO 8601, por exemplo `2013-03-20T17:03:43.221Z`.

LastInventoryDate

A data UTC quando o Amazon Glacier concluiu o inventário de cofre mais recente. Para obter mais informações sobre como iniciar um inventário para um cofre, consulte Initiate Job (trabalhos POST).

Tipo: uma representação de string no formato de data ISO 8601, por exemplo `2013-03-20T17:03:43.221Z`.

NumberOfArchives

O número de arquivos no cofre de acordo com o último inventário de cofre. Esse campo retornará nulo se um inventário ainda não tiver sido executado no cofre; por exemplo, se você tiver acabado de criar o cofre.

Tipo: número

SizeInBytes

O tamanho total em bytes dos arquivos no cofre, inclusive eventuais custos indiretos por arquivo, desde a data do último inventário. Esse campo retornará `null` se um inventário ainda não tiver sido executado no cofre; por exemplo, se você tiver acabado de criar o cofre.

Tipo: número

VaultARN

O nome de recurso da Amazon (ARN) do cofre.

Tipo: string

VaultName

O nome do cofre que foi especificado no momento da criação. O nome do cofre também é incluído no ARN do cofre.

Tipo: string

Erros

Para obter informações sobre exceções e mensagens de erro do Amazon Glacier, consulte Respostas de erro.

Exemplos

Exemplo de solicitação

O exemplo a seguir demonstra como obter informações sobre o cofre chamado `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
```

Exemplo de resposta

```
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"
}
```

Seções relacionadas

- Create Vault (PUT vault)
- List Vaults (GET vaults)
- Delete Vault (DELETE vault)
- Initiate Job (trabalhos POST)
- Gerenciamento de identidade e acesso para o Amazon Glacier

Consulte também

Para obter mais informações sobre como usar essa API em um dos Amazon SDKs específicos de linguagem, consulte o seguinte:

- AWS Command Line Interface

Get Vault Access Policy (GET access-policy)

Description

Esta operação recupera o sub-recurso `access-policy` definido no cofre – para obter mais informações sobre como definir esse sub-recurso, consulte Set Vault Access Policy (PUT access-policy). Se não houver política de acesso definida no cofre, a operação retornará um erro `404 Not found`. Para obter mais informações sobre políticas de acesso ao cofre, consulte Políticas de acesso ao cofre.

Solicitações

Para retornar a política de acesso ao cofre atual, envie uma solicitação `GET` HTTP para o URI do sub-recurso `access-policy` do cofre.

Sintaxe

```
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**
>
> O `AccountId` valor é o Conta da AWS ID da conta proprietária do cofre. Você pode especificar um Conta da AWS ID ou, opcionalmente, um único '-' (hífen). Nesse caso, o Amazon Glacier usa Conta da AWS o ID associado às credenciais usadas para assinar a solicitação. Se você usar um ID de conta, não inclua nenhum hífen ('-') no ID.

Parâmetros da solicitação

Essa operação não usa parâmetros de solicitação.

Cabeçalhos de solicitação

Essa operação usa somente os cabeçalhos de solicitação que são comuns a todas as operações. Para informações sobre os cabeçalhos de solicitação comuns, consulte Cabeçalhos de solicitação comuns.

Corpo da Solicitação

Essa operação não tem um corpo de solicitação.

Respostas

Em resposta, o Amazon Glacier retorna a política de acesso ao cofre em formato JSON no corpo da resposta.

Sintaxe

```
HTTP/1.1 200 OK
```

```
x-amzn-RequestId: x-amzn-RequestId
Date: Date
Content-Type: application/json
Content-Length: length

{
  "Policy": "string"
}
```

Cabeçalhos de resposta

Essa operação usa somente os cabeçalhos de resposta que são comuns a maioria das respostas. Para informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Corpo da resposta

O corpo da resposta contém os seguintes campos JSON.

Política

A política de acesso ao cofre como uma string JSON, que usa"\" como um caractere de escape.

Tipo: string

Erros

Para obter informações sobre exceções e mensagens de erro do Amazon Glacier, consulte Respostas de erro.

Exemplos

O exemplo a seguir demonstra como obter uma política de acesso ao cofre.

Exemplo de solicitação

Neste exemplo, uma solicitação `GET` é enviada para o URI do sub-recurso `access-policy` de um cofre.

```
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
```

Exemplo de resposta

Se a solicitação tiver sido bem-sucedida, o Amazon Glacier retornará a política de acesso ao cofre como uma string JSON no corpo da resposta. A string JSON retornada usa"\" como um caractere de escape, conforme mostrado nos exemplos Set Vault Access Policy (PUT access-policy). No entanto, o exemplo a seguir mostra a string JSON retornada sem caracteres de escape para fins de legibilidade.

```
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"
            }
          }
        }
      ]
```

```
    }
  "
}
```

Seções relacionadas

- Política de acesso de exclusão do cofre (DELETE access-policy)
- Set Vault Access Policy (PUT access-policy)

Consulte também

Para obter mais informações sobre o uso dessa API em uma das linguagens específicas da Amazon SDKs, consulte o seguinte:

- AWS Command Line Interface

Get Vault Lock (GET lock-policy)

Description

Esta operação recupera os seguintes atributos do sub-recurso `lock-policy` definido no cofre especificado:

- A política do bloqueio de cofre definida no cofre.
- O estado do bloqueio de cofre, que é `InProgess` ou `Locked`.
- Quando o ID de bloqueio expira. O ID de bloqueio é usado para concluir o processo de bloqueio do cofre.
- Quando o bloqueio de cofre foi iniciado e colocado no estado `InProgress`.

Um bloqueio de cofre é colocado no estado `InProgress` chamando Initiate Vault Lock (POST lock-policy). Um bloqueio de cofre é colocado no estado `Locked` chamando Complete Vault Lock (POST lockId). Você pode interromper o processo de bloqueio de cofre chamando Abort Vault Lock (DELETE lock-policy). Para obter mais informações sobre o processo de bloqueio de cofre, consulte Vault Lock do Amazon Glacier.

Se não houver política de bloqueio de cofre definida no cofre, a operação retornará um erro `404 Not found`. Para obter mais informações sobre políticas de bloqueio de cofre, consulte Políticas do Vault Lock.

Solicitações

Para retornar a política de bloqueio de cofre atual e outros atributos, envie uma solicitação `GET` HTTP para o URI do sub-recurso `lock-policy` do cofre conforme mostrado no exemplo da sintaxe a seguir.

Sintaxe

```
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**
>
> O `AccountId` valor é o Conta da AWS ID da conta proprietária do cofre. Você pode especificar um Conta da AWS ID ou, opcionalmente, um único '-' (hífen). Nesse caso, o Amazon Glacier usa Conta da AWS o ID associado às credenciais usadas para assinar a solicitação. Se você usar um ID de conta, não inclua nenhum hífen ('-') no ID.

Parâmetros da solicitação

Essa operação não usa parâmetros de solicitação.

Cabeçalhos de solicitação

Essa operação usa somente os cabeçalhos de solicitação que são comuns a todas as operações. Para informações sobre os cabeçalhos de solicitação comuns, consulte Cabeçalhos de solicitação comuns.

Corpo da Solicitação

Essa operação não tem um corpo de solicitação.

Respostas

Em resposta, o Amazon Glacier retorna a política de acesso ao cofre em formato JSON no corpo da resposta.

Sintaxe

```
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"
}
```

Cabeçalhos de resposta

Essa operação usa somente os cabeçalhos de resposta que são comuns a maioria das respostas. Para informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Corpo da resposta

O corpo da resposta contém os seguintes campos JSON.

Política

A política de bloqueio de cofre como uma string JSON, que usa"\" como um caractere de escape.

Tipo: string

Estado

O estado do bloqueio de cofre.

Tipo: string

Valores válidos: `InProgress|Locked`

ExpirationDate

A data e hora UTC em que o ID de bloqueio expira. Esse valor poderá ser `null`, se o bloqueio de cofre estiver em um estado `Locked`.

Tipo: uma representação de string no formato de data ISO 8601, por exemplo `2013-03-20T17:03:43.221Z`.

CreationDate

A data e hora UTC em que o bloqueio de cofre foi colocado no estado `InProgress`.

Tipo: uma representação de string no formato de data ISO 8601, por exemplo `2013-03-20T17:03:43.221Z`.

Erros

Para obter informações sobre exceções e mensagens de erro do Amazon Glacier, consulte Respostas de erro.

Exemplos

O exemplo a seguir demonstra como obter uma política de bloqueio de cofre.

Exemplo de solicitação

Neste exemplo, uma solicitação `GET` é enviada para o URI do sub-recurso `lock-policy` de um cofre.

```
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
```

Exemplo de resposta

Se a solicitação tiver sido bem-sucedida, o Amazon Glacier retornará a política de acesso ao cofre como uma string JSON no corpo da resposta. A string JSON retornada usa"\" como um caractere de escape, conforme mostrado na solicitação de exemplo Initiate Vault Lock (POST lock-policy). No

entanto, o exemplo a seguir mostra a string JSON retornada sem caracteres de escape para fins de legibilidade.

```
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"
}
```

Seções relacionadas

- Abort Vault Lock (DELETE lock-policy)

- Complete Vault Lock (POST lockId)

- Initiate Vault Lock (POST lock-policy)

Consulte também

Para obter mais informações sobre o uso dessa API em uma das linguagens específicas da Amazon SDKs, consulte o seguinte:

- AWS Command Line Interface

Get Vault Notifications (GET notification-configuration)

Description

Essa operação recupera o sub-recurso `notification-configuration` definido no cofre (consulte Definir configuração de notificação de cofre (PUT notification-configuration). Se a configuração de notificação de um cofre não estiver definida, a operação retornará um erro `404 Not Found`. Para obter mais informações sobre notificações de cofre, consulte Configurar notificações do cofre no Amazon Glacier.

Solicitações

Para recuperar as informações de configuração de notificação, envie uma solicitação `GET` para o URI do sub-recurso `notification-configuration` de um cofre.

Sintaxe

```
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**
>
> O `AccountId` valor é o Conta da AWS ID da conta proprietária do cofre. Você pode especificar um Conta da AWS ID ou, opcionalmente, um único '-' (hífen). Nesse caso, o

> Amazon Glacier usa Conta da AWS o ID associado às credenciais usadas para assinar a solicitação. Se você usar um ID de conta, não inclua nenhum hífen ('-') no ID.

Parâmetros da solicitação

Essa operação não usa parâmetros de solicitação.

Cabeçalhos de solicitação

Essa operação usa somente os cabeçalhos de solicitação que são comuns a todas as operações. Para informações sobre os cabeçalhos de solicitação comuns, consulte Cabeçalhos de solicitação comuns.

Corpo da Solicitação

Essa operação não tem um corpo de solicitação.

Respostas

Sintaxe

```
HTTP/1.1 200 OK
x-amzn-RequestId: x-amzn-RequestId
Date: Date
Content-Type: application/json
Content-Length: length
{
  "Events": [
    String,
    ...
  ],
  "SNSTopic": String
}
```

Cabeçalhos de resposta

Essa operação usa somente os cabeçalhos de resposta que são comuns a maioria das respostas. Para informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Corpo da resposta

O corpo da resposta contém os seguintes campos JSON.

Eventos

Uma lista de um ou mais eventos para os quais o Amazon Glacier enviará uma notificação ao tópico do Amazon SNS especificado. Para obter mais informações sobre eventos de cofre para os quais você possa configurar um cofre a fim de publicar notificações, consulte Definir configuração de notificação de cofre (PUT notification-configuration).

Tipo: matriz

SNSTopic

O tópico nome do recurso da Amazon (ARN) do Amazon Simple Notification Service (Amazon SNS). Para obter mais informações, consulte Conceitos básicos do Amazon SNS no Guia de conceitos básicos do Amazon Simple Notification Service.

Tipo: string

Erros

Para obter informações sobre exceções e mensagens de erro do Amazon Glacier, consulte Respostas de erro.

Exemplos

O exemplo a seguir demonstra como recuperar a configuração de notificação de um cofre.

Exemplo de solicitação

Neste exemplo, uma solicitação `GET` é enviada para o sub-recurso `notification-configuration` de um cofre.

```
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
```

Exemplo de resposta

Uma resposta bem-sucedida mostra o documento de configuração de registro em log da auditoria no corpo da resposta em formato JSON. Neste exemplo, a configuração mostra que as notificações de dois eventos (`ArchiveRetrievalCompleted` e `InventoryRetrievalCompleted`) são enviadas para o tópico do 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"
}
```

Seções relacionadas

- Delete Vault Notifications (DELETE notification-configuration)
- Definir configuração de notificação de cofre (PUT notification-configuration)
- Gerenciamento de identidade e acesso para o Amazon Glacier

Consulte também

Para obter mais informações sobre o uso dessa API em uma das linguagens específicas da Amazon SDKs, consulte o seguinte:

- AWS Command Line Interface

Initiate Vault Lock (POST lock-policy)

Descrição

Esta operação inicia o processo de bloqueio do cofre fazendo o seguinte:

- Instalar uma política de bloqueio de cofre no cofre especificado.
- Definir o estado de bloqueio do cofre como `InProgress`.
- Retornar um ID de bloqueio, usado para concluir o processo de bloqueio do cofre.

Você pode definir uma política de bloqueio para cada cofre, e essa política pode ter até 20 KB. Para obter mais informações sobre políticas de bloqueio de cofre, consulte Políticas do Vault Lock.

Você deverá concluir o processo de bloqueio do cofre dentro de 24 horas depois que o bloqueio de cofre entrar no estado `InProgress`. Depois da janela de 24 horas, o ID de bloqueio vai expirar, o cofre sairá automaticamente do estado `InProgress` e a política de bloqueio do cofre será removida do cofre. Você chama Complete Vault Lock (POST lockId) para concluir o processo de bloqueio do cofre definindo o estado do bloqueio de cofre como `Locked`.

> **Note**
>
> Depois que um bloqueio de cofre estiver no estado `Locked`, você não poderá iniciar um novo bloqueio para o cofre.

Você pode interromper o processo de bloqueio de cofre chamando Abort Vault Lock (DELETE lock-policy). Você pode obter o estado do bloqueio de cofre chamando Get Vault Lock (GET lock-policy). Para obter mais informações sobre o processo de bloqueio de cofre, consulte Vault Lock do Amazon Glacier.

Se essa operação for chamada quando o bloqueio de cofre estiver no estado `InProgress`, a operação retornará um erro `AccessDeniedException`. Quando o bloqueio de cofre está no estado `InProgress`, você deverá chamar Abort Vault Lock (DELETE lock-policy) antes de iniciar uma nova política de bloqueio de cofre.

Solicitações

Para iniciar o processo de bloqueio de cofre, envie uma solicitação HTTP `POST` para o URI do sub-recurso `lock-policy` do cofre, conforme mostrado no exemplo de sintaxe a seguir.

Sintaxe

```
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**
>
> O valor de `AccountId` é o ID da Conta da AWS. Esse valor deve corresponder ao ID de Conta da AWS associado às credenciais usadas para assinar a solicitação. Você pode especificar um ID da Conta da AWS ou, de opcionalmente, um '-' (hífen) único. Nesse caso, o Amazon Glacier usa o ID de Conta da AWS associado às credenciais usadas para assinar a solicitação. Se você especificar o ID da conta, não inclua nenhum hífen ('-') no ID.

Parâmetros da solicitação

Essa operação não usa parâmetros de solicitação.

Cabeçalhos de solicitação

Essa operação usa somente os cabeçalhos de solicitação que são comuns a todas as operações. Para informações sobre os cabeçalhos de solicitação comuns, consulte Cabeçalhos de solicitação comuns.

Corpo da Solicitação

O corpo da solicitação contém os campos JSON a seguir.

Política

A política de bloqueio de cofre como uma string JSON, que usa"\" como um caractere de escape.

Tipo: string

Obrigatório: Sim

Respostas

Em resposta, o Amazon Glacier retornará uma resposta `HTTP 201 Created`, se a política for aceita.

Sintaxe

```
HTTP/1.1 201 Created
x-amzn-RequestId: x-amzn-RequestId
Date: Date
x-amz-lock-id: lockId
```

Cabeçalhos de resposta

Uma resposta bem-sucedida inclui os seguintes cabeçalhos de resposta, além dos cabeçalhos de resposta que são comuns a todas as operações. Para mais informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Nome	Descrição
`x-amz-lock-id`	O ID de bloqueio, usado para concluir o processo de bloqueio do cofre. Tipo: string

Corpo da resposta

Essa operação não retorna um corpo de resposta.

Erros

Para obter informações sobre exceções e mensagens de erro do Amazon Glacier, consulte Respostas de erro.

Exemplos

Exemplo de solicitação

O exemplo a seguir envia uma solicitação PUT HTTP para criar o URI do sub-recurso `lock-policy` do cofre. A string JSON `Policy` usa "\" como um caractere de 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\"}}}]}"}
```

Exemplo de resposta

Se a solicitação for bem-sucedida, o Amazon Glacier retornará uma resposta `HTTP 201 Created`, conforme mostrado no exemplo a seguir.

```
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
```

Seções relacionadas

- Abort Vault Lock (DELETE lock-policy)
- Complete Vault Lock (POST lockId)
- Get Vault Lock (GET lock-policy)

Consulte também

Para obter mais informações sobre como usar essa API em um dos Amazon SDKs específicos de linguagem, consulte o seguinte:

- AWS Command Line Interface

Listar tags para cofre (GET tags)

Esta operação lista todas as tags anexadas a um cofre. A operação retornará um mapa vazio, se não houver tags. Para obter mais informações sobre tags, consulte Marcação de recursos do Amazon Glacier.

Sintaxe da Solicitação

Para listar as tags de um cofre, envie uma solicitação GET HTTP para o URI das tags conforme mostrado no exemplo da sintaxe a seguir.

```
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**
>
> O valor de `AccountId` é o ID da Conta da AWS. Esse valor deve corresponder ao ID de Conta da AWS associado às credenciais usadas para assinar a solicitação. Você pode especificar um ID da Conta da AWS ou, de opcionalmente, um '-' (hífen) único. Nesse caso, o Amazon Glacier usa o ID de Conta da AWS associado às credenciais usadas para assinar a solicitação. Se você especificar o ID da conta, não inclua nenhum hífen ('-') no ID.

Parâmetros da solicitação

Essa operação não usa parâmetros de solicitação.

Cabeçalhos de solicitação

Essa operação usa somente os cabeçalhos de solicitação que são comuns a todas as operações. Para informações sobre os cabeçalhos de solicitação comuns, consulte Cabeçalhos de solicitação comuns.

Corpo da Solicitação

Essa operação não tem um corpo de solicitação.

Respostas

Se a operação for bem-sucedida, o serviço reenviará uma resposta `200 OK` HTTP.

Sintaxe da resposta

```
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"
      }
}
```

Cabeçalhos de resposta

Essa operação usa somente os cabeçalhos de resposta que são comuns a maioria das respostas. Para informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Corpo da resposta

O corpo da resposta contém os seguintes campos JSON.

Tags da

As tags anexadas ao cofre. Cada tag é composta de uma chave e de um valor.

Tipo: mapa de string para string

Obrigatório: Sim

Erros

Para obter informações sobre exceções e mensagens de erro do Amazon Glacier, consulte Respostas de erro.

Exemplos

Exemplo: listar tags de um cofre

O exemplo a seguir lista as tags de um cofre.

Exemplo de solicitação

Neste exemplo, uma solicitação GET é enviada para recuperar uma lista de tags do cofre especificado.

```
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
```

Exemplo de resposta

Se a solicitação foi bem-sucedida, o Amazon Glacier retorna um `HTTP 200 OK` com uma lista de tags para o cofre, conforme mostrado no exemplo a seguir.

```
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"
      }
}
```

Seções relacionadas

- Add Tags To Vault (POST tags add)

- Remove Tags From Vault (POST tags remove)

Consulte também

Para obter mais informações sobre como usar essa API em um dos Amazon SDKs específicos de linguagem, consulte o seguinte:

- AWS Command Line Interface

List Vaults (GET vaults)

Descrição

Essa operação lista todos os cofres próprios chamando a conta do usuário. A lista retornada na resposta é classificada por nome de cofre ASCII.

Por padrão, essa operação retorna até 10 itens por solicitação. Se houver mais cofres a serem listados, o campo `marker` no corpo da resposta conterá o Amazon Resource Name (ARN – Nome de recurso da Amazon) do cofre no qual deve continuar a lista com uma nova solicitação List Vaults; do contrário, o campo `marker` será `null`. Na próxima solicitação List Vaults, defina o parâmetro `marker` como o valor retornado pelo Amazon Glacier nas respostas à solicitação List Vaults anterior. Você também pode limitar o número de cofres retornados na resposta especificando o parâmetro `limit` na solicitação.

Solicitações

Para obter uma lista de cofres, envie uma solicitação `GET` para o recurso cofres.

Sintaxe

```
GET /AccountId/vaults HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: Date
Authorization: SignatureValue
```

```
x-amz-glacier-version: 2012-06-01
```

> **Note**
>
> O valor de `AccountId` é o ID da Conta da AWS. Esse valor deve corresponder ao ID de Conta da AWS associado às credenciais usadas para assinar a solicitação. Você pode especificar um ID da Conta da AWS ou, de opcionalmente, um '-' (hífen) único. Nesse caso, o Amazon Glacier usa o ID de Conta da AWS associado às credenciais usadas para assinar a solicitação. Se você especificar o ID da conta, não inclua nenhum hífen ('-') no ID.

Parâmetros da solicitação

Esta operação usa os parâmetros de solicitação a seguir.

Nome	Descrição	Obrigatório
`limit`	O número máximo de cofres a serem retornados. O limite padrão é 10. O número de cofres retornados pode ser menor que o limite especificado, mas o número de cofres retornados jamais excede o limite. Tipo: string Restrições: valor do inteiro mínimo de 1. Valor do inteiro máximo de 10.	Não
`marker`	Uma string usada na paginação. `marker` especifica o ARN do cofre com o qual a listagem de cofres deve começar. (O cofre especificado por `marker` não está incluído na lista retornada.) Obtenha o valor `marker` de uma resposta List Vaults anterior. Você precisará incluir o `marker` somente se continuar a paginação dos resultados iniciada em uma solicitação List Vaults anterior. Especificar um valor vazio ("") para o marcador retorna uma lista de cofres a partir do primeiro cofre.	Não

Nome	Descrição	Obrigatório
	Tipo: string Restrições: nenhuma	

Cabeçalhos de solicitação

Essa operação usa somente os cabeçalhos de solicitação que são comuns a todas as operações. Para informações sobre os cabeçalhos de solicitação comuns, consulte Cabeçalhos de solicitação comuns.

Corpo da Solicitação

Essa operação não tem um corpo de solicitação.

Respostas

Sintaxe

```
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
   },
   ...
  ]
}
```

Cabeçalhos de resposta

Essa operação usa somente os cabeçalhos de resposta que são comuns a maioria das respostas. Para informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Corpo da resposta

O corpo da resposta contém os seguintes campos JSON.

CreationDate

A data em que o cofre foi criado, em Coordinated Universal Time (UTC – Tempo universal coordenado).

Tipo: string Uma representação de string no formato de data ISO 8601, por exemplo `2013-03-20T17:03:43.221Z`.

LastInventoryDate

A data do inventário de cofre mais recente, em UTC. Esse campo poderá ser nulo se um inventário ainda não tiver sido executado no cofre; por exemplo, se você tiver acabado de criar o cofre. Para obter mais informações sobre como iniciar um inventário para um cofre, consulte Initiate Job (trabalhos POST).

Tipo: uma representação de string no formato de data ISO 8601, por exemplo `2013-03-20T17:03:43.221Z`.

Marker

O `vaultARN` que representa onde continuar a paginação dos resultados. Você usa o `marker` em outra solicitação List Vaults para obter mais cofres na lista. Se não houver cofres, esse valor será `null`.

Tipo: string

NumberOfArchives

O número de arquivos no cofre desde a data do último inventário de cofre.

Tipo: número

SizeInBytes

O tamanho total, em bytes, de todos os arquivos no cofre, inclusive eventuais custos indiretos por arquivo, desde a data do último inventário.

Tipo: número

VaultARN

O nome de recurso da Amazon (ARN) do cofre.

Tipo: string

VaultList

Uma matriz de objetos, com cada objeto apresentando uma descrição de um cofre.

Tipo: matriz

VaultName

O nome do cofre.

Tipo: string

Erros

Para obter informações sobre exceções e mensagens de erro do Amazon Glacier, consulte Respostas de erro.

Exemplos

Exemplo: List All Vaults

O exemplo a seguir lista os cofres. Como os parâmetros `marker` e `limit` não são especificados na solicitação, até 10 cofres são retornados.

Exemplo de solicitação

```
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
```

Exemplo de resposta

O `Marker` é `null`, o que indica não haver mais cofres a serem listados.

```
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"
   }
  ]
}
```

Exemplo: Partial List of Vaults

O exemplo a seguir retorna dois cofres começando pelo cofre especificado pelo `marker`.

Exemplo de solicitação

```
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
```

Exemplo de resposta

Dois cofres são retornados na lista. O `Marker` contém o ARN do cofre para continuar a paginação em outra solicitação List Vaults.

```
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"
   }
  ]
}
```

Seções relacionadas

- Create Vault (PUT vault)
- Delete Vault (DELETE vault)
- Initiate Job (trabalhos POST)
- Gerenciamento de identidade e acesso para o Amazon Glacier

Consulte também

Para obter mais informações sobre como usar essa API em um dos Amazon SDKs específicos de linguagem, consulte o seguinte:

- AWS Command Line Interface

Remove Tags From Vault (POST tags remove)

Essa operação remove uma ou mais tags do conjunto de tags anexado a um cofre. Para obter mais informações sobre tags, consulte Marcação de recursos do Amazon Glacier.

Essa operação é idempotente. A operação será bem-sucedida, mesmo se não houver tags anexadas ao cofre.

Sintaxe da Solicitação

Para remover tags de um cofre, envie uma solicitação HTTP POST para o URI das tags conforme mostrado no exemplo da sintaxe a seguir.

```
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**
>
> O valor de `AccountId` é o ID da Conta da AWS. Esse valor deve corresponder ao ID de Conta da AWS associado às credenciais usadas para assinar a solicitação. Você pode especificar um ID da Conta da AWS ou, de opcionalmente, um '-' (hífen) único. Nesse caso, o Amazon Glacier usa o ID de Conta da AWS associado às credenciais usadas para assinar a solicitação. Se você especificar o ID da conta, não inclua nenhum hífen ('-') no ID.

Parâmetros da solicitação

Nome	Descrição	Obrigatório
`operation=remove`	Um parâmetro de string de consulta única `operation` com um valor de `remove` para diferenciá-lo de Add Tags To Vault (POST tags add).	Sim

Cabeçalhos de solicitação

Essa operação usa somente os cabeçalhos de solicitação que são comuns a todas as operações. Para informações sobre os cabeçalhos de solicitação comuns, consulte Cabeçalhos de solicitação comuns.

Corpo da Solicitação

O corpo da solicitação contém os campos JSON a seguir.

TagKeys

Uma lista de chaves de tag. Toda tag correspondente é removida do cofre.

Tipo: matriz de strings

Restrição de tamanho: mínimo de 1 item na lista. Máximo de 10 itens na lista.

Obrigatório: Sim

Respostas

Se a ação for bem-sucedida, o serviço reenviará uma resposta `204 No Content` HTTP com um corpo HTTP vazio.

Sintaxe

```
HTTP/1.1 204 No Content
x-amzn-RequestId: x-amzn-RequestId
Date: Date
```

Cabeçalhos de resposta

Essa operação usa somente os cabeçalhos de resposta que são comuns a maioria das respostas. Para informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Corpo da resposta

Essa operação não retorna um corpo de resposta.

Erros

Para obter informações sobre exceções e mensagens de erro do Amazon Glacier, consulte Respostas de erro.

Exemplos

Exemplo de solicitação

O exemplo a seguir envia uma solicitação HTTP POST para remover as tags especificadas.

```
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"
   ]
}
```

Exemplo de resposta

Se a solicitação for bem-sucedida, o Amazon Glacier retornará `HTTP 204 No Content`, conforme mostrado no exemplo a seguir.

```
HTTP/1.1 204 No Content
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:02:00 GMT
```

Seções relacionadas

- Add Tags To Vault (POST tags add)
- Listar tags para cofre (GET tags)

Consulte também

Para obter mais informações sobre como usar essa API em um dos Amazon SDKs específicos de linguagem, consulte o seguinte:

- AWS Command Line Interface

Set Vault Access Policy (PUT access-policy)

Description

Esta operação configura uma política de acesso para um cofre e substituirá uma política existente. Para configurar uma política de acesso, envie uma solicitação PUT para o sub-recurso `access-policy` do cofre. Você pode definir uma política de acesso por cofre, e a política pode ter até 20 KB. Para obter mais informações sobre políticas de acesso ao cofre, consulte Políticas de acesso ao cofre.

Solicitações

Sintaxe

Para definir uma política de acesso ao cofre, envie uma solicitação PUT HTTP para o URI do sub-recurso `access-policy` do cofre conforme mostrado no exemplo da sintaxe a seguir.

```
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**
>
> O `AccountId` valor é o Conta da AWS ID da conta proprietária do cofre. Você pode especificar um Conta da AWS ID ou, opcionalmente, um único '-' (hífen). Nesse caso, o Amazon Glacier usa Conta da AWS o ID associado às credenciais usadas para assinar a solicitação. Se você usar um ID de conta, não inclua nenhum hífen ('-') no ID.

Parâmetros da solicitação

Essa operação não usa parâmetros de solicitação.

Cabeçalhos de solicitação

Essa operação usa somente os cabeçalhos de solicitação que são comuns a todas as operações. Para informações sobre os cabeçalhos de solicitação comuns, consulte Cabeçalhos de solicitação comuns.

Corpo da Solicitação

O corpo da solicitação contém os campos JSON a seguir.

Política

A política de acesso ao cofre como uma string JSON, que usa"\" como um caractere de escape.

Tipo: string

Obrigatório: Sim

Respostas

Em resposta, o Amazon Glacier retornará `204 No Content`, se a política for aceita.

Sintaxe

```
HTTP/1.1 204 No Content
x-amzn-RequestId: x-amzn-RequestId
Date: Date
```

Cabeçalhos de resposta

Essa operação usa somente os cabeçalhos de resposta que são comuns a maioria das respostas. Para informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Corpo da resposta

Essa operação não retorna um corpo de resposta.

Erros

Para obter informações sobre exceções e mensagens de erro do Amazon Glacier, consulte Respostas de erro.

Exemplos

Exemplo de solicitação

O exemplo a seguir envia uma solicitação PUT HTTP para criar o URI do sub-recurso `access-policy` do cofre. A string JSON `Policy` usa "\" como um caractere de 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\"}]}"}
```

Exemplo de resposta

Se a solicitação for bem-sucedida, o Amazon Glacier retornará `HTTP 204 No Content`, conforme mostrado no exemplo a seguir.

```
HTTP/1.1 204 No Content
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:02:00 GMT
```

Seções relacionadas

- Política de acesso de exclusão do cofre (DELETE access-policy)
- Get Vault Access Policy (GET access-policy)

Consulte também

Para obter mais informações sobre o uso dessa API em uma das linguagens específicas da Amazon SDKs, consulte o seguinte:

- AWS Command Line Interface

Definir configuração de notificação de cofre (PUT notification-configuration)

Description

Recuperar um arquivo e um inventário de cofre são operações assíncronas no Amazon Glacier para as quais você deve primeiro iniciar um trabalho e aguardar que ele seja concluído para poder fazer download da saída do trabalho. Você pode configurar um cofre para publicar uma mensagem em um tópico do Amazon Simple Notification Service (Amazon SNS) quando esses trabalhos forem concluídos. Você pode usar essa operação para definir a configuração de notificação no cofre. Para obter mais informações, consulte Configurar notificações do cofre no Amazon Glacier.

Para configurar notificações de cofre, envie uma solicitação PUT para o sub-recurso `notification-configuration` do cofre. Uma configuração de notificação é específica de um cofre; por isso, ela também é conhecida como um sub-recurso de cofre. A solicitação deve incluir um documento JSON que forneça um tópico do Amazon Simple Notification Service (Amazon SNS) e os eventos para os quais você deseja que o Amazon Glacier envie notificações ao tópico.

Você pode configurar um cofre a fim de publicar uma notificação para os seguintes eventos de cofre:

- **`ArchiveRetrievalCompleted`**: este evento ocorre quando um trabalho que tenha sido iniciado para uma recuperação de arquivo é concluído (Initiate Job (trabalhos POST)). O status do trabalho concluído pode ser `Succeeded` ou `Failed`. A notificação enviada para o tópico do SNS é a mesma saída retornada de Trabalho de descrição (GET JobID).
- **`InventoryRetrievalCompleted`**: este evento ocorre quando um trabalho que tenha sido iniciado para uma recuperação de inventário é concluído (Initiate Job (trabalhos POST)). O status do trabalho concluído pode ser `Succeeded` ou `Failed`. A notificação enviada para o tópico do SNS é a mesma saída retornada de Trabalho de descrição (GET JobID).

Os tópicos do Amazon SNS devem conceder permissão ao cofre para poder publicar notificações no tópico.

Solicitações

Para definir a configuração de notificação no cofre, envie uma solicitação PUT para o URI do sub-recurso `notification-configuration` do cofre. Você especifica a configuração no corpo da solicitação. A configuração inclui o nome do tópico do Amazon SNS e um conjunto de eventos que disparam uma notificação para cada tópico.

Sintaxe

```
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**
>
> O `AccountId` valor é o Conta da AWS ID da conta proprietária do cofre. Você pode especificar um Conta da AWS ID ou, opcionalmente, um único '-' (hífen). Nesse caso, o Amazon Glacier usa Conta da AWS o ID associado às credenciais usadas para assinar a solicitação. Se você usar um ID de conta, não inclua nenhum hífen ('-') no ID.

Parâmetros da solicitação

Essa operação não usa parâmetros de solicitação.

Cabeçalhos de solicitação

Essa operação usa somente os cabeçalhos de solicitação que são comuns a todas as operações. Para informações sobre os cabeçalhos de solicitação comuns, consulte Cabeçalhos de solicitação comuns.

Corpo da Solicitação

O JSON no corpo da solicitação contém os campos a seguir.

Eventos

Uma matriz de um ou mais eventos para os quais você deseja que o Amazon Glacier envie uma notificação.

Valores válidos: `ArchiveRetrievalCompleted` | `InventoryRetrievalCompleted`

Obrigatório: sim

Tipo: matriz

SNSTopic

O ARN do tópico do Amazon SNS. Para obter mais informações, vá para Conceitos básicos do Amazon SNS no Guia de conceitos básicos do Amazon Simple Notification Service.

Obrigatório: sim

Tipo: string

Respostas

Em resposta, o Amazon Glacier retornará `204 No Content`, se a configuração de notificação for aceita.

Sintaxe

```
HTTP/1.1 204 No Content
x-amzn-RequestId: x-amzn-RequestId
Date: Date
```

Cabeçalhos de resposta

Essa operação usa somente os cabeçalhos de solicitação que são comuns a todas as operações. Para informações sobre os cabeçalhos de solicitação comuns, consulte Cabeçalhos de solicitação comuns.

Corpo da resposta

Essa operação não retorna um corpo de resposta.

Erros

Para obter informações sobre exceções e mensagens de erro do Amazon Glacier, consulte Respostas de erro.

Exemplos

O exemplo a seguir demonstra como configurar uma notificação de cofre.

Exemplo de solicitação

A solicitação a seguir define a configuração de notificação `examplevault`, de maneira que notificações para dois eventos (`ArchiveRetrievalCompleted` e `InventoryRetrievalCompleted`) são enviadas para o tópico `arn:aws:sns:us-west-2:012345678901:mytopic` do Amazon SNS.

```
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"
}
```

Exemplo de resposta

Uma resposta bem-sucedida retorna um `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
```

Seções relacionadas

- Get Vault Notifications (GET notification-configuration)

- Delete Vault Notifications (DELETE notification-configuration)
- Gerenciamento de identidade e acesso para o Amazon Glacier

Consulte também

Para obter mais informações sobre o uso dessa API em uma das linguagens específicas da Amazon SDKs, consulte o seguinte:

- AWS Command Line Interface

Operações de arquivo

Estas são as operações de arquivo disponíveis para serem usadas no Amazon Glacier.

Tópicos

- Delete Archive (DELETE archive)
- Upload Archive (POST archive)

Delete Archive (DELETE archive)

Descrição

Essa operação exclui um arquivo de um cofre. Você pode excluir um arquivo por vez de um cofre. Para excluir o arquivo, você deve fornecer o ID de arquivo na solicitação de exclusão. Você pode obter o ID de arquivo fazendo download do inventário do cofre que contém o arquivo. Para obter mais informações sobre como fazer download do inventário de cofre, consulte Fazer download de um inventário de cofre no Amazon Glacier.

Depois de excluir um arquivo, você ainda poderá fazer uma solicitação bem-sucedida para iniciar um trabalho a fim de recuperar o arquivo excluído, mas o trabalho de recuperação de arquivo falhará.

Recuperações de arquivo que estejam em andamento para um ID de arquivo quando você exclui o arquivo podem ser bem-sucedidas ou não, de acordo com os seguintes cenários:

- Se o trabalho de recuperação de arquivo estiver preparando ativamente os dados para download quando o Amazon Glacier receber a solicitação de exclusão de arquivo, a operação de recuperação de arquivo poderá falhar.

- Se o trabalho de recuperação de arquivo tiver preparado com êxito o arquivo para download quando o Amazon Glacier receber a solicitação de exclusão de arquivo, você poderá fazer download da saída.

Para obter mais informações sobre recuperação de arquivo, consulte Baixando um arquivo no Amazon Glacier.

Essa operação é idempotente. Tentar excluir um arquivo já excluído não resulta em um erro.

Solicitações

Para excluir um arquivo, você envia uma solicitação `DELETE` para o URI do recurso.

Sintaxe

```
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**
>
> O valor de `AccountId` é o ID de Conta da AWS da conta proprietária do cofre. Você pode especificar um ID da Conta da AWS ou, de opcionalmente, um '-' (hífen) único. Nesse caso, o Amazon Glacier usa o ID de Conta da AWS associado às credenciais usadas para assinar a solicitação. Se você usar um ID de conta, não inclua nenhum hífen ('-') no ID.

Parâmetros da solicitação

Essa operação não usa parâmetros de solicitação.

Cabeçalhos de solicitação

Essa operação usa somente os cabeçalhos de solicitação que são comuns a todas as operações. Para informações sobre os cabeçalhos de solicitação comuns, consulte Cabeçalhos de solicitação comuns.

Corpo da Solicitação

Essa operação não tem um corpo de solicitação.

Respostas

Sintaxe

```
HTTP/1.1 204 No Content
x-amzn-RequestId: x-amzn-RequestId
Date: Date
```

Cabeçalhos de resposta

Essa operação usa somente os cabeçalhos de resposta que são comuns a maioria das respostas. Para informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Corpo da resposta

Essa operação não retorna um corpo de resposta.

Erros

Para obter informações sobre exceções e mensagens de erro do Amazon Glacier, consulte Respostas de erro.

Exemplos

O exemplo a seguir demonstra como excluir um arquivo do cofre chamado `examplevault`.

Exemplo de solicitação

O ID do arquivo a ser excluído é especificado como um sub-recurso de `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
```

Exemplo de resposta

Se a solicitação for bem-sucedida, o Amazon Glacier responderá com `204 No Content` para indicar que o arquivo é excluído.

```
HTTP/1.1 204 No Content
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:00:00 GMT
```

Seções relacionadas

- Initiate Multipart Upload (POST multipart-uploads)
- Upload Archive (POST archive)
- Gerenciamento de identidade e acesso para o Amazon Glacier

Upload Archive (POST archive)

Descrição

Essa operação adiciona um arquivo a um cofre. Para um upload bem-sucedido, os dados são mantidos de maneira durável. Em resposta, o Amazon Glacier retorna o ID de arquivo no cabeçalho `x-amz-archive-id` da resposta. Você deve salvar o ID do arquivo retornado, de maneira que possa acessar o arquivo depois.

Você deve fornecer um hash de árvore SHA256 dos dados cujo upload está fazendo. Para obter informações sobre como computar um hash de árvore SHA256, consulte Computar somas de verificação.

> **Note**
>
> O hash da árvore SHA256 só é necessário para a ação Upload Archive (POST archive) ao usar a API. Não é necessário ao usar o AWS CLI.

Ao fazer upload de um arquivo, você pode especificar uma descrição de arquivo de até 1.024 caracteres ASCII imprimíveis. O Amazon Glacier retorna a descrição do arquivo quando você recupera o arquivo ou obtém o inventário de cofre. O Amazon Glacier não interpreta a descrição de maneira alguma. Uma descrição de arquivo não precisa ser exclusiva. Você não pode usar a descrição para recuperar ou classificar a lista de arquivos.

Exceto para a descrição de arquivo opcional, o Amazon Glacier não dá suporte a metadados adicionais dos arquivos. O ID de arquivo é uma sequência de caracteres invisível da qual você não pode inferir nenhum significado sobre o arquivo. Assim, convém manter metadados sobre os arquivos no lado do cliente. Para obter mais informações, consulte Trabalhar com arquivos no Amazon Glacier.

Os arquivos são imutáveis. Depois de fazer upload de um arquivo, você não poderá editar o arquivo nem a descrição.

Solicitações

Para fazer upload de um arquivo, use o método HTTP POST e delimite a solicitação ao sub-recurso `archives` do cofre no qual você deseja salvar o arquivo. A solicitação deve incluir o tamanho da carga útil do arquivo, a soma de verificação (hash de árvore SHA256) e pode incluir uma descrição do arquivo.

Sintaxe

```
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**
>
> O valor de `AccountId` é o ID de Conta da AWS da conta proprietária do cofre. Você pode especificar um ID da Conta da AWS ou, de opcionalmente, um '-' (hífen) único. Nesse caso,

o Amazon Glacier usa o ID de Conta da AWS associado às credenciais usadas para assinar a solicitação. Se você usar um ID de conta, não inclua nenhum hífen ('-') no ID.

Parâmetros da solicitação

Essa implementação da operação não usa parâmetros de solicitação.

Cabeçalhos de solicitação

Essa operação usa os seguintes cabeçalhos de solicitação, além dos cabeçalhos de solicitação que são comuns a todas as operações. Para mais informações sobre os cabeçalhos de solicitação comuns, consulte Cabeçalhos de solicitação comuns.

Nome	Descrição	Obrigatório
`Content-Length`	O tamanho do objeto em bytes. Para obter mais informações, acesse http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.13. Tipo: número Padrão: nenhum Restrições: nenhuma	Sim
`x-amz-archive-description`	A descrição opcional do arquivo cujo upload você está fazendo. Ela pode ser uma descrição em linguagem simples ou algum identificador que você opta por atribuir. A descrição não precisa ser exclusiva entre os arquivos. Quando você recupera um inventário de cofre (consulte Initiate Job (trabalhos POST)), ela inclui essa descrição para cada um dos arquivos retornados em resposta. Tipo: string Padrão: nenhum	Não

Nome	Descrição	Obrigatório
	Restrições: a descrição deve ser menor que ou igual a 1.024 bytes caracteres. Os caracteres permitidos são ASCII 7 bits sem códigos de controle – mais especificamente, valores ASCII 32—126 decimais ou 0x20—0x7E hexadecimais.	
`x-amz-content-sha256`	A soma de verificação SHA256 (um hash linear) da carga útil. Ele não é o mesmo valor especificado por você no cabeçalho `x-amz-sha256-tree-hash` . Tipo: string Padrão: nenhum Restrições: nenhuma	Sim
`x-amz-sha256-tree-hash`	A soma de verificação computada pelo usuário, hash de árvore SHA256, da carga útil. Para obter informações sobre como computar o hash de árvore SHA256, consulte Computar somas de verificação. Se calcular uma soma de verificação diferente da carga útil, o Amazon Glacier rejeitará a solicitação. Tipo: string Padrão: nenhum Restrições: nenhuma	Sim

Corpo da Solicitação

O corpo da solicitação contém os dados cujo upload deve ser feito.

Respostas

Em resposta, o Amazon Glacier armazena de maneira durável o arquivo e retorna um caminho de URI para o ID de arquivo.

Sintaxe

```
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
```

Cabeçalhos de resposta

Uma resposta bem-sucedida inclui os seguintes cabeçalhos de resposta, além dos cabeçalhos de resposta que são comuns a todas as operações. Para mais informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Nome	Descrição
`Location`	O caminho do URI relativo do recurso de arquivo recém-adicionado. Tipo: string
`x-amz-archive-id`	O ID do arquivo. Esse valor também está incluído como parte do cabeçalho `Location`. Tipo: string
`x-amz-sha256-tree-hash`	A soma de verificação do arquivo computada pelo Amazon Glacier. Tipo: string

Corpo da resposta

Essa operação não retorna um corpo de resposta.

Erros

Para obter informações sobre exceções e mensagens de erro do Amazon Glacier, consulte Respostas de erro.

Exemplos

Exemplo de solicitação

O exemplo a seguir mostra uma solicitação para fazer upload de um arquivo.

```
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).>
```

Exemplo de resposta

A resposta bem-sucedida abaixo tem um cabeçalho `Location` onde você pode obter o ID atribuído pelo Amazon Glacier ao arquivo.

```
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
```

Seções relacionadas

- Trabalhar com arquivos no Amazon Glacier
- Fazer upload de arquivos grandes em partes (Multipart Upload)

- Delete Archive (DELETE archive)
- Gerenciamento de identidade e acesso para o Amazon Glacier

Operações de multipart upload

Estas são as operações de upload multiparte disponíveis para serem usadas no Amazon Glacier.

Tópicos

- Abort Multipart Upload (DELETE uploadID)
- Complete Multipart Upload (POST uploadID)
- Initiate Multipart Upload (POST multipart-uploads)
- List Parts (GET uploadID)
- List Multipart Uploads (GET multipart-uploads)
- Upload Part (PUT uploadID)

Abort Multipart Upload (DELETE uploadID)

Description

Esta operação de multipart upload interrompe um multipart upload identificado pelo ID de upload.

Depois que a solicitação Abort Multipart Upload for bem-sucedida, você não poderá usar o ID de upload para fazer upload de mais partes ou realizar outras operações. A interrupção de um multipart upload concluído falha. No entanto, a interrupção de um upload já anulado será bem-sucedida, por um curto período.

Essa operação é idempotente.

Para obter informações sobre o multipart upload, consulte Fazer upload de arquivos grandes em partes (Multipart Upload).

Solicitações

Para interromper um multipart upload, envie uma solicitação `DELETE` HTTP para o URI do sub-recurso `multipart-uploads` do cofre e identifique o ID de multipart upload específico como parte do URI.

Sintaxe

```
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**
>
> O `AccountId` valor é o Conta da AWS ID da conta proprietária do cofre. Você pode especificar um Conta da AWS ID ou, opcionalmente, um único '-' (hífen). Nesse caso, o Amazon Glacier usa Conta da AWS o ID associado às credenciais usadas para assinar a solicitação. Se você usar um ID de conta, não inclua nenhum hífen ('-') no ID.

Parâmetros da solicitação

Essa operação não usa parâmetros de solicitação.

Cabeçalhos de solicitação

Essa operação usa somente os cabeçalhos de solicitação que são comuns a todas as operações. Para informações sobre os cabeçalhos de solicitação comuns, consulte Cabeçalhos de solicitação comuns.

Corpo da Solicitação

Essa operação não tem um corpo de solicitação.

Respostas

Sintaxe

```
HTTP/1.1 204 No Content
x-amzn-RequestId: x-amzn-RequestId
Date: Date
```

Cabeçalhos de resposta

Essa operação usa somente os cabeçalhos de resposta que são comuns a maioria das respostas. Para informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Corpo da resposta

Essa operação não retorna um corpo de resposta.

Erros

Para obter informações sobre exceções e mensagens de erro do Amazon Glacier, consulte Respostas de erro.

Exemplo

Exemplo de solicitação

No exemplo a seguir, uma solicitação DELETE é enviada para o URI de um recurso de ID de multipart upload.

```
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
```

Exemplo de resposta

```
HTTP/1.1 204 No Content
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:00:00 GMT
```

Seções relacionadas

- Initiate Multipart Upload (POST multipart-uploads)

- Upload Part (PUT uploadID)
- Complete Multipart Upload (POST uploadID)
- List Multipart Uploads (GET multipart-uploads)
- List Parts (GET uploadID)
- Fazer upload de arquivos grandes em partes (Multipart Upload)
- Gerenciamento de identidade e acesso para o Amazon Glacier

Complete Multipart Upload (POST uploadID)

Descrição

Você chama essa operação upload multiparte para informar ao Amazon Glacier que o upload de todas as partes do arquivo foi feito e que o Amazon Glacier agora pode montar o arquivo com base nas partes carregadas.

Para obter informações sobre o multipart upload, consulte Fazer upload de arquivos grandes em partes (Multipart Upload).

Depois da montagem e da gravação do arquivo no cofre, o Amazon Glacier retornará o ID de arquivo do recurso de arquivo recém-criado. Depois de fazer upload de um arquivo, você deve salvar o ID do arquivo retornado para recuperar o arquivo mais tarde.

Na solicitação, você deve incluir o hash de árvore SHA256 computado de todo o arquivo cujo upload foi feito. Para obter informações sobre como computar um hash de árvore SHA256, consulte Computar somas de verificação. No lado do servidor, o Amazon Glacier também cria o hash de árvore SHA256 do arquivo montado. Se os valores forem correspondentes, o Amazon Glacier salvará o arquivo no cofre; do contrário, ele retornará um erro e a operação falhará. A operação List Parts (GET uploadID) retorna uma lista de partes cujo upload foi feito para um multipart upload específico. Ela inclui informações da soma de verificação de cada parte carregada que podem ser usadas para depurar um problema de soma de verificação inválida.

Além disso, o Amazon Glacier também verifica se há intervalos de conteúdo não encontrados. Ao fazer upload de partes, você especifica valores de intervalo identificando onde cada parte fica na montagem final do arquivo. Ao montar o arquivo final, o Amazon Glacier verifica se há algum intervalo de conteúdo não encontrado e, se houver algum intervalo de conteúdo não encontrado, o Amazon Glacier retornará um erro e a operação Upload Multiparte Completo falhará.

Complete Multipart Upload é uma operação idempotente. Depois do primeiro complete multipart upload bem-sucedido, se você chamar a operação novamente dentro de um curto período, a operação será bem-sucedida e retornará o mesmo ID do arquivo. Isso será útil se você enfrentar um problema de rede ou receber um erro de servidor 500, quando poderá repetir a solicitação Complete Multipart Upload e obter o mesmo ID de arquivo sem criar arquivos duplicados. No entanto, depois que o multipart upload for concluído, você não poderá chamar a operação List Parts, e o multipart upload não será exibido na resposta List Multipart Uploads, mesmo se for possível uma conclusão idempotente.

Solicitações

Para concluir um upload multiparte, você envia uma solicitação HTTP POST para o URI do ID de upload criado pelo Amazon Glacier em resposta à solicitação Iniciar Upload Multiparte. Esse é o mesmo URI usado por você ao fazer upload das partes. Além dos cabeçalhos obrigatórios comuns, você deve incluir o resultado do hash de árvore SHA256 de todo o arquivo e o tamanho total do arquivo em bytes.

Sintaxe

```
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**
>
> O valor de `AccountId` é o ID de Conta da AWS da conta proprietária do cofre. Você pode especificar um ID da Conta da AWS ou, de opcionalmente, um '-' (hífen) único. Nesse caso, o Amazon Glacier usa o ID de Conta da AWS associado às credenciais usadas para assinar a solicitação. Se você usar um ID de conta, não inclua nenhum hífen ('-') no ID.

Parâmetros da solicitação

Essa operação não usa parâmetros de solicitação.

Cabeçalhos de solicitação

Essa operação usa os seguintes cabeçalhos de solicitação, além dos cabeçalhos de solicitação que são comuns a todas as operações. Para mais informações sobre os cabeçalhos de solicitação comuns, consulte Cabeçalhos de solicitação comuns.

Nome	Descrição	Obrigatório
`x-amz-archive-size`	O tamanho total, em bytes, de todo o arquivo. Esse valor deve ser a soma de todos os tamanhos das partes individuais cujo upload você fez. Tipo: string Padrão: nenhum Restrições: nenhuma	Sim
`x-amz-sha256-tree-hash`	O hash de árvore SHA256 de todo o arquivo. Trata-se do hash de árvore SHA256 das partes individuais. Se o valor especificado por você na solicitação não corresponder ao hash de árvore SHA256 do arquivo montado final computado pelo Amazon Glacier, o Amazon Glacier retornará um erro e a solicitação falhará. Tipo: string Padrão: nenhum Restrições: nenhuma	Sim

Elementos da solicitação

Essa operação não usa elementos de solicitação.

Respostas

O Amazon Glacier cria um hash de árvore SHA256 de todo o arquivo. Se o valor corresponder ao hash de árvore SHA256 de todo o arquivo especificado por você na solicitação, o Amazon Glacier adicionará o arquivo ao cofre. Em resposta, ele retorna o cabeçalho `Location` HTTP com o caminho de URL do recurso de arquivo recém-adicionado. Se o tamanho do arquivo ou SHA256 enviado por você na solicitação não for correspondente, o Amazon Glacier retornará um erro e o upload permanecerá no estado incompleto. Será possível tentar novamente a operação Complete Multipart Upload depois com valores corretos, quando você poderá criar um arquivo com êxito. Se um upload multiparte não for concluído, o Amazon Glacier acabará recuperando o ID do upload.

Sintaxe

```
HTTP/1.1 201 Created
x-amzn-RequestId: x-amzn-RequestId
Date: Date
Location: Location
x-amz-archive-id: ArchiveId
```

Cabeçalhos de resposta

Uma resposta bem-sucedida inclui os seguintes cabeçalhos de resposta, além dos cabeçalhos de resposta que são comuns a todas as operações. Para mais informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Nome	Descrição
`Location`	O caminho do URI relativo do arquivo recém-criado. Esse URL inclui o ID de arquivo gerado pelo Amazon Glacier. Tipo: string
`x-amz-archive-id`	O ID do arquivo. Esse valor também está incluído como parte do cabeçalho `Location`. Tipo: string

Campos de resposta

Essa operação não retorna um corpo de resposta.

Exemplo

Exemplo de solicitação

Neste exemplo, uma solicitação HTTP POST é enviada para o URI que foi retornado por uma solicitação Initiate Multipart Upload. A solicitação especifica o hash de árvore SHA256 de todo o arquivo e o tamanho total do arquivo.

```
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
```

Exemplo de resposta

A resposta de exemplo a seguir mostra que o Amazon Glacier criou com êxito um arquivo das partes cujo upload você fez. A resposta inclui o ID de arquivo com caminho completo.

```
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
```

Agora você pode enviar solicitações HTTP para o URI do recurso/arquivo recém-adicionado. Por exemplo, você pode enviar uma solicitação GET para recuperar o arquivo.

Seções relacionadas

- Initiate Multipart Upload (POST multipart-uploads)
- Upload Part (PUT uploadID)
- Abort Multipart Upload (DELETE uploadID)
- List Multipart Uploads (GET multipart-uploads)
- List Parts (GET uploadID)
- Fazer upload de arquivos grandes em partes (Multipart Upload)
- Delete Archive (DELETE archive)
- Gerenciamento de identidade e acesso para o Amazon Glacier

Initiate Multipart Upload (POST multipart-uploads)

Description

Essa operação inicia um multipart upload (consulte Fazer upload de arquivos grandes em partes (Multipart Upload)). O Amazon Glacier cria um recurso de upload multiparte e retorna o ID na resposta. Você usará esse ID de upload em operações de multipart upload subsequentes.

Ao iniciar um multipart upload, você especifica o tamanho da parte em número de bytes. O tamanho da parte deve ser um mebibyte (MiB) (1024 kibibytes [KiB]) multiplicado por uma potência de 2 — por exemplo, 1048576 (1 MiB), 2097152 (2 MiB), 4194304 (4 MiB), 8388608 (8 MiB) e assim por diante. O tamanho da parte mínimo permitido é 1 MiB, e o máximo é 4 gibibytes (GiB).

Toda parte cujo upload você faz usando esse ID de upload, exceto a última, deve ter o mesmo tamanho. A última pode ser do mesmo tamanho ou menor. Por exemplo, suponhamos que você queira fazer upload de um arquivo de 16,2 MiB. Se iniciar o multipart upload com um tamanho de parte de 4 MiB, você fará upload de quatro partes de 4 MiB cada e uma parte de 0,2 MiB.

> **Note**
>
> Você não precisa saber o tamanho do arquivo ao iniciar um upload multiparte porque o Amazon Glacier não exige que especifique o tamanho de todo o arquivo.

Depois de concluir o upload multiparte, o Amazon Glacier removerá o recurso de upload multiparte referenciado pelo ID. O Amazon Glacier também removerá o recurso de upload multiparte se você cancelar o upload multiparte, ou ele poderá ser removido se não houver atividade por um período

de 24 horas. O ID ainda poderá estar disponível depois de 24 horas, mas os aplicativos não devem esperar esse comportamento.

Solicitações

Para iniciar um multipart upload, você envia uma solicitação HTTP `POST` para o URI do sub-recurso `multipart-uploads` do cofre no qual deseja salvar o arquivo. A solicitação deve incluir o tamanho da parte e pode incluir uma descrição do arquivo.

Sintaxe

```
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**
>
> O `AccountId` valor é o Conta da AWS ID da conta proprietária do cofre. Você pode especificar um Conta da AWS ID ou, opcionalmente, um único '-' (hífen). Nesse caso, o Amazon Glacier usa Conta da AWS o ID associado às credenciais usadas para assinar a solicitação. Se você usar um ID de conta, não inclua nenhum hífen ('-') no ID.

Parâmetros da solicitação

Essa operação não usa parâmetros de solicitação.

Cabeçalhos de solicitação

Essa operação usa os seguintes cabeçalhos de solicitação, além dos cabeçalhos de solicitação que são comuns a todas as operações. Para mais informações sobre os cabeçalhos de solicitação comuns, consulte Cabeçalhos de solicitação comuns.

Name (Nome)	Description	Obrigatório
`x-amz-part-size`		Sim

Name (Nome)	Description	Obrigatório
	O tamanho de cada parte, exceto a última, em bytes. A última parte pode ser menor que o tamanho dessa parte. Tipo: string Padrão: nenhum Restrições: o tamanho da parte deve ser um mebibyte (1024 KiB) multiplicado por uma potência de 2 — por exemplo, 1048576 (1 MiB), 2097152 (2 MiB), 4194304 (4 MiB), 8388608 (8 MiB) e assim por diante. O tamanho da parte mínimo permitido é 1 MiB, e o máximo é 4 GiB (4096 MiB).	

Name (Nome)	Description	Obrigatório
`x-amz-archive-description`	Descrição do arquivo cujo upload você está fazendo em partes. Ela pode ser uma descrição em linguagem simples ou algum identificador exclusivo que você opta por atribuir. Quando você recupera um inventário de cofre (consulte Initiate Job (trabalhos POST)), o inventário inclui essa descrição para cada um dos arquivos retornados em resposta. Os espaços à esquerda em descrições de arquivo são removidos. Tipo: string Padrão: nenhum Restrições: a descrição deve ser menor que ou igual a 1.024 bytes. Os caracteres permitidos são ASCII 7 bits sem códigos de controle, mais especificamente valores ASCII 32-126 decimais ou 0x20-0x7E hexadecimais.	Não

Corpo da Solicitação

Essa operação não tem um corpo de solicitação.

Respostas

Na resposta, o Amazon Glacier cria um recurso de upload multiparte identificado por um ID e retorna o caminho do URI relativo do ID de upload multiparte.

Sintaxe

```
HTTP/1.1 201 Created
x-amzn-RequestId: x-amzn-RequestId
Date: Date
```

```
Location: Location
x-amz-multipart-upload-id: multiPartUploadId
```

Cabeçalhos de resposta

Uma resposta bem-sucedida inclui os seguintes cabeçalhos de resposta, além dos cabeçalhos de resposta que são comuns a todas as operações. Para mais informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Name (Nome)	Description
`Location`	O caminho do URI relativo do ID de upload multiparte criado pelo Amazon Glacier. Você pode usar esse caminho do URI para delimitar as solicitações a fim de fazer upload de partes e concluir o multipart upload. Tipo: string
`x-amz-multipart-upload-id`	O ID do multipart upload. Esse valor também está incluído como parte do cabeçalho `Location`. Tipo: string

Corpo da resposta

Essa operação não retorna um corpo de resposta.

Erros

Para obter informações sobre exceções e mensagens de erro do Amazon Glacier, consulte Respostas de erro.

Exemplo

Exemplo de solicitação

O exemplo a seguir inicia um multipart upload enviando uma solicitação HTTP `POST` para o URI do sub-recurso `multipart-uploads` de um cofre chamado `examplevault`. A solicitação inclui

cabeçalhos para especificar o tamanho da parte de 4 MiB (4.194.304 bytes) e a descrição de arquivo opcional.

```
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
```

Exemplo de resposta

O Amazon Glacier cria um recurso de upload multiparte e o adiciona ao sub-recurso `multipart-uploads` do cofre. O cabeçalho de resposta `Location` inclui o caminho do URI relativo para o ID de multipart upload.

```
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
```

Para obter informações sobre como fazer upload de partes individuais, consulte Upload Part (PUT uploadID).

Seções relacionadas

- Upload Part (PUT uploadID)
- Complete Multipart Upload (POST uploadID)
- Abort Multipart Upload (DELETE uploadID)
- List Multipart Uploads (GET multipart-uploads)
- List Parts (GET uploadID)

- Delete Archive (DELETE archive)
- Fazer upload de arquivos grandes em partes (Multipart Upload)
- Gerenciamento de identidade e acesso para o Amazon Glacier

List Parts (GET uploadID)

Description

Essa operação multipart upload lista as partes de um arquivo cujo upload foi feito em um multipart upload específico identificado por um ID de upload. Para obter informações sobre o multipart upload, consulte Fazer upload de arquivos grandes em partes (Multipart Upload).

Você pode fazer essa solicitação a qualquer momento durante um multipart upload em andamento antes de concluir o multipart upload. O Amazon Glacier retorna a lista de partes classificada pelo intervalo especificado em cada upload da parte. Se você enviar uma solicitação List Parts depois de concluir o upload multiparte, o Amazon Glacier retornará um erro.

A operação List Parts dá suporte à paginação. Você deve sempre verificar o campo `Marker` no corpo da resposta para ver se há um marcador no qual continuar a lista. Se não houver mais itens, o campo `marker` será `null`. Se o `marker` não for nulo, para obter o próximo conjunto de partes, envie outra solicitação List Parts com o parâmetro `marker` da solicitação definido como o valor do marcador retornado pelo Amazon Glacier em resposta à solicitação List Parts anterior.

Você também pode limitar o número de partes retornadas na resposta especificando o parâmetro `limit` na solicitação.

Solicitações

Sintaxe

Para listar as partes de um multipart upload em andamento, você envia uma solicitação `GET` para o URI do recurso de ID de multipart upload. O ID de multipart upload é retornado quando você inicia um multipart upload (Initiate Multipart Upload (POST multipart-uploads)). Você também pode especificar parâmetros `marker` e `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**
>
> O `AccountId` valor é o Conta da AWS ID da conta proprietária do cofre. Você pode especificar um Conta da AWS ID ou, opcionalmente, um único '-' (hífen). Nesse caso, o Amazon Glacier usa Conta da AWS o ID associado às credenciais usadas para assinar a solicitação. Se você usar um ID de conta, não inclua nenhum hífen ('-') no ID.

Parâmetros da solicitação

Name (Nome)	Description	Obrigatório
`limit`	O número máximo de partes a serem retornadas. O limite padrão é 50. O número de partes retornadas pode ser menor que o limite especificado, mas o número de partes retornadas jamais excede o limite. Tipo: string Restrições: valor do inteiro mínimo de 1. Valor do inteiro máximo de 50.	Não
`marker`	Uma string invisível usada na paginação. `marker` especifica a parte na qual a listagem de partes deve começar. Obtenha o valor `marker` da resposta de uma resposta List Parts anterior. Você somente precisará incluir o `marker` se continuar a paginação dos resultados iniciada em uma solicitação List Parts anterior. Tipo: string Restrições: nenhuma	Não

Cabeçalhos de solicitação

Essa operação usa somente os cabeçalhos de resposta que são comuns a maioria das respostas. Para informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Corpo da Solicitação

Essa operação não tem um corpo de solicitação.

Respostas

Sintaxe

```
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
}
```

Cabeçalhos de resposta

Essa operação usa somente os cabeçalhos de resposta que são comuns a maioria das respostas. Para informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Corpo da resposta

O corpo da resposta contém os seguintes campos JSON.

ArchiveDescription

A descrição do arquivo que tiver sido especificada na solicitação Initiate Multipart Upload. Este campo será `null` se nenhuma descrição de arquivo tiver sido especificada na operação Initiate Multipart Upload.

Tipo: string

CreationDate

A hora UTC em que o multipart upload foi iniciado.

Tipo: string Uma representação de string no formato de data ISO 8601, por exemplo `2013-03-20T17:03:43.221Z`.

Marker

Uma string invisível que representa onde continuar a paginação dos resultados. Você usa o `marker` em uma nova solicitação List Parts para obter mais trabalhos na lista. Se não houver partes, esse valor será `null`.

Tipo: string

MultipartUploadId

O ID do upload ao qual as partes estão associadas.

Tipo: string

PartSizeInBytes

O tamanho da parte em bytes. Trata-se do mesmo valor especificado por você na solicitação Initiate Multipart Upload.

Tipo: número

Partes

A lista dos tamanhos de partes do multipart upload. Cada objeto na matriz contém um `RangeBytes` e um par de nome/valor `sha256-tree-hash`.

Tipo: matriz

RangeInBytes

O intervalo de bytes de uma parte, inclusive do valor máximo do intervalo.

Tipo: string

SHA256TreeHash

O valor de hash da SHA256 árvore que o Amazon Glacier calculou para a peça. Esse campo jamais será `null`.

Tipo: string

VaultARN

O nome do recurso da Amazon (ARN) do cofre para o qual o multipart upload foi iniciado.

Tipo: string

Erros

Para obter informações sobre exceções e mensagens de erro do Amazon Glacier, consulte Respostas de erro.

Exemplos

Exemplo: List Parts de um multipart upload

O exemplo a seguir lista todas as partes de um upload. O exemplo envia uma solicitação `GET` HTTP para o URI do ID de multipart upload específico de um multipart upload em andamento e retorna até 1.000 partes.

Exemplo de solicitação

```
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
```

Exemplo de resposta

Na resposta, o Amazon Glacier retorna uma lista de partes cujo upload foi feito associadas ao ID de upload multiparte especificado. Neste exemplo, existem apenas duas partes. O campo `Marker` é `null`, o que indica não haver mais partes do multipart upload.

```
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"
}
```

Exemplo: List Parts de um multipart upload (especificar o marcador e limitar os parâmetros de solicitação)

O exemplo a seguir demonstra como usar a paginação para obter um número limitado de resultados. O exemplo envia uma solicitação GET HTTP para o URI do ID de multipart upload específico de um multipart upload em andamento para retornar uma parte. Um parâmetro `marker` de partida

especifica em qual parte a lista de partes deve ser iniciada. Você pode obter o valor `marker` da resposta de uma solicitação anterior para uma lista de partes. Além disso, neste exemplo, o parâmetro `limit` é definido como 1 e retorna uma parte. O campo `Marker` não é `null`, o que indica haver pelo menos mais uma parte a ser obtida.

Exemplo de solicitação

```
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
```

Exemplo de resposta

Na resposta, o Amazon Glacier retorna uma lista de partes cujo upload foi feito associadas ao ID de upload multiparte em andamento especificado.

```
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"
```

```
}
```

Seções relacionadas

- Initiate Multipart Upload (POST multipart-uploads)
- Upload Part (PUT uploadID)
- Complete Multipart Upload (POST uploadID)
- Abort Multipart Upload (DELETE uploadID)
- List Multipart Uploads (GET multipart-uploads)
- Fazer upload de arquivos grandes em partes (Multipart Upload)
- Gerenciamento de identidade e acesso para o Amazon Glacier

List Multipart Uploads (GET multipart-uploads)

Description

Esta operação de multipart upload lista multipart uploads em andamento para o cofre. Multipart upload em andamento é um multipart upload que foi iniciado por uma solicitação Initiate Multipart Upload (POST multipart-uploads), mas que ainda não foi concluído ou foi interrompido. A lista retornada na resposta List Multipart Upload não tem ordem garantida.

A operação List Multipart Uploads dá suporte à paginação. Por padrão, essa operação retorna até 50 multipart uploads na resposta. Você deve sempre verificar o campo `marker` no corpo da resposta para um marcador no qual continuar a lista; se não houver mais itens, o campo `marker` será `null`.

Se o `marker` não for nulo, para obter o próximo conjunto de uploads multiparte, envie outra solicitação Listar Uploads Multiparte com o parâmetro da solicitação `marker` definido como o valor do marcador retornado pelo Amazon Glacier em resposta à solicitação Listar Uploads Multiparte.

Observe a diferença entre essa operação e a operação List Parts (GET uploadID). A operação List Multipart Uploads lista todos os multipart uploads para um cofre. A operação List Parts retorna partes de um multipart upload específico identificado por um ID de upload.

Para obter informações sobre o multipart upload, consulte Fazer upload de arquivos grandes em partes (Multipart Upload).

Solicitações

Sintaxe

Para listar multipart uploads, envie uma solicitação `GET` para o URI do sub-recurso `multipart-uploads` do cofre. Você também pode especificar parâmetros `marker` e `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**
>
> O `AccountId` valor é o Conta da AWS ID da conta proprietária do cofre. Você pode especificar um Conta da AWS ID ou, opcionalmente, um único '-' (hífen). Nesse caso, o Amazon Glacier usa Conta da AWS o ID associado às credenciais usadas para assinar a solicitação. Se você usar um ID de conta, não inclua nenhum hífen ('-') no ID.

Parâmetros da solicitação

Name (Nome)	Description	Obrigatório
`limit`	Especifica o número máximo de uploads retornados no corpo da resposta. Se não for especificado, a operação List Uploads retornará até 50 uploads. Tipo: string Restrições: valor do inteiro mínimo de 1. Valor do inteiro máximo de 50.	Não
`marker`	Uma string invisível usada na paginação. `marker` especifica o upload no qual a listagem de uploads deve começar. Obtenha	Não

Name (Nome)	Description	Obrigatório
	o valor `marker` de uma resposta List Uploads anterior. Você somente precisará incluir o `marker` se continuar a paginação dos resultados iniciada em uma solicitação List Uploads anterior. Tipo: string Restrições: nenhuma	

Cabeçalhos de solicitação

Essa operação usa somente os cabeçalhos de resposta que são comuns a maioria das respostas. Para informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Corpo da Solicitação

Essa operação não tem um corpo de solicitação.

Respostas

Sintaxe

```
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
```

```
    },
    ...
  ]
}
```

Cabeçalhos de resposta

Essa operação usa somente os cabeçalhos de resposta que são comuns a maioria das respostas. Para informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Corpo da resposta

O corpo da resposta contém os seguintes campos JSON.

ArchiveDescription

A descrição do arquivo que tiver sido especificada na solicitação Initiate Multipart Upload. Este campo será `null` se nenhuma descrição de arquivo tiver sido especificada na operação Initiate Multipart Upload.

Tipo: string

CreationDate

A hora UTC em que o multipart upload foi iniciado.

Tipo: string Uma representação de string no formato de data ISO 8601, por exemplo `2013-03-20T17:03:43.221Z`.

Marker

Uma string invisível que representa onde continuar a paginação dos resultados. Você usa o `marker` em uma nova solicitação List Multipart Uploads para obter mais uploads na lista. Se não houver uploads, esse valor será `null`.

Tipo: string

PartSizeInBytes

O tamanho da parte especificado na solicitação Initiate Multipart Upload (POST multipart-uploads). Este é o tamanho de todas as partes no upload, exceto a última parte, que pode ser menor do que esse tamanho.

Tipo: número

MultipartUploadId

O ID do multipart upload.

Tipo: string

UploadsList

Uma lista de metadados sobre objetos de multipart upload. Cada item na lista contém um conjunto de pares nome/valor para o upload correspondente, inclusive `ArchiveDescription`, `CreationDate`, `MultipartUploadId`, `PartSizeInBytes` e `VaultARN`.

Tipo: matriz

VaultARN

O Amazon Resource Name (ARN – Nome de recurso da Amazon) do cofre que contém o arquivo.

Tipo: string

Erros

Para obter informações sobre exceções e mensagens de erro do Amazon Glacier, consulte Respostas de erro.

Exemplos

Exemplo: listar todos os multipart uploads

O exemplo a seguir lista todos os multipart uploads em andamento para o cofre. O exemplo mostra uma solicitação `GET` HTTP para o URI do sub-recurso `multipart-uploads` de um cofre especificado. Como os parâmetros `marker` e `limit` não são especificados na solicitação, até 1.000 multipart uploads em andamento são retornados.

Exemplo de solicitação

```
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
```

Exemplo de resposta

Na resposta, o Amazon Glacier retorna uma lista de todos os uploads multiparte em andamento para o cofre especificado. O campo `marker` é `null`, o que indica que não há mais uploads a serem listados.

```
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"
    }
```

```
  ]
}
```

Exemplo: lista parcial de multipart uploads

O exemplo a seguir demonstra como usar a paginação para obter um número limitado de resultados. O exemplo mostra uma solicitação GET HTTP para o URI do sub-recurso `multipart-uploads` de um cofre especificado. Neste exemplo, o parâmetro `limit` é definido como 1, o que significa que somente um upload é retornado na lista, e o parâmetro `marker` indica o ID de multipart upload no qual a lista retornada começa.

Exemplo de solicitação

```
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
```

Exemplo de resposta

Na resposta, o Amazon Glacier retorna uma lista de não mais do que dois uploads multiparte em andamento para o cofre especificado, começando pelo marcador especificado e retornando dois resultados.

```
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"
    }
  ]
}
```

Seções relacionadas

- Initiate Multipart Upload (POST multipart-uploads)
- Upload Part (PUT uploadID)
- Complete Multipart Upload (POST uploadID)
- Abort Multipart Upload (DELETE uploadID)
- List Parts (GET uploadID)
- Fazer upload de arquivos grandes em partes (Multipart Upload)
- Gerenciamento de identidade e acesso para o Amazon Glacier

Upload Part (PUT uploadID)

Descrição

Esta operação de multipart upload faz upload de uma parte de um arquivo. Você pode fazer upload de partes de arquivo em qualquer ordem porque, na solicitação Upload Part, especifica o intervalo de bytes no arquivo montado cujo upload será feito nessa parte. Você também pode fazer upload dessas partes em paralelo. Você pode fazer upload de até 10.000 partes de um multipart upload.

Para obter informações sobre o multipart upload, consulte Fazer upload de arquivos grandes em partes (Multipart Upload).

O Amazon Glacier rejeita a parte do upload se qualquer uma das seguintes condições for verdadeira:

- Hash de árvore SHA256 não correspondente: para garantir que dados da parte não sejam corrompidos na transmissão, você computa um hash de árvore SHA256 da parte e o inclui na solicitação. Mediante o recebimento dos dados da parte, o Amazon Glacier também computa um

hash de árvore SHA256. Se os dois valores de hash não forem correspondentes, a operação vai falhar. Para obter informações sobre como computar um hash de árvore SHA256, consulte Computar somas de verificação.

- Hash linear SHA256 não correspondente: obrigatório para autorização, você computa o hash linear SHA256 de toda a carga útil cujo upload foi feito e o inclui na solicitação. Para obter informações sobre como computar um hash linear SHA256, consulte Computar somas de verificação.
- Tamanho da parte não correspondente: o tamanho de cada parte, exceto a última, deve corresponder ao tamanho especificado na solicitação Initiate Multipart Upload (POST multipart-uploads) correspondente. O tamanho da última parte deve ser igual a ou menor que o tamanho especificado.

> **Note**
>
> Se fizer upload de uma parte cujo tamanho seja menor que o tamanho especificado por você na solicitação de multipart upload de iniciação e que a parte não seja a última, a solicitação upload part será bem-sucedida. No entanto, a solicitação Complete Multipart Upload subsequente falhará.

- Intervalo não alinhado: o valor do intervalo de bytes na solicitação não se alinha ao tamanho da parte especificado na solicitação de iniciação correspondente. Por exemplo, se você especificar um tamanho de parte de 4.194.304 bytes (4 MB), então, de 0 a 4.194.303 bytes (4 MB —1) e de 4.194.304 (4 MB) a 8.388.607 (8 MB —1) serão intervalos de partes válidos. No entanto, se você definir um valor de intervalo de 2 MB a 6 MB, o intervalo não se alinhará ao tamanho da parte, e o upload falhará.

Essa operação é idempotente. Se você fizer upload da mesma parte várias vezes, os dados incluídos na solicitação mais recente substituirão os dados cujo upload foi feito anteriormente.

Solicitações

Você envia essa solicitação PUT HTTP para o URI do ID de upload retornado pela solicitação Initiate Multipart Upload. O Amazon Glacier usa o ID de upload para associar uploads de partes a um upload multiparte específico. A solicitação deve incluir um hash de árvore SHA256 dos dados da parte (cabeçalho `x-amz-SHA256-tree-hash`), um hash linear SHA256 de toda a carga útil (cabeçalho `x-amz-content-sha256`), o intervalo de bytes (cabeçalho `Content-Range`) e o tamanho da parte em bytes (cabeçalho `Content-Length`).

Sintaxe

```
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**
>
> O valor de `AccountId` é o ID de Conta da AWS da conta proprietária do cofre. Você pode especificar um ID da Conta da AWS ou, de opcionalmente, um '-' (hífen) único. Nesse caso, o Amazon Glacier usa o ID de Conta da AWS associado às credenciais usadas para assinar a solicitação. Se você usar um ID de conta, não inclua nenhum hífen ('-') no ID.

Parâmetros da solicitação

Essa operação não usa parâmetros de solicitação.

Cabeçalhos de solicitação

Essa operação usa os seguintes cabeçalhos de solicitação, além dos cabeçalhos de solicitação que são comuns a todas as operações. Para mais informações sobre os cabeçalhos de solicitação comuns, consulte Cabeçalhos de solicitação comuns.

Nome	Descrição	Obrigatório
`Content-Length`	Identifica o tamanho da parte em bytes. Tipo: string Padrão: nenhum Restrições: nenhuma	Não

Nome	Descrição	Obrigatório
`Content-Range`	Identifica o intervalo de bytes no arquivo montado cujo upload será feito nessa parte. O Amazon Glacier usa essas informações para montar o arquivo na sequência apropriada. O formato desse cabeçalho segue RFC 2616. Um cabeçalho de exemplo é `Content-Range:bytes 0-4194303/*` . Tipo: string Padrão: nenhum Restrições: o intervalo não pode ser maior que o tamanho da parte especificado por você quando iniciou o multipart upload.	Sim
`x-amz-content-sha256`	A soma de verificação SHA256 (um hash linear) da carga útil cujo upload foi feito. Ele não é o mesmo valor especificado por você no cabeçalho `x-amz-sha256-tree-hash` . Tipo: string Padrão: nenhum Restrições: nenhuma	Sim

Nome	Descrição	Obrigatório
`x-amz-sha256-tree-hash`	Especifica um hash de árvore SHA256 dos dados cujo upload está sendo feito. Para obter informaçõ es sobre como computar um hash de árvore SHA256, consulte Computar somas de verificação. Tipo: string Padrão: nenhum Restrições: nenhuma	Sim

Corpo da Solicitação

O corpo da solicitação contém os dados cujo upload deve ser feito.

Respostas

Mediante um upload da parte bem-sucedido, o Amazon Glacier retorna uma resposta `204 No Content`.

Sintaxe

```
HTTP/1.1 204 No Content
x-amzn-RequestId: x-amzn-RequestId
Date: Date
x-amz-sha256-tree-hash: ChecksumComputedByAmazonGlacier
```

Cabeçalhos de resposta

Uma resposta bem-sucedida inclui os seguintes cabeçalhos de resposta, além dos cabeçalhos de resposta que são comuns a todas as operações. Para mais informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Nome	Descrição
`x-amz-sha256-tree-hash`	

Nome	Descrição
	O hash de árvore SHA256 computado pelo Amazon Glacier para a parte cujo upload foi feito. Tipo: string

Corpo da resposta

Essa operação não retorna um corpo de resposta.

Exemplo

A solicitação a seguir faz upload de uma parte de 4 MB. A solicitação define o intervalo de bytes para fazer dessa a primeira parte no arquivo.

Exemplo de solicitação

O exemplo envia uma solicitação PUT HTTP para fazer upload de uma parte de 4 MB. A solicitação é enviada para o URI do ID de upload retornado pela solicitação Initiate Multipart Upload. O cabeçalho `Content-Range` identifica a parte como a primeira parte dos 4 MB de dados do arquivo.

```
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
```

Para fazer upload da próxima parte, o procedimento é o mesmo; no entanto, você deve calcular um novo hash de árvore SHA256 da parte cujo upload está sendo feito e também especificar um novo intervalo de bytes onde a parte entrará no conjunto final. A solicitação a seguir faz upload de outra parte usando o mesmo ID de upload. A solicitação especifica os próximos 4 MB do arquivo após a solicitação anterior e um tamanho da parte de 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
```

O upload das partes pode ser feito em qualquer ordem; o Amazon Glacier usa a especificação do intervalo para cada parte a fim de determinar a ordem na qual montá-las.

Exemplo de resposta

```
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
```

Seções relacionadas

- Initiate Multipart Upload (POST multipart-uploads)
- Upload Part (PUT uploadID)
- Complete Multipart Upload (POST uploadID)
- Abort Multipart Upload (DELETE uploadID)
- List Multipart Uploads (GET multipart-uploads)
- List Parts (GET uploadID)
- Fazer upload de arquivos grandes em partes (Multipart Upload)
- Gerenciamento de identidade e acesso para o Amazon Glacier

Operações de trabalho

Estas são as operações de trabalho disponíveis para serem usadas no Amazon Glacier.

Tópicos

- Trabalho de descrição (GET JobID)
- Get Job Output (GET output)
- Initiate Job (trabalhos POST)
- List Jobs (GET jobs)

Trabalho de descrição (GET JobID)

Description

Esta operação retorna informações sobre um trabalho iniciado anteriormente, incluindo a data de início do trabalho, o usuário que o iniciou, o código de status do trabalho/mensagem e o tópico do Amazon Simple Notification Service (Amazon SNS) para notificar depois que o Amazon Glacier concluir o trabalho. Para obter mais informações sobre como iniciar um trabalho, consulte Initiate Job (trabalhos POST).

> **Note**
>
> Essa operação permite verificar o status do trabalho. No entanto, é altamente recomendável configurar um tópico do Amazon SNS e especificá-lo na solicitação do trabalho de inicialização para que o Amazon Glacier possa notificar o tópico após a conclusão do trabalho.

Um ID do trabalho não vai expirar por pelo menos 24 horas depois que o Amazon Glacier concluir o trabalho.

Solicitações

Sintaxe

Para obter informações sobre um trabalho, você usa o método GET HTTP e delimita a solicitação ao trabalho específico. O caminho do URI relativo é o mesmo retornado pelo Amazon Glacier quando você iniciou o trabalho.

```
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**
>
> O `AccountId` valor é o Conta da AWS ID da conta proprietária do cofre. Você pode especificar um Conta da AWS ID ou, opcionalmente, um único '-' (hífen). Nesse caso, o Amazon Glacier usa Conta da AWS o ID associado às credenciais usadas para assinar a solicitação. Se você usar um ID de conta, não inclua nenhum hífen ('-') no ID.

> **Note**
>
> Na solicitação, se você omitir o `JobID`, a resposta retornará uma lista de todos os trabalhos ativos no cofre especificado. Para obter mais informações sobre como listar trabalhos, consulte List Jobs (GET jobs).

Parâmetros da solicitação

Essa operação não usa parâmetros de solicitação.

Cabeçalhos de solicitação

Essa operação usa somente os cabeçalhos de solicitação que são comuns a todas as operações. Para informações sobre os cabeçalhos de solicitação comuns, consulte Cabeçalhos de solicitação comuns.

Corpo da Solicitação

Essa operação não tem um corpo de solicitação.

Respostas

Sintaxe

```
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"
}
```

Cabeçalhos de resposta

Essa operação usa somente os cabeçalhos de resposta que são comuns a maioria das respostas. Para informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Corpo da resposta

O corpo da resposta contém os seguintes campos JSON.

Ação

O tipo de trabalho. É `ArchiveRetrieval`, `InventoryRetrieval` ou `Select`.

Tipo: string

ArchiveId

O ID de arquivo solicitado para um trabalho de seleção ou de recuperação de arquivo. Do contrário, esse campo será `null`.

Tipo: string

ArquivoSHA256TreeHash

O hash da SHA256 árvore de todo o arquivo para um trabalho de recuperação de arquivamento. Para trabalhos de recuperação do inventário, esse campo é `null`.

Tipo: string

ArchiveSizeInBytes

Para um trabalho `ArchiveRetrieval`, trata-se do tamanho em bytes do arquivo solicitado para download. Para o trabalho `InventoryRetrieval`, o valor é `null`.

Tipo: número

Completed

O status do trabalho. Quando um trabalho de recuperação de arquivo ou inventário for concluído, você obterá a saída do trabalho usando o Get Job Output (GET output).

Tipo: booliano

CompletionDate

A hora UTC (Horário Coordenado Universal) em que a solicitação de trabalho foi concluída. Enquanto o trabalho estiver em andamento, o valor será nulo.

Tipo: string

CreationDate

A hora UTC em que o trabalho foi criado.

Tipo: uma representação de string no formato de data ISO 8601, por exemplo `2013-03-20T17:03:43.221Z`.

InventoryRetrievalParameters

Os parâmetros de entrada usados em uma recuperação de inventário de intervalo.

Tipo: InventoryRetrievalJobInput objeto

InventorySizeInBytes

Para um trabalho `InventoryRetrieval`, trata-se do tamanho em bytes do inventário solicitado para download. No trabalho `ArchiveRetrieval` ou `Select`, o valor é `null`.

Tipo: número

JobDescription

A descrição do trabalho fornecida por você quando iniciou o trabalho.

Tipo: string

JobId

O ID que identifica o trabalho no Amazon Glacier.

Tipo: string

JobOutputPath

Contém o local de saída do trabalho.

Tipo: string

OutputLocation

Um objeto com informações sobre o local onde os resultados e os erros do trabalho de seleção são armazenados.

Tipo: OutputLocation objeto

RetrievalByteRange

O intervalo de bytes recuperados para trabalhos de recuperação de arquivamento no formato "*StartByteValue*-". *EndByteValue* Se você não especificar um intervalo na recuperação do arquivo, o arquivo inteiro será recuperado; também é StartByteValueigual a 0 e é igual ao tamanho do arquivo EndByteValuemenos 1. Em trabalhos de seleção ou de recuperação de inventário, este campo é `null`.

Tipo: string

SelectParameters

Um objeto com informações sobre os parâmetros usados em uma seleção.

Tipo: SelectParameters objeto

SHA256TreeHash

O valor do hash da SHA256 árvore para o intervalo solicitado de um arquivo. Se a solicitação Initiate Job (trabalhos POST) para um arquivo tiver especificado um intervalo alinhado ao hash de árvore, esse campo retornará um valor. Para obter mais informações sobre o alinhamento ao hash de árvore para recuperações do intervalo de arquivos, consulte Receber somas de verificação durante o download de dados.

Para o caso específico quando todo o arquivo for recuperado, esse valor será igual ao valor `ArchiveSHA256TreeHash`.

Esse campo é `null` nas seguintes situações:

- Os trabalhos de recuperação de arquivo que especificam um intervalo não alinhado ao hash de árvore.

- Trabalhos de arquivamento que especificam um intervalo igual a todo o arquivamento e cujo status do trabalho seja `InProgress`.
- Trabalhos de inventário.
- Trabalhos de seleção.

Tipo: string

SNSTopic

Um tópico do Amazon SNS que recebe notificação.

Tipo: string

StatusCode

O código que indica o status do trabalho.

Valores válidos: `InProgress` | `Succeeded` | `Failed`

Tipo: string

StatusMessage

Uma mensagem amigável que descreve o status do trabalho.

Tipo: string

Nível

O nível de acesso a dados a ser usado na seleção ou na recuperação de arquivo.

Valores válidos: `Bulk` | `Expedited` | `Standard`

Tipo: string

VaultARN

O nome de recurso da Amazon (ARN) do cofre do qual o trabalho é um sub-recurso.

Tipo: string

Erros

Para obter informações sobre exceções e mensagens de erro do Amazon Glacier, consulte Respostas de erro.

Exemplos

O exemplo a seguir mostra a solicitação de um trabalho que recupera um arquivo.

Solicitação de exemplo: obter descrição do trabalho

```
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
```

Exemplo de resposta

O corpo da resposta inclui JSON que descreve o trabalho especificado. Observe que, nos trabalhos de recuperação de inventário e de arquivo, os campos JSON são iguais. No entanto, quando um campo não se aplica ao tipo de trabalho, o valor é `null`. Esta é uma resposta de exemplo para um trabalho de recuperação de arquivo. Observe o seguinte:

- O valor do campo `Action` é `ArchiveRetrieval`.
- O campo `ArchiveSizeInBytes` mostra o tamanho do arquivo solicitado no trabalho de recuperação do arquivo.
- O `ArchiveSHA256TreeHash` campo mostra o hash da SHA256 árvore de todo o arquivo.
- O campo `RetrievalByteRange` mostra o intervalo solicitado no pedido Initiate Job. Neste exemplo, todo o arquivo é solicitado.
- O `SHA256TreeHash` campo mostra o hash da SHA256 árvore para o intervalo solicitado na solicitação de Initiate Job. Neste exemplo, ele é igual ao campo `ArchiveSHA256TreeHash`, o que significa que todo o arquivo foi solicitado.
- O valor do campo `InventorySizeInBytes` é `null` porque não se aplica a um trabalho de recuperação de arquivo.

```
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"
}
```

Esta é uma resposta de exemplo para um trabalho de recuperação de inventário. Observe o seguinte:

- O valor do campo `Action` é `InventoryRetrieval`.
- Os valores de campo `ArchiveSizeInBytes`, `ArchiveSHA256TreeHash` e `RetrievalByteRange` são nulos porque não se aplicam a um trabalho de recuperação de inventário.
- O valor de campo `InventorySizeInBytes` é `null` porque o trabalho ainda está em andamento e não preparou totalmente o inventário para download. Se o trabalho tivesse sido concluído antes da solicitação do trabalho de descrição, esse campo informaria o tamanho do resultado.

```
{
   "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"
}
```

Esta é uma resposta de exemplo para um trabalho de recuperação de inventário completo que contém um marcador usado para continuar a paginação da recuperação do inventário de cofre.

```
{
    "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
    },
}
```

Seções relacionadas

- Get Job Output (GET output)
- Gerenciamento de identidade e acesso para o Amazon Glacier

Get Job Output (GET output)

Description

Essa operação faz download da saída do trabalho iniciado por você usando a operação Initiate Job (trabalhos POST). Dependendo do tipo de trabalho especificado por você quando iniciou o trabalho, a saída será o conteúdo de um arquivo ou um inventário de cofre.

Você pode fazer download de toda a saída do ou de uma parte da saída especificando um intervalo de bytes. Para trabalhos de recuperação de arquivo e inventário, você deve verificar o tamanho obtido por download em relação ao tamanho retornado nos cabeçalhos da resposta Obter saída de trabalho.

Para trabalhos de recuperação de arquivo, você também deve verificar se o tamanho é o esperado. Se você fizer download de uma parte da saída, o tamanho esperado se baseará no intervalo de bytes especificado. Por exemplo, se especificar um intervalo de `bytes=0-1048575`, você deverá verificar se o tamanho do download é de 1.048.576 bytes. Se você fizer download de um arquivo inteiro, o tamanho esperado será o tamanho do arquivo quando fez upload dele no Amazon Glacier. O tamanho esperado também é retornado nos cabeçalhos da resposta Obter saída de trabalho.

Em caso de um trabalho de recuperação de arquivo, dependendo do intervalo de bytes especificado por você, o Amazon Glacier retorna a soma de verificação da parte dos dados. Para garantir que a parte obtida por download seja de dados corretos, compute a soma de verificação no cliente, verifique se os valores são correspondentes e se o tamanho é o esperado.

Um ID do trabalho não vai expirar por pelo menos 24 horas depois que o Amazon Glacier concluir o trabalho. Ou seja, você pode fazer download do resultado do trabalho dentro do período de 24 horas depois que o Amazon Glacier concluir o trabalho.

Solicitações

Sintaxe

Para recuperar uma saída de trabalho, você envia a solicitação `GET` HTTP para o URI do `output` do trabalho específico.

```
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**
>
> O `AccountId` valor é o Conta da AWS ID da conta proprietária do cofre. Você pode especificar um Conta da AWS ID ou, opcionalmente, um único '-' (hífen). Nesse caso, o Amazon Glacier usa Conta da AWS o ID associado às credenciais usadas para assinar a solicitação. Se você usar um ID de conta, não inclua nenhum hífen ('-') no ID.

Parâmetros da solicitação

Essa operação não usa parâmetros de solicitação.

Cabeçalhos de solicitação

Essa operação usa os seguintes cabeçalhos de solicitação, além dos cabeçalhos de solicitação que são comuns a todas as operações. Para mais informações sobre os cabeçalhos de solicitação comuns, consulte Cabeçalhos de solicitação comuns.

Name (Nome)	Description	Obrigatório
`Range`		Não

Name (Nome)	Description	Obrigatório
	O intervalo de bytes a ser recuperado da saída. Por exemplo, se você quiser fazer download dos primeiros 1.048.576 bytes, especifique o intervalo como `bytes=0-1048575` . Para obter mais informações, acesse Definição do campo de cabeçalho do intervalo. O intervalo se refere a qualquer intervalo especificado na solicitação Initiate Job. Por padrão, essa operação faz download de toda a saída. Se a saída do trabalho for grande, você poderá usar o cabeçalho `Range` para recuperar uma parte da saída. Isso permite fazer download de toda a saída em blocos de bytes menores. Por exemplo, suponhamos que você tenha 1 GB de saída de trabalho cujo download deseja fazer e opte por fazer download de blocos de 128 MB de dados por vez, em um total de oito solicitações Get Job Output. Você usará o seguinte processo para fazer download da saída do trabalho: 1. Faça download de uma saída de bloco de 128 MB de dados especificando o intervalo de bytes apropriado com o cabeçalho `Range`. Verifique se todos os 128 MB de dados foram recebidos. 2. Com os dados, a resposta incluirá uma soma de verificação da carga útil. Você computa a soma de verificação da carga útil no cliente e a compara com a soma de verificação recebida na resposta para garantir que tenha recebido todos os dados esperados. 3. Repita as etapas 1 e 2 para todos os oito blocos de 128 MB de dados de saída, sempre especificando o intervalo de bytes apropriado. 4. Depois do download de todas as partes da saída do trabalho, você terá uma lista de oito valores de soma de verificação. Compute o hash de árvore desses valores para encontrar a soma de verificação de toda a saída. Usando a operação Trabalho de descrição (GET JobID), obtenha	

Name (Nome)	Description	Obrigatório
	informações do trabalho que forneceu a saída. A resposta inclui a soma de verificação de todo o arquivo armazenado no Amazon Glacier. Você compara esse valor com a soma de verificação computada para garantir que tenha feito download de todo o conteúdo do arquivo sem erros. Tipo: string Padrão: nenhum Restrições: nenhuma	

Corpo da Solicitação

Essa operação não tem um corpo de solicitação.

Respostas

Sintaxe

Para uma solicitação de recuperação que retorne todos os dados do trabalho, a resposta da saída do trabalho retorna um código de resposta `200 OK`. Quando o conteúdo parcial é solicitado, por exemplo, se você tiver especificado o cabeçalho `Range` na solicitação, o código de resposta `206 Partial Content` será retornado.

```
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.]
```

Cabeçalhos de resposta

Cabeçalho	Description
`Content-Range`	O intervalo de bytes retornado pelo Amazon Glacier. Se somente o download da saída parcial tiver sido feito, a resposta fornecerá o intervalo de bytes retornado pelo Amazon Glacier. Por exemplo, `bytes 0-1048575/8388608` retorna o primeiro 1 MB de 8 MB. Para obter mais informações sobre o cabeçalho `Content-Range` , acesse Definição do campo de cabeçalho Content-Range. Tipo: string
`Content-Type`	O Content-Type depende da saída do trabalho ser um arquivo ou um inventário de cofre. • Para dados de arquivo, o Content-Type é `application/octet-stream` . • Para inventário de cofre, se você tiver solicitado o formato CSV quando iniciou o trabalho, o Content-Type será `text/csv`. Do contrário, por padrão, o inventário de cofre é retornado como JSON, e o Content-Type será `application/json` . Tipo: string
`x-amz-sha256-tree-hash`	A soma de verificação dos dados na resposta. Esse cabeçalho é retornado somente quando se recupera a saída de um trabalho de recuperação de arquivo. Além disso, esse cabeçalho é exibido quando o intervalo de dados recuperados solicitado no pedido Initiate Job está alinhado ao hash de árvore e o intervalo para download no Get

Cabeçalho	Description
	Job Output também está alinhado ao hash de árvore. Para obter mais informações sobre intervalos alinhados ao hash de árvore, consulte Receber somas de verificação durante o download de dados. Por exemplo, se na solicitação Initiate Job, tiver especificado um intervalo alinhado ao hash de árvore (o que inclui todo o arquivo), você receberá a soma de verificação dos dados cujo download fez nas seguintes condições: • Você recebe todo o intervalo dos dados recuperados. • Você solicita um intervalo de bytes dos dados recuperados que tem um tamanho de um megabyte (1.024 KB) multiplicado por uma potência de 2 e que começa e termina em um múltiplo do tamanho do intervalo solicitado. Por exemplo, se tiver 3,1 MB de dados recuperados e especificar um intervalo a ser retornado que comece em 1 MB e termine em 2 MB, o `x-amz-sha256-tree-hash` será retornado como um cabeçalho de resposta. • Você solicita um intervalo a ser retornado dos dados recuperados que vai para o final dos dados, e o início do intervalo é um múltiplo do tamanho do intervalo a ser recuperado arredondado para a próxima potência de dois, mas não menor que um megabyte (1.024 KB). Por exemplo, se tiver 3,1 MB de dados recuperados e especificar um intervalo a ser retornado que comece em 2 MB e termine em 3,1 MB (o final dos dados), o `x-amz-sha256-tree-hash` será retornado como um cabeçalho de resposta. Tipo: string

Corpo da resposta

O Amazon Glacier retorna a saída do trabalho no corpo da resposta. Dependendo do tipo de trabalho, a saída pode ser o conteúdo do arquivo ou o inventário de cofre. No caso de um inventário de cofre, por padrão, a lista de inventários é retornada como o corpo JSON a seguir.

```
{
 "VaultARN": String,
 "InventoryDate": String,
 "ArchiveList": [
      {"ArchiveId": String,
       "ArchiveDescription": String,
       "CreationDate": String,
       "Size": Number,
       "SHA256TreeHash": String
      },
      ...
    ]
}
```

Se você tiver solicitado o formato de saída Comma-Separated Values (CSV – Valores separados por vírgula) quando iniciou o trabalho de inventário, o inventário de cofre será retornado em formato CSV no corpo. O formato CSV tem cinco colunas "ArchiveId“," ArchiveDescription “," CreationDate “, “Tamanho” e "SHA256TreeHash" com as mesmas definições dos campos JSON correspondentes.

> **Note**
>
> No formato CSV retornado, os campos podem ser retornados com o campo inteiro entre aspas duplas. Os campos que contenham uma vírgula ou aspas duplas sempre são retornados entre aspas duplas. Por exemplo, `my archive description,1` é retornado como `"my archive description,1"`. Caracteres de aspas duplas que estejam entre campos entre aspas duplas têm escape com um caractere de barra invertida antes deles. Por exemplo, `my archive description,1"2` é retornado como `"my archive description,1\"2"` e `my archive description,1\"2` é retornado como `"my archive description,1\\"2"`. O caractere de barra invertida não é de escape.

O corpo da resposta JSON contém os seguintes campos JSON:

ArchiveDescription

A descrição de um arquivo.

Tipo: string

ArchiveId

O ID de um arquivo.

Tipo: string

ArchiveList

Uma matriz de metadados de arquivo. Cada objeto na matriz representa metadados de um arquivo contido no cofre.

Tipo: matriz

CreationDate

A data e hora UTC em que o arquivo foi criado.

Tipo: uma representação de string no formato de data ISO 8601, por exemplo `2013-03-20T17:03:43.221Z`.

InventoryDate

A data e hora UTC do inventário mais recente do cofre que foi concluído após alterações terem sido feitas no cofre. Embora o Amazon Glacier prepare um inventário de cofre uma vez por dia, a data do inventário será atualizada somente se tiver ocorrido adições ou exclusões de arquivo no cofre desde o último inventário.

Tipo: uma representação de string no formato de data ISO 8601, por exemplo `2013-03-20T17:03:43.221Z`.

SHA256TreeHash

O hash de árvore do arquivo.

Tipo: string

Tamanho

O tamanho em bytes do arquivo.

Tipo: número

VaultARN

O Amazon Resource Name (ARN – Nome de recurso da Amazon) do qual a recuperação do arquivo foi solicitada.

Tipo: string

Erros

Para obter informações sobre exceções e mensagens de erro do Amazon Glacier, consulte Respostas de erro.

Exemplos

O exemplo a seguir mostra a solicitação de um trabalho que recupera um arquivo.

Exemplo 1: fazer download da saída

Este exemplo recupera dados preparados pelo Amazon Glacier em resposta à solicitação do trabalho de recuperação de arquivo de iniciação.

Exemplo de solicitação

```
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
```

Exemplo de resposta

Esta é uma resposta de exemplo de um trabalho de recuperação de arquivo. O cabeçalho `Content-Type` é `application/octet-stream` e esse cabeçalho `x-amz-sha256-tree-hash` está incluído na resposta, o que significa que todos os dados do trabalho são retornados.

```
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.]
```

Esta é uma resposta de exemplo de um trabalho de recuperação de inventário. O cabeçalho `Content-Type` é `application/json`. A resposta também não inclui o cabeçalho `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"
   }
```

```
  ]
}
```

Exemplo 2: fazer download somente da saída parcial

Este exemplo recupera somente uma parte do arquivo preparado pelo Amazon Glacier em resposta à solicitação do trabalho de recuperação de arquivo de iniciação. A solicitação usa o cabeçalho `Range` opcional para recuperar somente os primeiros 1.024 bytes.

Exemplo de solicitação

```
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
```

Exemplo de resposta

A resposta bem-sucedida a seguir mostra a resposta `206 Partial Content`. Nesse caso, a resposta também inclui um cabeçalho `Content-Range` que especifica o intervalo de bytes retornado pelo Amazon Glacier.

```
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.]
```

Seções relacionadas

- Trabalho de descrição (GET JobID)

- Initiate Job (trabalhos POST)
- Gerenciamento de identidade e acesso para o Amazon Glacier

Initiate Job (trabalhos POST)

Essa operação inicia os seguintes tipos de trabalhos no Amazon Glacier:

- `archive-retrieval`: recupera um arquivo
- `inventory-retrieval`: cria um inventário de um cofre

Tópicos

- Inicialização de um trabalho de recuperação de arquivo ou inventário de cofre
- Solicitações
- Respostas
- Exemplos
- Seções relacionadas

Inicialização de um trabalho de recuperação de arquivo ou inventário de cofre

Recuperar um arquivo ou um inventário de cofre são operações assíncronas que exigem que você inicie um trabalho. Após ser iniciado, o trabalho não pode ser cancelado. A recuperação é um processo de duas etapas:

1. Inicie um trabalho de recuperação usando a operação Initiate Job (trabalhos POST).

   > **Important**
   >
   > Uma política de recuperação de dados pode fazer com que a solicitação do trabalho de recuperação de inicialização apresente falha com `PolicyEnforcedException`. Para obter mais informações sobre políticas de recuperação de dados, consulte Políticas de recuperação de dados do Amazon Glacier. Para obter mais informações sobre a exceção `PolicyEnforcedException`, consulte Respostas de erro.

2. Após a conclusão do trabalho, faça download dos bytes usando a operação Get Job Output (GET output).

A solicitação de recuperação é executada de maneira assíncrona. Quando você inicia um trabalho de recuperação, o Amazon Glacier cria um trabalho e retorna um ID de trabalho na resposta. Quando o Amazon Glacier conclui o trabalho, você pode receber a saída do trabalho (dados do arquivo ou do inventário). Para saber informações sobre como obter a saída do trabalho, consulte a operação Get Job Output (GET output).

O trabalho deverá ser concluído para você obter a saída. Para determinar quando um trabalho está concluído, você tem as seguintes opções:

- Usar uma notificação do Amazon SNS: você pode especificar um tópico do Amazon SNS no qual o Amazon Glacier poderá publicar uma notificação depois que o trabalho for concluído. Você pode especificar um tópico do SNS por solicitação de trabalho. A notificação só será enviada depois que o Amazon Glacier concluir o trabalho. Além de especificar um tópico do SNS por solicitação de trabalho, você pode configurar notificações para um cofre, de maneira que as notificações de trabalho sejam enviadas para todas as recuperações. Para obter mais informações, consulte Definir configuração de notificação de cofre (PUT notification-configuration).
- Obter detalhes do trabalho:você pode fazer uma solicitação Trabalho de descrição (GET JobID) para obter informações de status do trabalho enquanto um trabalho está em andamento. No entanto, é mais eficiente usar uma notificação do Amazon SNS para determinar quando um trabalho está concluído.

> **Note**
>
> As informações obtidas por você por meio da notificação são as mesmas recebidas quando se chama Trabalho de descrição (GET JobID).

Se, para um evento específico, você adicionar a configuração de notificação no cofre e também especificar um tópico do SNS na solicitação do trabalho de iniciação, o Amazon Glacier enviará ambas as notificações. Para obter mais informações, consulte Definir configuração de notificação de cofre (PUT notification-configuration).

O inventário de cofre

O Amazon Glacier atualiza um inventário de cofre aproximadamente uma vez por dia, começando no dia em que você faz o primeiro upload para o cofre. Se não houver adições ou exclusões de arquivo

no cofre desde o último inventário, a data do inventário não será atualizada. Quando você inicia um trabalho para um inventário do cofre, o Amazon Glacier retorna o último inventário gerado, que é point-in-time um instantâneo e não dados em tempo real.

Depois que o Amazon Glacier criar o primeiro inventário para o cofre, normalmente levará de meio dia a um dia para que esse inventário esteja disponível para recuperação.

Talvez você não ache útil recuperar um inventário de cofre para cada upload de arquivo. No entanto, suponhamos que você mantenha um banco de dados no lado do cliente associando metadados sobre os arquivos cujo upload fez para o Amazon Glacier. Nesse caso, talvez você ache o inventário de cofre útil para reconciliar informações, conforme necessário, no seu banco de dados com o inventário de cofre real. Para obter mais informações sobre os campos de dados retornados em uma saída do trabalho de inventário, consulte Corpo da resposta.

Recuperação do inventário de intervalo

Você pode limitar o número de itens do inventário recuperados filtrando a data de criação do arquivo ou definindo um limite.

Filtrar por data de criação do arquivo

Você pode recuperar itens de inventário para arquivos criados entre `StartDate` e `EndDate` especificando valores para esses parâmetros na solicitação Iniciar trabalho. Os arquivos criados em `StartDate` ou após essa data e antes de `EndDate` serão retornados. Se fornecer somente `StartDate` sem `EndDate`, recuperará o inventário de todos os arquivos criados em `StartDate` ou após essa data. Se fornecer somente `EndDate` sem `StartDate`, terá de volta o inventário de todos os arquivos criados antes de `EndDate`.

Limitar itens de inventário por recuperação

Você pode limitar o número de itens de inventário retornados definindo o parâmetro `Limit` na solicitação Iniciar trabalho. A saída do trabalho de inventário contém itens de inventário até o `Limit` especificado. Se houver mais itens de inventário disponíveis, o resultado será paginado. Depois que um trabalho for concluído, você poderá usar a operação Trabalho de descrição (GET JobID) para obter um marcador a ser usado em uma solicitação subsequente Iniciar trabalho. O marcador indicará o ponto de partida para recuperar o próximo conjunto de itens de inventário. Você pode percorrer todo o inventário fazendo várias solicitações de Iniciar trabalho com o marcador da saída anterior de Descrever trabalho. Para fazer isso, obtenha um marcador em Descrever trabalho que retorne um valor nulo, indicando que não há mais itens de inventário disponíveis.

Você pode usar o parâmetro `Limit` com os parâmetros do intervalo de datas.

Recuperação do arquivo no intervalo

Você pode iniciar a recuperação de todo o arquivo ou de um intervalo do arquivo. No caso da recuperação de um intervalo do arquivo, você especifica um intervalo de bytes a ser retornado ou todo o arquivo. O intervalo especificado deve ser alinhado a megabyte (MB). Em outras palavras, o valor de início do intervalo deve ser divisível por 1 MB e o valor final do intervalo mais 1 deve ser divisível por 1 MB ou igual ao final do arquivo. Se a recuperação do arquivo no intervalo não estiver alinhada a megabyte, essa operação retornará uma resposta `400`. Além disso, para garantir que você obtenha valores da soma de verificação dos dados cujo download você faz usando Obter saída de trabalho (Get Job Output (GET output)), o intervalo deve estar alinhado ao hash de árvore. Para obter mais informações sobre intervalos alinhados ao hash de árvore, consulte Receber somas de verificação durante o download de dados.

Níveis expressos, padrão e em massa

Ao iniciar um trabalho de recuperação de arquivo, você pode especificar uma das seguintes opções no campo `Tier` do corpo da solicitação:

- **`Expedited`**: expresso permite que você acesse rapidamente seus dados quando são feitas solicitações de urgência ocasionais para recuperar arquivos. Exceto para os arquivos maiores (mais de 250 MB), os dados acessados por meio de níveis expressos são disponibilizados geralmente entre 1 e 5 minutos.
- **`Standard`**: padrão permite que você acesse qualquer um dos arquivos em várias horas. Os dados acessados usando o nível Padrão normalmente são disponibilizados entre 3 e 5 horas. Essa é a opção padrão para solicitações de trabalho que não especificam a opção de nível.
- **`Bulk`**: em massa é o nível de menor custo do Amazon Glacier, permitindo recuperar grandes quantidades de dados, até mesmo petabytes, em um dia e com um custo baixo. Os dados acessados usando o nível Em massa normalmente são disponibilizados entre 5 e 12 horas.

Para obter mais informações recuperações expressas e em massa, consulte Recuperação de arquivos do Amazon Glacier.

Solicitações

Para iniciar um trabalho, você usa o método `POST` HTTP e delimita a solicitação ao sub-recurso `jobs` do cofre. Você especifica detalhes da solicitação de trabalho no documento JSON da solicitação. O tipo de trabalho é especificado com o campo `Type`. Se desejar, você pode especificar

um campo `SNSTopic` para indicar um tópico do Amazon SNS no qual o Amazon Glacier poderá publicar uma notificação depois que o trabalho for concluído.

> **Note**
>
> Para publicar uma notificação no Amazon SNS, você deve criar o tópico por conta própria, caso ele ainda não exista. O Amazon Glacier não cria o tópico para você. O tópico deve ter permissões para receber publicações de um cofre do Amazon Glacier. O Amazon Glacier não verifica se o cofre tem permissão para publicação no tópico. Se as permissões não forem configuradas corretamente, talvez você não receba uma notificação, mesmo depois da conclusão do trabalho.

Sintaxe

Esta é a sintaxe de solicitação para iniciar um trabalho.

```
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**
>
> O `AccountId` valor é o Conta da AWS ID da conta proprietária do cofre. Você pode especificar um Conta da AWS ID ou, opcionalmente, um único '-' (hífen). Nesse caso, o Amazon Glacier usa Conta da AWS o ID associado às credenciais usadas para assinar a solicitação. Se você usar um ID de conta, não inclua nenhum hífen ('-') no ID.

Corpo da Solicitação

A solicitação aceita os seguintes dados no formato JSON no corpo da solicitação.

jobParameters

Fornece opções para especificar as informações do trabalho.

Tipo: jobParameters objeto

Obrigatório: sim

Respostas

O Amazon Glacier cria o trabalho. Na resposta, ele retorna o URI do trabalho.

Sintaxe

```
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
```

Cabeçalhos de resposta

Cabeçalho	Description
`Location`	O caminho de URI relativo do trabalho. Você pode usar esse caminho de URI para encontrar o status do trabalho. Para obter mais informações, consulte Trabalho de descrição (GET JobID). Tipo: string Padrão: nenhum
`x-amz-job-id`	O ID do trabalho. Esse valor também está incluído como parte do cabeçalho `Location`. Tipo: string Padrão: nenhum
`x-amz-job-output-path`	O caminho para o local onde os resultados de seleção são armazenados. Tipo: string Padrão: nenhum

Corpo da resposta

Essa operação não retorna um corpo de resposta.

Erros

Essa operação inclui os seguintes erros, além dos possíveis erros comuns a todas as operações do Amazon Glacier. Para obter informações sobre erros do Amazon Glacier e uma lista de códigos de erro, consulte Respostas de erro.

Código	Description	Código de status HTTP	Tipo
`InsufficientCapacityException`	Retornado se houver capacidade insuficiente para processar essa solicitação expressa. Esse erro se aplica somente a recuperações expressas, e não a recuperações padrão ou em massa.	`503 Service Unavailable`	Servidor

Exemplos

Solicitação de exemplo: iniciar um trabalho de recuperação do arquivo

```
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"
}
```

Este é um exemplo do corpo de uma solicitação que especifica um intervalo do arquivo a ser recuperado usando-se o campo `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"
}
```

Exemplo de resposta

```
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
```

Solicitação de exemplo: iniciar um trabalho de recuperação do inventário

A solicitação a seguir inicia um trabalho de recuperação do inventário para obter uma lista de arquivos do cofre `examplevault`. O `Format` definido como `CSV` no corpo da solicitação indica que o inventário é retornado no formato 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"
}
```

Exemplo de resposta

```
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
```

Exemplos de solicitações: Iniciar um trabalho de recuperação de inventário usando a filtragem de dados com um limite definido e uma solicitação posterior para recuperar a próxima página dos itens de inventário.

A solicitação a seguir inicia um trabalho de recuperação de inventário de cofre usando a filtragem de data e definindo um limite.

```
{
    "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"
    },
}
```

A solicitação a seguir é um exemplo de uma solicitação subsequente para recuperar a próxima página de itens de inventário usando um marcador obtido de Trabalho de descrição (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
    },
}
```

Exemplo de resposta

```
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/
```

Seções relacionadas

- Trabalho de descrição (GET JobID)
- Get Job Output (GET output)
- Gerenciamento de identidade e acesso para o Amazon Glacier

List Jobs (GET jobs)

Description

Essa operação lista trabalhos de um cofre, incluindo os trabalhos em andamento e os que foram concluídos recentemente.

> **Note**
>
> O Amazon Glacier mantém trabalhos concluídos recentemente por um período antes de excluí-los; no entanto, ele acaba removendo os trabalhos concluídos. A saída de trabalhos

concluídos pode ser recuperada. Manter os trabalhos por um período após eles terem sido concluídos permite obter uma saída de trabalho caso você perca a notificação de conclusão do trabalho ou caso a primeira tentativa de download apresente falha. Por exemplo, suponhamos que você inicie um trabalho de recuperação de arquivo para fazer download de um arquivo. Depois que o trabalho é concluído, você começa a fazer download do arquivo, mas encontra um erro de rede. Nesse cenário, você poderá tentar novamente e fazer download do arquivo enquanto o trabalho existir.

A operação `List Jobs` oferece suporte à paginação. Você deve sempre verificar o campo de resposta `Marker`. Se não houver mais trabalhos a serem listados, o campo `Marker` será definido como `null`. Se houver mais trabalhos a serem listados, o campo `Marker` será definido como um valor não nulo, que você pode usar para continuar a paginação da lista. Para retornar uma lista de trabalhos que comece em um específico, defina o parâmetro de solicitação `marker` como o valor `Marker` para esse trabalho que você obteve de uma `List Jobs`solicitação anterior.

Você pode definir um limite máximo para o número de trabalhos retornados na resposta especificando o parâmetro `limit` na solicitação. O limite padrão é 50. O número de trabalhos retornados pode ser menor que o limite, mas o número de trabalhos retornados jamais excede o limite.

Além disso, você pode filtrar a lista de trabalhos retornados especificando o parâmetro `statuscode` opcional ou o parâmetro `completed`, ou ambos. Usando o parâmetro `statuscode`, você pode especificar para retornar somente trabalhos correspondentes ao status `InProgress`, `Succeeded` ou `Failed`. Usando o `completed` parâmetro, você pode especificar para retornar somente trabalhos que tenham sido concluídos (`true`) ou trabalhos que não tenham sido concluídos (`false`).

Solicitações

Sintaxe

Para retornar uma lista de trabalhos de todos os tipos, envie uma solicitação `GET` para o URI do sub-recurso `jobs` do cofre.

```
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**
>
> O `AccountId` valor é o Conta da AWS ID da conta proprietária do cofre. Você pode especificar um Conta da AWS ID ou, opcionalmente, um único '-' (hífen). Nesse caso, o Amazon Glacier usa Conta da AWS o ID associado às credenciais usadas para assinar a solicitação. Se você usar um ID de conta, não inclua nenhum hífen ('-') no ID.

Parâmetros da solicitação

Name (Nome)	Description	Obrigatório
`completed`	O estado dos trabalhos a serem retornados. É possível especificar `true` ou `false`. Tipo: booliano Restrições: nenhuma	Não
`limit`	O número máximo de trabalhos a serem retornados. O limite padrão é 50. O número de trabalhos retornados pode ser menor que o limite especificado, mas o número de trabalhos retornados jamais excede o limite. Tipo: string Restrições: valor do inteiro mínimo de 1. Valor do inteiro máximo de 50.	Não
`marker`	Uma string invisível usada na paginação que especifica o trabalho no qual a listagem dos trabalhos deve começar. Você obtém o valor `marker` de uma `List Jobs`resposta anterior. Você somente precisará incluir o `marker` se continuar a paginação dos resultados iniciada em uma `List Jobs` solicitação anterior.	Não

Name (Nome)	Description	Obrigatório
	Tipo: string Restrições: nenhuma	
`statuscode`	O tipo de status do trabalho a ser retornado. Tipo: string Restrições: um dos valores `InProgress` , `Succeeded` ou `Failed`.	Não

Cabeçalhos de solicitação

Essa operação usa somente os cabeçalhos de resposta que são comuns a maioria das respostas. Para informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Corpo da Solicitação

Essa operação não tem um corpo de solicitação.

Respostas

Sintaxe

```
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"
}
```

Cabeçalhos de resposta

Essa operação usa somente os cabeçalhos de resposta que são comuns a maioria das respostas. Para informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Corpo da resposta

O corpo da resposta contém os seguintes campos JSON.

JobList

Uma lista de objetos de trabalho. Cada objeto de trabalho contém metadados que descrevem o trabalho.

Tipo: matriz de objetos GlacierJobDescription

Marker

Uma string invisível que representa onde continuar a paginação dos resultados. Você usa o valor `marker` em uma nova `List Jobs`solicitação para obter mais trabalhos na lista. Se não houver mais trabalhos a serem listados, esse valor será `null`.

Tipo: string

Erros

Para obter informações sobre exceções e mensagens de erro do Amazon Glacier, consulte Respostas de erro.

Exemplos

Os exemplos a seguir demonstram como retornar informações sobre trabalhos de cofre. O primeiro exemplo retorna uma lista de dois trabalhos, e o segundo exemplo retorna um subconjunto de trabalhos.

Exemplo: retornar todos os trabalhos

Exemplo de solicitação

A solicitação `GET` a seguir retorna os trabalhos de um cofre.

```
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
```

Exemplo de resposta

A resposta a seguir inclui um trabalho de recuperação de arquivo e um trabalho de recuperação de inventário que contém um marcador usado para continuar a paginação da recuperação do inventário de cofre. A resposta também mostra o campo `Marker` definido como `null`, o que indica que não há mais trabalhos a serem listados.

```
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
}
```

Exemplo: retornar uma lista parcial de trabalhos

Exemplo de solicitação

A solicitação GET a seguir retorna o trabalho especificado pelo parâmetro `marker`. Definir o `limit`parâmetro como 2especifica que até dois trabalhos são retornados.

```
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
```

Exemplo de resposta

A resposta a seguir mostra dois trabalhos retornados e o campo `Marker` definido como um valor não nulo que pode ser usado para continuar a paginação da lista de trabalhos.

```
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"
}
```


Seções relacionadas

- Trabalho de descrição (GET JobID)
- Gerenciamento de identidade e acesso para o Amazon Glacier

Tipos de dados usados em operações de trabalho

Estes são os tipos de dados usados com as operações de trabalho no Amazon Glacier.

Tópicos

- CSVInput
- CSVOutput
- Criptografia
- GlacierJobDescription
- Concessão

- Grantee
- InputSerialization
- InventoryRetrievalJobInput
- jobParameters
- OutputLocation
- OutputSerialization
- S3Location
- SelectParameters

CSVInput

Contém informações sobre o arquivo de valores separados por vírgulas (CSV).

Conteúdo

Comentários

Um único caractere usado para indicar que uma linha deve ser ignorada quando o caractere está presente no início da linha.

Tipo: string

Obrigatório: não

FieldDelimiter

Um único caractere usado para separar campos individuais em um registro. O caractere deve ser um `\n`, `\r` ou um caractere ASCII no intervalo 32–126. O padrão é uma vírgula (`,`).

Tipo: string

Padrão: ,

Obrigatório: não

FileHeaderInfo

Um valor que descreve o que fazer com a primeira linha da entrada.

Tipo: string

Valores válidos: `Use` | `Ignore` | `None`

Obrigatório: não

QuoteCharacter

Um único caractere usado como um caractere de escape em que o delimitador de campo é parte do valor.

Tipo: string

Obrigatório: não

QuoteEscapeCharacter

Um único caractere usado para fazer o escape do caractere de aspas em um valor que já recebeu escape.

Tipo: string

Obrigatório: não

RecordDelimiter

Um único caractere usado para separar registros individuais.

Tipo: string

Obrigatório: não

Mais informações

- Initiate Job (trabalhos POST)

CSVOutput

Contém informações sobre o formato de valores separados por vírgulas (CSV) em que os resultados do trabalho são armazenados.

Conteúdo

FieldDelimiter

Um único caractere usado para separar campos individuais em um registro.

Tipo: string

Obrigatório: não

QuoteCharacter

Um único caractere usado como um caractere de escape em que o delimitador de campo é parte do valor.

Tipo: string

Obrigatório: não

QuoteEscapeCharacter

Um único caractere usado para fazer o escape do caractere de aspas em um valor que já recebeu escape.

Tipo: string

Obrigatório: não

QuoteFields

Um valor que indica se todos os campos de saída devem ser colocados entre aspas.

Valores válidos: `ALWAYS` | `ASNEEDED`

Tipo: string

Obrigatório: não

RecordDelimiter

Um único caractere usado para separar registros individuais.

Tipo: string

Obrigatório: não

Mais informações

- Initiate Job (trabalhos POST)

Criptografia

Contém informações sobre a criptografia usada para armazenar os resultados do trabalho no Amazon S3.

Conteúdo

Criptografia

O algoritmo de criptografia no lado do servidor usado para armazenar os resultados do trabalho no Amazon S3. O padrão é sem criptografia.

Tipo: string

Valores válidos: `aws:kms` | `AES256`

Obrigatório: não

KMSContext

Opcional. Se o tipo de criptografia for `aws:kms`, você poderá usar esse valor para especificar o contexto da criptografia nos resultados do trabalho.

Tipo: string

Obrigatório: não

KMSKeyId

O ID da chave do AWS Key Management Service (AWS KMS) a ser usado na criptografia do objeto.

Tipo: string

Obrigatório: não

Mais informações

- Initiate Job (trabalhos POST)

GlacierJobDescription

Contém a descrição de um trabalho do Amazon Glacier.

Conteúdo

Ação

O tipo de trabalho. É `ArchiveRetrieval`, `InventoryRetrieval` ou `Select`.

Tipo: string

ArchiveId

O ID de arquivo solicitado para um trabalho de seleção ou de recuperação de arquivo. Do contrário, esse campo será `null`.

Tipo: string

ArchiveSHA256TreeHash

O hash de árvore SHA256 de todo o arquivo de uma recuperação do arquivo. Para trabalhos de recuperação do inventário, esse campo é `null`.

Tipo: string

ArchiveSizeInBytes

Para um trabalho `ArchiveRetrieval`, trata-se do tamanho em bytes do arquivo solicitado para download. Para o trabalho `InventoryRetrieval`, o valor é `null`.

Tipo: número

Concluído

`true` se o trabalho for concluído;, do contrário `false`.

Tipo: booliano

CompletionDate

A data em que o trabalho foi concluído.

A hora UTC (Horário Coordenado Universal) em que a solicitação de trabalho foi concluída. Enquanto o trabalho estiver em andamento, o valor será nulo.

Tipo: uma representação de string no formato de data ISO 8601, por exemplo `2013-03-20T17:03:43.221Z`.

CreationDate

A data UTC quando o trabalho foi iniciado.

Tipo: uma representação de string no formato de data ISO 8601, por exemplo `2013-03-20T17:03:43.221Z`.

InventoryRetrievalParameters

Os parâmetros de entrada usados em uma recuperação de inventário de intervalo.

Tipo: InventoryRetrievalJobInput objeto

InventorySizeInBytes

Para um trabalho `InventoryRetrieval`, trata-se do tamanho em bytes do inventário solicitado para download. No trabalho `ArchiveRetrieval` ou `Select`, o valor é `null`.

Tipo: número

JobDescription

A descrição do trabalho que você forneceu quando iniciou o trabalho.

Tipo: string

JobId

O ID que identifica o trabalho no Amazon Glacier.

Tipo: string

JobOutputPath

Contém o local de saída do trabalho.

Tipo: string

OutputLocation

Um objeto com informações sobre o local onde os resultados e os erros do trabalho de seleção são armazenados.

Tipo: OutputLocation objeto

RetrievalByteRange

O intervalo de bytes recuperado para trabalhos de recuperação de arquivo no formato "*`StartByteValue`*-*`EndByteValue`*". Se nenhum intervalo for especificado na recuperação do arquivo, então, todo o arquivo será recuperado e StartByteValue será igual a 0 e EndByteValue

será igual ao tamanho do arquivo menos 1. Para trabalhos de recuperação do inventário, esse campo é `null`.

Tipo: string

SelectParameters

Um objeto com informações sobre os parâmetros usados em uma seleção.

Tipo: SelectParameters objeto

SHA256TreeHash

O valor do hash de árvore SHA256 para o intervalo solicitado de um arquivo. Se a solicitação Initiate Job (trabalhos POST) para um arquivo tiver especificado um intervalo alinhado ao hash de árvore, esse campo retornará um valor. Para obter mais informações sobre o alinhamento ao hash de árvore para recuperações do intervalo de arquivos, consulte Receber somas de verificação durante o download de dados.

Quando todo o arquivo for recuperado, esse valor será igual ao valor de `ArchiveSHA256TreeHash`.

Esse campo é `null` nas seguintes situações:

- Os trabalhos de recuperação de arquivo que especificam um intervalo não alinhado ao hash de árvore.
- Trabalhos de arquivamento que especificam um intervalo igual a todo o arquivamento e cujo status do trabalho seja `InProgress`.
- Trabalhos de inventário.
- Trabalhos de seleção.

Tipo: string

SNSTopic

O nome do recurso da Amazon (ARN) que representa um tópico do Amazon SNS em que a notificação da conclusão ou da falha do trabalho é enviada, se a notificação tiver sido configurada na iniciação do trabalho (Initiate Job (trabalhos POST)).

Tipo: string

StatusCode

O código que indica o status do trabalho.

Valores válidos: `InProgress` | `Succeeded` | `Failed`

Tipo: string

StatusMessage

A mensagem de status do trabalho.

Tipo: string

Tier

O nível de acesso a dados a ser usado na seleção ou na recuperação de arquivo.

Valores válidos: `Expedited` | `Standard` | `Bulk`

Tipo: string

VaultARN

O ARN do cofre do qual o trabalho é um sub-recurso.

Tipo: string

Mais informações

- Initiate Job (trabalhos POST)

Concessão

Contém informações sobre uma concessão.

Conteúdo

Grantee

O favorecido.

Tipo: Grantee objeto

Obrigatório: não

Permissão

A permissão concedida ao favorecido.

Tipo: string

Valores válidos: `FULL_CONTROL` | `WRITE` | `WRITE_ACP` | `READ` | `READ_ACP`

Obrigatório: não

Mais informações

- Initiate Job (trabalhos POST)

Grantee

Contém informações sobre um favorecido.

Conteúdo

DisplayName

O nome da tela do favorecido.

Tipo: string

Obrigatório: não

EmailAddress

O endereço de e-mail do favorecido.

Tipo: string

Obrigatório: não

ID

O ID de usuário canônico do favorecido.

Tipo: string

Obrigatório: não

Tipo

O tipo do favorecido.

Tipo: string

Valores válidos: `AmazonCustomerByEmail` | `CanonicalUser` | `Group`

Obrigatório: não

URI

O URI do grupo do favorecido.

Tipo: string

Obrigatório: não

Mais informações

- Initiate Job (trabalhos POST)

InputSerialization

Descreve como o arquivo será serializado.

Conteúdo

CSV

Um objeto que descreve a serialização de um objeto codificado por CSV.

Tipo: CSVInput objeto

Obrigatório: não

Mais informações

- Initiate Job (trabalhos POST)

InventoryRetrievalJobInput

Fornece opções para especificar um trabalho de recuperação de inventário de intervalo.

Conteúdo

EndDate

O final do intervalo de datas, em UTC, para uma recuperação de inventário de cofre que inclua os arquivos criados antes dessa data.

Valores válidos: uma representação de string no formato de data ISO 8601 (`YYYY-MM-DDThh:mm:ssTZD`) em segundos, por exemplo `2013-03-20T17:03:43Z`.

Tipo: string Uma representação de string no formato de data ISO 8601 (`YYYY-MM-DDThh:mm:ssTZD`) em segundos, por exemplo `2013-03-20T17:03:43Z`.

Obrigatório: não

Formato

O formato de saída da lista de inventários de cofre, definido pela solicitação Initiate Job (trabalhos POST) durante a inicialização de um trabalho para recuperação de um inventário de cofre.

Valores válidos: `CSV` | `JSON`

Obrigatório: não

Tipo: string

Limite

O número máximo de itens de inventário que podem ser retornados para cada solicitação de recuperação de inventário de cofre.

Valores válidos: um valor inteiro maior ou igual a 1.

Tipo: string

Obrigatório: não

Marker

Uma string invisível que representa onde continuar a paginação dos resultados da recuperação de inventário do cofre. Você usa esse marcador em uma nova solicitação `Initiate Job` para obter itens de inventário adicionais. Se não houver itens de inventário, esse valor será nulo.

Tipo: string

Obrigatório: não

StartDate

O início do intervalo de datas, em UTC, para uma recuperação de inventário de cofre que inclua arquivos criados nessa ou após essa data.

Valores válidos: uma representação de string no formato de data ISO 8601 (`YYYY-MM-DDThh:mm:ssTZD`) em segundos, por exemplo `2013-03-20T17:03:43Z`.

Tipo: string Uma representação de string no formato de data ISO 8601 (`YYYY-MM-DDThh:mm:ssTZD`) em segundos, por exemplo `2013-03-20T17:03:43Z`.

Obrigatório: não

Mais informações

- Initiate Job (trabalhos POST)

jobParameters

Fornece opções para definir um trabalho.

Conteúdo

ArchiveId

O ID do arquivo desejado. Este campo só será obrigatório se o campo `Type` estiver definido como `select` ou `archive-retrieval`. Um erro ocorrerá se você especificar esse campo para uma solicitação de trabalho de recuperação de inventário.

Valores válidos: devem ser um ID de arquivo válido obtido de uma solicitação anterior feita ao Amazon Glacier.

Tipo: string

Obrigatório: sim quando `Type` for definido como `select` ou `archive-retrieval`.

Descrição

A descrição opcional do trabalho.

Valores válidos: a descrição deve ser menor ou igual a 1.024 bytes. Os caracteres permitidos são ASCII 7 bits sem códigos de controle — mais especificamente, valores ASCII 32–126 decimais ou 0x20–0x7E hexadecimais.

Tipo: string

Obrigatório: não

Formato

(Opcional) O formato da saída, ao iniciar um trabalho para recuperar um inventário de cofre. Se você estiver iniciando um trabalho de inventário e não especificar um campo `Format`, JSON será o formato padrão.

Valores válidos: `CSV` | `JSON`

Tipo: string

Obrigatório: não

InventoryRetrievalParameters

Os parâmetros de entrada usados em uma recuperação de inventário de intervalo.

Tipo: InventoryRetrievalJobInput objeto

Obrigatório: não

OutputLocation

Um objeto com informações sobre o local onde os resultados do trabalho de seleção são armazenados.

Tipo: OutputLocation objeto

Obrigatório: sim para trabalhos `select`.

RetrievalByteRange

O intervalo de bytes a ser recuperado para uma `archive-retrieval`, no formato "*`StartByteValue`*-*`EndByteValue`*". Se esse campo não for especificado, todo o arquivo será recuperado. Se esse campo for especificado, o intervalo de bytes deverá ser alinhado a megabyte (1024*1024). Alinhado a megabyte significa que StartByteValue deve ser divisível por

1 MB e EndByteValue mais 1 deve ser divisível por 1 MB ou ser o final do arquivo especificado como o valor do tamanho de bytes do arquivo menos 1. Se RetrievalByteRange não estiver alinhado a megabyte, essa operação retornará a resposta `400`.

Um erro ocorrerá se você especificar esse campo em uma solicitação de trabalho `inventory-retrieval` ou `select`.

Tipo: string

Obrigatório: não

SelectParameters

Um objeto com informações sobre os parâmetros usados em uma seleção.

Tipo: SelectParameters objeto

Obrigatório: não

SNSTopic

O nome do recurso da Amazon (ARN) do tópico do Amazon SNS, em que o Amazon Glacier envia uma notificação quando o trabalho é concluído e a saída fica pronta para fazer download. O tópico especificado publica a notificação para os assinantes.

O tópico do SNS deve existir. Do contrário, o Amazon Glacier não o criará para você. Além disso, o tópico do SNS deve ter uma política que permita à conta que criou o trabalho publicar mensagens no tópico. Para obter informações sobre nomes de tópico do SNS, consulte CreateTopic na Referência de API do Amazon Simple Notification Service.

Tipo: string

Obrigatório: não

Tier

O nível a ser usado para um trabalho de seleção ou de recuperação de arquivo. `Standard` é o valor padrão usado.

Valores válidos: `Expedited` | `Standard` | `Bulk`

Tipo: string

Obrigatório: não

Tipo

O tipo de trabalho. Você pode iniciar um trabalho para executar uma consulta select em um arquivo, recuperar um arquivo ou obter um inventário de um cofre.

Valores válidos: `select` | `archive-retrieval` | `inventory-retrieval`

Tipo: string

Obrigatório: sim

Mais informações

- Initiate Job (trabalhos POST)

OutputLocation

Contém informações sobre o local em que os resultados e os erros de trabalho são armazenados.

Conteúdo

S3

Um objeto que descreve um local do Amazon S3 para receber os resultados da solicitação de restauração.

Tipo: S3Location

Obrigatório: sim

Mais informações

- Initiate Job (trabalhos POST)

OutputSerialization

Descreve como a saída será serializada.

Conteúdo

CSV

Um objeto que descreve a serialização dos resultados de consulta codificados por valores separados por vírgulas (CSV).

Tipo: CSVOutput objeto

Obrigatório: não

Mais informações

- Initiate Job (trabalhos POST)

S3Location

Contém informações sobre o local do Amazon S3 em que os resultados do trabalho são armazenados.

Conteúdo

AccessControlList

Uma lista de concessões que controlam o acesso aos resultados armazenados.

Tipo: matriz de objetos Concessão

Obrigatório: não

BucketName

O nome do bucket do Amazon S3 no qual os resultados do trabalho são armazenados. O bucket deve estar na mesma região da AWS do cofre que contém o objeto de arquivo de entrada.

Tipo: string

Obrigatório: sim

CannedACL

A lista de controle de acesso pré-configurada a ser aplicada aos resultados do trabalho.

Tipo: string

Valores válidos: `private` | `public-read` | `public-read-write` | `aws-exec-read` | `authenticated-read` | `bucket-owner-read` | `bucket-owner-full-control`

Obrigatório: não

Criptografia

Um objeto com informações sobre a criptografia usada para armazenar os resultados do trabalho no Amazon S3.

Tipo: Criptografia objeto

Obrigatório: não

Prefixo

O prefixo que precede os resultados dessa solicitação. O tamanho máximo do prefixo é 512 bytes.

Tipo: string

Obrigatório: sim

StorageClass

A classe de armazenamento usada para armazenar os resultados do trabalho.

Tipo: string

Valores válidos: `STANDARD` | `REDUCED_REDUNDANCY` | `STANDARD_IA`

Obrigatório: não

Tags

O conjunto de tags aplicado aos resultados do trabalho.

Tipo: mapa de string para string

Obrigatório: não

UserMetadata

Um mapa de metadados a ser armazenado com os resultados do trabalho no Amazon S3.

Tipo: mapa de string para string

Obrigatório: não

Mais informações

- Initiate Job (trabalhos POST)

SelectParameters

Contém informações sobre os parâmetros usados na seleção.

Conteúdo

Expressão

A expressão usada para selecionar o objeto. A expressão não deve exceder a cota de 128.000 caracteres.

Tipo: string

Obrigatório: sim

ExpressionType

O tipo da expressão fornecida; por exemplo, `SQL`.

Valores Válidos: `SQL`

Tipo: string

Obrigatório: sim

InputSerialization

Descreve o formato de serialização do objeto na seleção.

Tipo: InputSerialization objeto

Obrigatório: não

OutputSerialization

Descreve como os resultados do trabalho de seleção são serializados.

Obrigatório: não

Tipo: OutputSerialization objeto

Mais informações

- Initiate Job (trabalhos POST)

Operações de recuperação de dados

Estas são as operações relacionadas à recuperação de dados disponíveis no Amazon Glacier.

Tópicos

- Política de recuperação para obter dados (GET policy)
- List Provisioned Capacity (GET provisioned-capacity)
- Purchase Provisioned Capacity (POST provisioned-capacity)
- Set Data Retrieval Policy (PUT policy)

Política de recuperação para obter dados (GET policy)

Descrição

Esta operação retorna a política de recuperação de dados atual da conta da Conta da AWS e da região da AWS especificada na solicitação GET. Para obter mais informações sobre políticas de recuperação de dados, consulte Políticas de recuperação de dados do Amazon Glacier.

Solicitações

Para retornar a política de recuperação de dados atual, envie uma solicitação GET HTTP para o URI da política de recuperação de dados conforme mostrado no exemplo de sintaxe a seguir.

Sintaxe

```
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**
>
> O valor de `AccountId` é o ID da Conta da AWS. Esse valor deve corresponder ao ID de Conta da AWS associado às credenciais usadas para assinar a solicitação. Você pode especificar um ID da Conta da AWS ou, de opcionalmente, um '-' (hífen) único. Nesse caso, o Amazon Glacier usa o ID de Conta da AWS associado às credenciais usadas para assinar a solicitação. Se você especificar o ID da conta, não inclua nenhum hífen ('-') no ID.

Parâmetros da solicitação

Essa operação não usa parâmetros de solicitação.

Cabeçalhos de solicitação

Essa operação usa somente os cabeçalhos de solicitação que são comuns a todas as operações. Para informações sobre os cabeçalhos de solicitação comuns, consulte Cabeçalhos de solicitação comuns.

Corpo da Solicitação

Essa operação não tem um corpo de solicitação.

Respostas

Sintaxe

```
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
          }
        ]
    }
}
```

Cabeçalhos de resposta

Essa operação usa somente os cabeçalhos de resposta que são comuns a maioria das respostas. Para informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Corpo da resposta

O corpo da resposta contém os seguintes campos JSON.

BytesPerHour

O número máximo de bytes que podem ser recuperados em uma hora.

Este campo estará presente somente se o valor do campo Strategy for `BytesPerHour`.

Tipo: número

Regras

A regra da política. Embora esse seja um tipo de lista, atualmente deve haver somente uma regra, que contém um campo Strategy e, como opção, um campo BytesPerHour.

Tipo: matriz

Strategy

O tipo de política de recuperação de dados.

Tipo: string

Valores válidos: `BytesPerHour|FreeTier|None`. `BytesPerHour` é equivalente a selecionar Max Retrieval Rate (Taxa de recuperação máxima) no console. `FreeTier` é equivalente a selecionar Somente nível gratuito no console. `None` é equivalente a selecionar No Retrieval Policy (Nenhuma política de recuperação) no console. Para obter mais informações sobre como

selecionar políticas de recuperação de dados no console, consulte Políticas de recuperação de dados do Amazon Glacier.

Erros

Para obter informações sobre exceções e mensagens de erro do Amazon Glacier, consulte Respostas de erro.

Exemplos

O exemplo a seguir demonstra como obter uma política de recuperação de dados.

Exemplo de solicitação

Neste exemplo, uma solicitação GET é enviada para o URI do local de uma política.

```
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
```

Exemplo de resposta

Uma resposta bem-sucedida mostra a política de recuperação de dados no corpo da resposta em formato 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"
          }
       ]
    }
}
```

Seções relacionadas

- Set Data Retrieval Policy (PUT policy)
- Initiate Job (trabalhos POST)

List Provisioned Capacity (GET provisioned-capacity)

Esta operação lista a capacidade provisionada para a Conta da AWS especificada. Para obter mais informações sobre a capacidade provisionada, consulte Opções de recuperação de arquivos.

Uma unidade de capacidade provisionada dura um mês a partir da data e hora da compra, que é a data de início. A unidade expira na data de expiração, que é exatamente um mês depois da data de início até o segundo mais próximo.

Se a data de início é dia 31 de um mês, a data de expiração será o último dia do mês seguinte. Por exemplo, se a data de início é dia 31 de agosto, a data de expiração será dia 30 de setembro. Se a data de início é dia 31 de janeiro, a data de expiração será dia 28 de fevereiro. Você pode consultar essa funcionalidade em Exemplo de resposta.

Sintaxe da solicitação

Para listar a capacidade de recuperação provisionada para uma conta, envie uma solicitação GET HTTP para o URI de capacidade provisionado conforme mostrado no exemplo de sintaxe a seguir.

```
GET /AccountId/provisioned-capacity HTTP/1.1
Host: glacier.Region.amazonaws.com
Date: Date
Authorization: SignatureValue
x-amz-glacier-version: 2012-06-01
```

> **Note**
>
> O `AccountId` valor é o Conta da AWS ID. Esse valor deve corresponder ao Conta da AWS ID associado às credenciais usadas para assinar a solicitação. Você pode especificar um Conta da AWS ID ou, opcionalmente, um único '`-`' (hífen). Nesse caso, o Amazon Glacier usa Conta da AWS o ID associado às credenciais usadas para assinar a solicitação. Se você especificar o ID da conta, não inclua nenhum hífen ('-') no ID.

Parâmetros da solicitação

Essa operação não usa parâmetros de solicitação.

Cabeçalhos de solicitação

Essa operação usa somente os cabeçalhos de solicitação que são comuns a todas as operações. Para informações sobre os cabeçalhos de solicitação comuns, consulte Cabeçalhos de solicitação comuns.

Corpo da Solicitação

Essa operação não tem um corpo de solicitação.

Respostas

Se a operação for bem-sucedida, o serviço reenviará uma resposta `200 OK` HTTP.

Sintaxe da resposta

```
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"
```

```
        }
}
```

Cabeçalhos de resposta

Essa operação usa somente os cabeçalhos de resposta que são comuns a maioria das respostas. Para informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Corpo da resposta

O corpo da resposta contém os seguintes campos JSON.

CapacityId

O ID que identifica a unidade de capacidade provisionada.

Tipo: string

StartDate

A data em que a unidade de capacidade provisionada foi comprada, em Coordinated Universal Time (UTC – Tempo universal coordenado).

Tipo: string Uma representação de string no formato de data ISO 8601, por exemplo `2013-03-20T17:03:43.221Z`.

ExpirationDate

A data em que a unidade de capacidade provisionada expira, em UTC.

Tipo: string Uma representação de string no formato de data ISO 8601, por exemplo `2013-03-20T17:03:43.221Z`.

Erros

Para obter informações sobre exceções e mensagens de erro do Amazon Glacier, consulte Respostas de erro.

Exemplos

O exemplo a seguir lista as unidades de capacidade provisionada para uma conta.

Exemplo de solicitação

Neste exemplo, uma solicitação GET é enviada para recuperar uma lista das unidades de capacidade provisionada para a conta especificada.

```
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
```

Exemplo de resposta

Se a solicitação for bem-sucedida, o Amazon Glacier retornará um `HTTP 200 OK` com uma lista de unidades de capacidade provisionada para a conta, conforme mostrado no exemplo a seguir.

A unidade de capacidade provisionada listada primeiro é um exemplo de unidade com data de início no dia 31 de janeiro de 2017 e data de expiração no dia 28 de fevereiro de 2017. Conforme mencionado anteriormente, se a data de início é dia 31 de um mês, a data de expiração será o último dia do mês seguinte.

```
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" ",
      },
      ...
```

```
}
```

Seções relacionadas

- Purchase Provisioned Capacity (POST provisioned-capacity)

Purchase Provisioned Capacity (POST provisioned-capacity)

Esta operação adquire uma unidade de capacidade provisionada para uma Conta da AWS.

Uma unidade de capacidade provisionada dura um mês a partir da data e hora da compra, que é a data de início. A unidade expira na data de expiração, que é exatamente um mês depois da data de início até o segundo mais próximo.

Se a data de início é dia 31 de um mês, a data de expiração será o último dia do mês seguinte. Por exemplo, se a data de início é dia 31 de agosto, a data de expiração será dia 30 de setembro. Se a data de início é dia 31 de janeiro, a data de expiração será dia 28 de fevereiro.

A capacidade provisionada ajuda a garantir que sua capacidade de recuperação para recuperações expressas esteja disponível quando você precisar dela. Cada unidade de capacidade garante que pelo menos três recuperações rápidas possam ser realizadas a cada cinco minutos e fornece até 150 MB/s de taxa de transferência de recuperação. Para obter mais informações sobre a capacidade provisionada, consulte Opções de recuperação de arquivos.

> **Note**
>
> Há um limite de duas unidades de capacidade provisionada por. Conta da AWS

Solicitações

Para comprar uma unidade de capacidade provisionada para Conta da AWS enviar uma `POST` solicitação HTTP para o URI de capacidade provisionada.

Sintaxe

```
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**
>
> O `AccountId` valor é o Conta da AWS ID. Esse valor deve corresponder ao Conta da AWS ID associado às credenciais usadas para assinar a solicitação. Você pode especificar um Conta da AWS ID ou, opcionalmente, um único '-' (hífen). Nesse caso, o Amazon Glacier usa Conta da AWS o ID associado às credenciais usadas para assinar a solicitação. Se você especificar o ID da conta, não inclua nenhum hífen ('-') no ID.

Parâmetros da solicitação

Cabeçalhos de solicitação

Essa operação usa somente os cabeçalhos de solicitação que são comuns a todas as operações. Para informações sobre os cabeçalhos de solicitação comuns, consulte Cabeçalhos de solicitação comuns.

Corpo da Solicitação

Essa operação não tem um corpo de solicitação.

Respostas

Se a solicitação de operação for bem-sucedida, o serviço retornará uma resposta `201 Created` HTTP.

Sintaxe

```
HTTP/1.1 201 Created
x-amzn-RequestId: x-amzn-RequestId
Date: Date
x-amz-capacity-id: CapacityId
```

Cabeçalhos de resposta

Uma resposta bem-sucedida inclui os seguintes cabeçalhos de resposta, além dos cabeçalhos de resposta que são comuns a todas as operações. Para mais informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Name (Nome)	Description
`x-amz-capacity-id`	O ID que identifica a unidade de capacidade provisionada. Tipo: string

Corpo da resposta

Essa operação não retorna um corpo de resposta.

Erros

Essa operação inclui os seguintes erros, além dos possíveis erros comuns a todas as operações do Amazon Glacier. Para obter informações sobre erros do Amazon Glacier e uma lista de códigos de erro, consulte Respostas de erro.

Código	Description	Código de status HTTP	Tipo
`LimitExceededException`	Retornado se a determinada solicitação excederia o limite da conta de unidades de capacidade provisionada.	`400 Bad Request`	Cliente

Exemplos

O exemplo a seguir compra a capacidade provisionada para uma conta.

Exemplo de solicitação

O exemplo a seguir envia uma solicitação POST HTTP para comprar uma unidade de capacidade provisionada.

```
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
```

Exemplo de resposta

Se a solicitação for bem-sucedida, o Amazon Glacier retornará uma resposta `HTTP 201 Created`, conforme mostrado no exemplo a seguir.

```
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
```

Seções relacionadas

- List Provisioned Capacity (GET provisioned-capacity)

Set Data Retrieval Policy (PUT policy)

Descrição

Esta operação define e impõe uma política de recuperação de dados na região da AWS especificada na solicitação PUT. Você pode definir uma política por região da AWS para uma Conta da AWS. A política é imposta após alguns minutos de uma operação PUT bem-sucedida.

A operação set policy não afeta trabalhos de recuperação que estavam em andamento antes da imposição da política. Para obter mais informações sobre políticas de recuperação de dados, consulte Políticas de recuperação de dados do Amazon Glacier.

Solicitações

Sintaxe

Para definir uma política de recuperação de dados, envie uma solicitação HTTP PUT para o URI da política de recuperação de dados conforme mostrado no exemplo de sintaxe a seguir.

```
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**
>
> O valor de `AccountId` é o ID da Conta da AWS. Esse valor deve corresponder ao ID de Conta da AWS associado às credenciais usadas para assinar a solicitação. Você pode especificar um ID da Conta da AWS ou, de opcionalmente, um '-' (hífen) único. Nesse caso, o Amazon Glacier usa o ID de Conta da AWS associado às credenciais usadas para assinar a solicitação. Se você especificar o ID da conta, não inclua nenhum hífen ('-') no ID.

Parâmetros da solicitação

Essa operação não usa parâmetros de solicitação.

Cabeçalhos de solicitação

Essa operação usa somente os cabeçalhos de solicitação que são comuns a todas as operações. Para informações sobre os cabeçalhos de solicitação comuns, consulte Cabeçalhos de solicitação comuns.

Corpo da Solicitação

O corpo da solicitação contém os campos JSON a seguir.

BytesPerHour

O número máximo de bytes que podem ser recuperados em uma hora.

Este campo somente será obrigatório se o valor do campo Strategy for `BytesPerHour`. A operação PUT será rejeitada se o campo Strategy não for definido como `BytesPerHour` e você definir esse campo.

Tipo: número

Obrigatório: sim, se o campo Strategy for definido como `BytesPerHour`. Do contrário, não.

Valores válidos: valor do inteiro mínimo de 1. Valor do inteiro máximo de 2^63 - 1 inclusive.

Regras

A regra da política. Embora esse seja um tipo de lista, atualmente deve haver somente uma regra, que contém um campo Strategy e, como opção, um campo BytesPerHour.

Tipo: matriz

Obrigatório: Sim

Strategy

O tipo de política de recuperação de dados a ser definido.

Tipo: string

Obrigatório: Sim

Valores válidos: `BytesPerHour`|`FreeTier`|None. `BytesPerHour` é equivalente a selecionar Max Retrieval Rate (Taxa de recuperação máxima) no console. `FreeTier` é equivalente a selecionar Somente nível gratuito no console. None é equivalente a selecionar No Retrieval Policy (Nenhuma política de recuperação) no console. Para obter mais informações sobre como selecionar políticas de recuperação de dados no console, consulte Políticas de recuperação de dados do Amazon Glacier.

Respostas

Sintaxe

```
HTTP/1.1 204 No Content
```

```
x-amzn-RequestId: x-amzn-RequestId
Date: Date
```

Cabeçalhos de resposta

Essa operação usa somente os cabeçalhos de resposta que são comuns a maioria das respostas. Para informações sobre os cabeçalhos de resposta comuns, consulte Cabeçalhos de resposta comuns.

Corpo da resposta

Essa operação não retorna um corpo de resposta.

Erros

Para obter informações sobre exceções e mensagens de erro do Amazon Glacier, consulte Respostas de erro.

Exemplos

Exemplo de solicitação

O exemplo a seguir envia uma solicitação HTTP PUT com o campo Strategy definido como `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
          }
       ]
    }
```

```
}
```

O exemplo a seguir envia uma solicitação HTTP PUT com o campo Strategy definido como `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"
          }
       ]
    }
}
```

O exemplo a seguir envia uma solicitação HTTP PUT com o campo Strategy definido como `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"
          }
       ]
    }
```

```
}
```

Exemplo de resposta

Se a solicitação for bem-sucedida, o Amazon Glacier define a política e retorna uma resposta HTTP `204 No Content`, conforme mostrado no exemplo a seguir.

```
HTTP/1.1 204 No Content
x-amzn-RequestId: AAABZpJrTyioDC_HsOmHae8EZp_uBSJr6cnGOLKp_XJCl-Q
Date: Wed, 10 Feb 2017 12:02:00 GMT
```

Seções relacionadas

- Política de recuperação para obter dados (GET policy)
- Initiate Job (trabalhos POST)

Histórico do documento

- Versão atual do produto: 01-06-2012

A tabela a seguir descreve as alterações importantes em cada versão do Guia do desenvolvedor do Amazon Glacier de 5 de julho de 2018 em diante. Para receber notificações sobre atualizações dessa documentação, é possível inscrever-se em um feed RSS.

Alteração	Descrição	Data
Horários de início aprimorados para solicitações de restauração padrão feitas por meio do Operações em Lote do S3	As recuperações padrão para solicitações de restauração feitas por meio do Operações em Lote do S3 agora podem começar em minutos. Para receber mais informações, consulte Opções de recuperação de arquivo.	9 de agosto de 2023
O Amazon S3 oferece suporte a taxas de solicitações de restauração mais altas para S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive	O Amazon S3 oferece suporte a solicitações de restauração a uma taxa de até 1.000 transações por segundo, por Conta da AWS , para as classes de armazenamento S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive.	15 de novembro de 2022
Mudança de nome do Amazon Glacier	O Amazon Glacier agora é Amazon Glacier para melhor refletir a integração do Glacier com o Amazon S3.	20 de novembro de 2018
Atualizações agora disponíveis em RSS	Agora, você pode assinar um feed RSS para receber notificações sobre atualizaç	5 de julho de 2018

ões do Guia do desenvolvedor do Amazon Glacier.

Atualizações anteriores

A tabela a seguir descreve as alterações importantes em cada versão do Guia do Desenvolvedor do Amazon Glacier de antes de 5 de julho de 2018.

Alteração	Description	Data de lançamento
Recuperações de dados expressas e em massa	O Amazon Glacier agora dá suporte às recuperações de dados expressas e em massa, além de recuperaç ões padrão. Para obter mais informações, consulte Opções de recuperação de arquivos.	21 de novembro de 2016
Vault Lock	Agora, o Amazon Glacier oferece suporte ao Vault Lock, que permite implantar e aplicar facilmente controles de conformidade em cofres do Amazon Glacier individuais com uma política do Vault Lock. Para obter mais informações, consulte Vault Lock do Amazon Glacier e Políticas do Vault Lock.	8 de julho de 2015
Marcação do cofre	Agora, o Amazon Glacier permite marcar os cofres do Amazon Glacier para um gerenciamento mais fácil de recursos e custos. As tags são rótulos que você pode definir e associar aos seus cofres, e o uso de tags adiciona recursos de filtragem às operações, como AWS relatórios de custos. Para obter mais informaçõ es, consulte Marcação de recursos do Amazon Glacier e Marcar seus Cofres do Amazon Glacier.	22 de junho de 2015
Políticas de acesso a cofres	Agora, o Amazon Glacier oferece suporte ao gerenciamento de acesso aos cofres do Amazon Glacier individuais usando políticas de acesso ao cofre. Agora você pode definir uma política de acesso diretamente em um cofre, o que facilita a	27 de abril de 2015

Alteração	Description	Data de lançamento
	concessão de acesso ao cofre para usuários e grupos de negócios internos da organização, bem como para parceiros de negócios externos. Para obter mais informações, consulte Políticas de acesso ao cofre.	
Políticas de recuperação de dados e registro em log de auditoria	O Amazon Glacier agora dá suporte a políticas de recuperação de dados e registro em log de auditoria . As políticas de recuperação de dados permitem definir facilmente limites de recuperação de dados e simplificar o gerenciamento de custos da recuperação de dados. Você pode definir seus próprios limites de recuperação de dados com alguns cliques no Console de gerenciamento da AWS ou usando a API do Amazon Glacier. Para obter mais informações, consulte Políticas de recuperação de dados do Amazon Glacier. Além disso, o Amazon Glacier agora oferece suporte ao registro de auditoria com AWS CloudTrail, que registra as chamadas de API do Amazon Glacier para sua conta e entrega os arquivos de log em um bucket do Amazon S3 que você especificar. Para obter mais informações, consulte Registro de chamadas de API do Amazon Glacier com AWS CloudTrail.	11 de dezembro de 2014
Atualizações feitas em amostras do Java	Atualizadas as amostras de código do Java neste guia que usam AWS SDK para Java.	27 de junho de 2014
Limitar recuperação do inventário de cofre	Agora você pode limitar o número de itens do inventário de cofre recuperados filtrando a data de criação do arquivo ou definindo um limite. Para obter mais informações sobre como limitar a recuperação do inventário, consulte Recuperação do inventário de intervalo no tópico Initiate Job (trabalhos POST).	31 de dezembro de 2013

Alteração	Description	Data de lançamento
Removido e desatualizado URLs	Removido o URLs que apontava para a antiga página de credenciais de segurança dos exemplos de código.	26 de julho de 2013
Suporte para recuperações de intervalos	O Amazon Glacier agora dá suporte à recuperação de intervalos específicos dos arquivos. Você pode iniciar um trabalho solicitando ao Amazon Glacier para preparar um arquivo inteiro ou uma parte dele para download subsequente. Quando um arquivo é muito grande, você pode achar mais econômico iniciar vários trabalhos sequenciais a fim de preparar o arquivo. Para obter mais informações, consulte Baixando um arquivo no Amazon Glacier.	13 de novembro de 2012
Novo guia	Esta é a primeira versão do Guia do desenvolvedor do Amazon Glacier.	20 de agosto de 2012

Glossário da AWS

Para obter a terminologia mais recente da AWS, consulte o glossário da AWS na Referência do Glossário da AWS.