

---

# Amazon Simple Storage Service

Guia do desenvolvedor

Versão da API 2006-03-01



## Amazon Simple Storage Service: Guia do desenvolvedor

Copyright © 2019 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

## Table of Contents

O que é o Amazon S3? .....	1
Como eu faço? .....	1
Introdução .....	2
Visão geral do Amazon S3 e este guia .....	2
Vantagens do Amazon S3 .....	2
Conceitos do Amazon S3 .....	3
Buckets .....	3
Objetos .....	3
Chaves .....	3
Regiões .....	4
Modelo de consistência de dados do Amazon S3 .....	4
Recursos do Amazon S3 .....	6
Classes de armazenamento .....	6
Políticas de buckets .....	6
AWS Identity and Access Management .....	7
Listas de controle de acesso .....	7
Versionamento .....	8
Operações .....	8
Interfaces de programação de aplicativos (APIs) do Amazon S3 .....	8
A interface REST .....	8
A interface SOAP .....	9
Pagar pelo Amazon S3 .....	9
Serviços relacionados .....	9
Fazer solicitações .....	10
Sobre as chaves de acesso .....	10
Chaves de acesso da conta da AWS .....	10
Chaves de acesso do usuário do IAM .....	10
Credenciais de segurança temporárias .....	11
Endpoints de solicitações .....	11
Fazer solicitações por meio do IPv6 .....	12
Conceitos básicos do IPv6 .....	12
Usar endereços do IPv6 em políticas do IAM .....	13
Testar a compatibilidade com endereços IP .....	14
Usar endpoints de pilha dupla .....	14
Fazer solicitações usando os SDKs da AWS .....	18
Usar credenciais de usuário do IAM ou da conta da AWS .....	18
Usar credenciais temporárias de usuário do IAM .....	25
Usar credenciais temporárias de usuário federado .....	34
Fazer solicitações usando a API REST .....	45
Endpoints de pilha dupla (API REST) .....	46
Hospedagem virtual de buckets .....	46
Redirecionamento de solicitação e a API REST .....	51
Buckets .....	54
Criação de um bucket .....	54
Sobre permissões .....	56
Acesso a um bucket .....	56
Opcões de configuração de bucket .....	57
Restrições e limitações .....	59
Regras para nomeação .....	59
Exemplos de criação de um bucket .....	60
Usar o console do Amazon S3 .....	61
Usar o AWS SDK for Java .....	61
Usar o AWS SDK para .NET .....	62
Uso do AWS SDK para Ruby Versão 3 .....	63

Uso de outros AWS SDKs .....	63
Exclusão ou esvaziamento do bucket .....	63
Excluir um Bucket .....	64
Esvaziar um bucket .....	66
Criptografia padrão para um bucket .....	68
Como configurar a criptografia padrão de bucket do Amazon S3 .....	69
Mudar das políticas de bucket para a criptografia padrão para realização de criptografia .....	69
Usar a criptografia padrão com a replicação entre regiões .....	69
Monitoramento da criptografia padrão com o CloudTrail e o CloudWatch .....	70
Mais informações .....	71
Configuração de site de bucket .....	71
Usar o Console de gerenciamento da AWS .....	71
Usar o AWS SDK for Java .....	71
Usar o AWS SDK para .NET .....	73
Usar o SDK para PHP .....	74
Uso dos REST API .....	75
Transfer Acceleration .....	75
Por que usar o Transfer Acceleration? .....	76
Conceitos básicos .....	76
Requisitos para usar o Amazon S3 Transfer Acceleration .....	77
Exemplos do Transfer Acceleration .....	78
Buckets de Pagamento pelo solicitante .....	83
Configurar com o console .....	83
Configurar com a API REST .....	84
Detalhes da cobrança .....	86
Controle de acesso .....	86
Relatórios de uso e faturamento .....	86
Relatórios de faturamento .....	87
Relatório de uso .....	89
Entender os relatórios de uso e faturamento .....	91
Usar tags de alocação de custos .....	99
Objetos .....	101
Chave de objeto e metadados .....	102
Chaves de objeto .....	102
Metadados do objeto .....	104
Classes de armazenamento .....	107
Classes de armazenamento de objetos acessados com frequência .....	107
Classe de armazenamento que otimiza automaticamente objetos muito e pouco acessados .....	108
Classes de armazenamento de objetos acessados com pouca frequência .....	108
Comparar as classes de armazenamento do Amazon S3 .....	110
Configurar a classe de armazenamento de um objeto .....	110
Sub-recursos .....	111
Versionamento .....	111
Marcação de objetos .....	114
Operações de API relacionadas à marcação de objetos .....	115
Marcação de objetos e informações adicionais .....	116
Gerenciamento de tags de objeto .....	119
Gerenciamento de ciclo de vida .....	122
Quando devo usar a configuração de ciclo de vida? .....	123
Como configuro um ciclo de vida? .....	123
Considerações adicionais .....	124
Elementos de configuração do ciclo de vida .....	130
Exemplos de configuração de ciclo de vida .....	137
Definir a configuração do ciclo de vida .....	147
Cross-Origin Resource Sharing (CORS, Compartilhamento de recursos de origem cruzada) .....	156
Compartilhamento de recursos de origem cruzada: cenários de caso de uso .....	156
Como faço para configurar CORS no meu bucket? .....	156

Como o Amazon S3 avalia a configuração de CORS em um bucket?	159
Ativação do CORS	159
Solução de problemas do CORS	165
Operações em objetos	165
Obtenção de objetos	166
Upload de objetos	175
Cópia de objetos	219
Lista de chaves de objeto	230
Excluir objetos	237
Selecionar conteúdo de objetos	255
Restaurar objetos arquivados	259
Consultar objetos arquivados	263
Análise de classe de armazenamento	267
Como configurar a análise de classe de armazenamento	267
Análise de classe de armazenamento	268
Como posso exportar dados de análise de classe de armazenamento?	271
Layout de arquivo de exportação de análise de classe de armazenamento do	272
APIs REST de análise do Amazon S3	272
Inventário do	273
Como configurar o inventário do Amazon S3	273
Buckets de inventário do Amazon S3	273
Configurar o inventário do Amazon S3	274
Listas de inventário	276
Consistência de inventário	277
Local das listas de inventário	277
O que é um manifesto de inventário?	278
Notificar quando o inventário estiver completo	280
>Consultar o inventário com o Athena	280
APIs REST de inventário do Amazon S3	281
Gerenciamento de acesso	282
Introdução	282
Visão geral	283
Como o Amazon S3 autoriza uma solicitação	288
Diretrizes para usar as opções disponíveis de política de acesso	293
Demonstrações com exemplo: gerenciar acesso	297
Uso de políticas de bucket e políticas de usuário	326
Visão geral da linguagem da política de acesso	326
Exemplos de políticas de bucket	358
Exemplos de política de usuário	367
Gerenciar o acesso com ACLs	390
Visão geral da Lista de controle de acesso (ACL)	390
Gerenciar ACLs	396
Bloquear acesso público	402
Configurações do Block Public Access	403
O significado de "público"	404
Permissões	406
Exemplos	406
Proteger dados	409
Criptografia de dados	409
Criptografia do lado do servidor	410
Criptografia no lado do cliente	440
Versionamento	448
Como configurar o versionamento de um bucket	449
Exclusão de MFA	449
Tópicos relacionados	450
Exemplos	451
Gerenciamento de objetos em um bucket com versionamento ativado	453

Gerenciamento de objetos em um bucket com versionamento suspenso .....	467
Bloquear objetos .....	470
Visão geral .....	471
Gerenciar bloqueios de objeto .....	474
Operações em lote .....	477
Terminologia .....	477
Os elementos básicos: trabalhos .....	478
Como especificar um manifesto .....	478
Criar um trabalho .....	479
Criar solicitação de tarefa .....	479
Criar resposta da tarefa .....	480
Conceder permissões para operações em lote .....	480
Recursos relacionados .....	483
Operações .....	483
PUT Object Copy (Copiar Objeto PUT) .....	486
Iniciar a restauração de um objeto .....	486
Invocar uma função do Lambda .....	487
Put Object ACL (ACL de objeto PUT) .....	488
Put Object Tagging (Marcação de objeto PUT) .....	488
Gerenciar trabalhos .....	489
Listar os trabalhos .....	489
Visualizar detalhes do trabalho .....	490
Como atribuir prioridade aos trabalhos .....	490
Status do trabalho .....	490
Como monitorar falhas nos trabalhos .....	492
Notificações e registro em log .....	493
Relatórios de conclusão .....	493
Hospedagem de sites estáticos .....	494
Endpoints de site .....	495
Principais diferenças entre o site da Amazon e o endpoint de API REST .....	495
Configuração de bucket para hospedagem de site .....	496
Habilitar a hospedagem de sites .....	496
Configuração de suporte a documento de índice .....	497
Permissões necessárias para acesso ao site .....	499
(Opcional) Configurar o registro em log de tráfego da web .....	500
(Opcional) Suporte a documento de erro personalizado .....	500
(Opcional) Configuração de um redirecionamento .....	502
Apresentações de exemplo .....	509
Exemplo: configuração de um site estático .....	509
Exemplo: configurar um site estático usando um domínio personalizado .....	511
Exemplo: acelere seu site com o Amazon CloudFront .....	517
Apagar recursos de exemplo .....	520
Notificações .....	522
Visão geral .....	522
Como habilitar notificações de evento .....	524
Tipos e destinos de notificações de evento .....	525
Tipos de evento compatíveis .....	525
Destinos compatíveis .....	526
Configurar notificações com filtragem de nomes de chaves de objetos .....	527
Exemplos de configurações válidas de notificação com filtragem de nome de chave de objeto .....	527
Exemplos de configurações de notificação com sobreposição inválida de prefixo/sufixo .....	529
Conceder permissões para publicar mensagens de notificação de vento a um destino .....	531
Conceder permissões para invocar uma função do AWS Lambda .....	531
Conceder permissões para publicar mensagens em um tópico do SNS ou em uma fila do SQS ....	532
Passo a passo do exemplo 1 .....	533
Resumo do passo a passo .....	534
Etapa 1: Criar um tópico do Amazon SNS .....	534

Etapa 2: Criar uma fila do Amazon SQS .....	535
Etapa 3: Adicionar a configuração de notificação ao bucket .....	536
Etapa 4: Testar a configuração .....	539
Passo a passo do exemplo 2 .....	539
Estrutura de mensagens de evento .....	539
Replicação entre regiões .....	544
Quando usar CRR .....	544
Requisitos para CRR .....	544
O que o Amazon S3 replica? .....	545
O que é replicado? .....	545
O que não é replicado? .....	546
Tópicos relacionados .....	547
Visão geral da configuração da CRR .....	547
Visão geral da configuração da replicação .....	548
Configurar permissões para CRR .....	556
Configurações adicionais de CRR .....	560
Configuração adicional da CRR: alteração do proprietário da réplica .....	560
Configuração adicional da CRR: replicação de objetos criptografados .....	563
Demonstrações da CRR .....	567
Exemplo 1 de CRR: Mesma conta da AWS .....	568
Exemplo 2 de CRR: diferentes contas da AWS .....	575
Exemplo 3 da CRR: alteração do proprietário da réplica .....	576
Exemplo 4 de CRR: Replicar objetos criptografados .....	580
CRR: informações do status .....	585
Tópicos relacionados .....	586
CRR: solução de problemas .....	586
Tópicos relacionados .....	587
CRR: considerações adicionais .....	587
Configuração de ciclo de vida e réplicas de objeto .....	588
Configuração do versionamento e configuração de replicação .....	588
Configuração de log e de replicação .....	588
CRR e região de destino .....	589
Pausar a configuração de replicação .....	589
Tópicos relacionados .....	589
Roteamento de solicitação .....	590
Redirecionamento de solicitação e a API REST .....	590
Roteamento de DNS .....	590
Redirecionamento de solicitação temporário .....	591
Redirecionamento permanente de solicitação .....	593
Exemplos de redirecionamento de solicitação .....	593
Considerações de DNS .....	594
Otimização do desempenho .....	595
Orientações sobre desempenho e taxa de solicitações .....	595
Cargas de trabalho que usam muito GET .....	595
Escalabilidade da janela de TCP .....	595
Reconhecimento seletivo de TCP .....	596
Monitoramento .....	597
Ferramentas de monitoramento .....	597
Ferramentas automatizadas .....	597
Ferramentas manuais .....	598
Métricas de monitoramento com o CloudWatch .....	598
Métricas e dimensões .....	599
Métricas diárias de armazenamento por buckets do CloudWatch do Amazon S3 .....	599
Métricas de solicitação do CloudWatch do Amazon S3 .....	600
Amazon S3 CloudWatch Dimensões .....	602
Acesso às métricas do CloudWatch .....	603
Recursos relacionados .....	604

Configurações de métricas para buckets .....	604
Entrega com melhor esforço de métricas do CloudWatch .....	604
Filtrar configurações de métricas .....	605
Como adicionar configurações de métricas .....	605
Registrar em log chamadas à API com o AWS CloudTrail .....	606
Informações do Amazon S3 no CloudTrail .....	606
Usar os logs do CloudTrail com os logs de acesso ao servidor do Amazon S3 e com o CloudWatch Logs .....	611
Exemplo: entradas do arquivo de log do Amazon S3 .....	611
Recursos relacionados .....	613
BitTorrent .....	614
Como você será cobrado pela entrega de BitTorrent .....	614
Uso do BitTorrent para recuperar objetos armazenados no Amazon S3 .....	615
Publicação de conteúdo com o Amazon S3 e o BitTorrent .....	616
Como tratar erros .....	617
A resposta de erro de REST .....	617
Cabeçalhos de resposta .....	617
Resposta de erro .....	618
A resposta de erro de SOAP .....	619
Melhores práticas para erros do Amazon S3 .....	619
Tentar InternalErrors novamente .....	619
Ajuste o aplicativo para erros repetidos de SlowDown .....	619
Erros isolados .....	620
Solução de problemas do Amazon S3 .....	621
Solucionar problemas do Amazon S3 por sintoma .....	621
Aumentos significativos em respostas HTTP 503 para solicitações para buckets com versionamento habilitado .....	621
Comportamento inesperado ao acessar buckets definidos com CORS .....	621
Obter os IDs da solicitação do Amazon S3 para o AWS Support .....	622
Usar o HTTP para obter IDs de solicitação .....	622
Usar um navegador da web para obter IDs de solicitação .....	622
Usar os AWS SDKs para obter IDs de solicitação .....	623
Usar o AWS CLI para obter IDs de solicitação .....	624
Tópicos relacionados .....	624
Registro de acesso em logs ao servidor .....	625
Como habilitar o registro em log de acesso ao servidor .....	625
Formato da chave de objeto de log .....	626
Como os logs são entregues? .....	627
Entrega de logs pelo servidor de melhor esforço .....	627
As alterações do status do registro de bucket em logs entram em vigor ao longo do tempo .....	627
Habilitar o registro usando o console .....	627
Habilitar o log por programação .....	628
Habilitar registro em log .....	628
Conceder as permissões WRITE e READ_ACP ao grupo de Entrega de logs .....	629
Exemplo: AWS SDK para .NET .....	629
Mais informações .....	631
Formato do log .....	631
Informações personalizadas do log de acesso .....	634
Considerações de programação para o formato do log de acesso ao servidor extensível .....	635
Registro em log adicional para operações de cópia .....	635
Excluir arquivos de log .....	638
Mais informações .....	638
AWS SDKs e Explorers .....	639
Especificar a versão da assinatura na autenticação de solicitações .....	640
Substituição do AWS Signature versão 2 para o Amazon S3 .....	641
Migração do Signature versão 2 para o Signature versão 4 .....	642
Configurar a CLI da AWS .....	645

Usar o AWS SDK for Java .....	646
A organização da API Java .....	647
Testar os exemplos de código Java do Amazon S3 .....	647
Usar o AWS SDK para .NET .....	647
A organização da API .NET .....	648
Executar os exemplos de código .NET do Amazon S3 .....	648
Usar o AWS SDK para PHP e executar exemplos do PHP .....	649
Níveis do AWS SDK para PHP .....	649
Executar exemplos do PHP .....	649
Recursos relacionados .....	650
Usar o AWS SDK para Ruby - versão 3 .....	650
A organização da API Ruby .....	650
Testar os exemplos de script do Ruby .....	650
Usar o AWS SDK for Python (Boto) .....	651
Usar os AWS Mobile SDKs para iOS e Android .....	651
Mais informações .....	651
Usar a biblioteca JavaScript do AWS Amplify .....	651
Mais informações .....	652
Apêndices .....	653
Apêndice A: uso da API SOAP .....	653
Elementos comuns da API SOAP .....	653
Como autenticar solicitações SOAP .....	654
Configurar políticas de acesso padrão com SOAP .....	655
Apêndice B: autenticação de solicitações (versão 2 do AWS Signature) .....	656
Autenticar solicitações usando a API REST .....	657
Assinar e autenticar as solicitações REST .....	659
Uploads baseados no navegador usando POST .....	668
Recursos .....	684
Referência SQL .....	685
Comando SELECT .....	685
Lista SELECT .....	685
Cláusula FROM .....	685
Cláusula WHERE .....	689
Cláusula LIMIT (apenas Amazon S3 Select) .....	689
Acesso de atributo .....	689
Diferenciação de letras maiúsculas e minúsculas de cabeçalho/nomes de atributo .....	690
Usar palavras-chave reservadas como termos definidos pelo usuário .....	691
Expressões escalares .....	691
Tipos de dados .....	692
Conversões de tipo de dados .....	692
Tipos de dados compatíveis .....	692
Operadores .....	693
Operadores lógicos .....	693
Operadores de comparação .....	693
Operadores de correspondência de padrões .....	693
Operadores matemáticos .....	693
Precedência do operador .....	694
Palavras-chave reservadas .....	694
Funções SQL .....	698
Funções agregadas (apenas Amazon S3 Select) .....	698
Funções condicionais .....	699
Funções da conversão .....	700
Funções de data .....	701
Funções de string .....	707
Histórico do documento .....	710
Atualizações anteriores .....	712
AWS Glossary .....	729

# O que é o Amazon S3?

O Amazon Simple Storage Service é armazenamento para a Internet. Ele foi projetado para facilitar a computação de escala na web para os desenvolvedores.

O Amazon S3 tem uma interface simples de serviços da web que você pode usar para armazenar e recuperar qualquer quantidade de dados, a qualquer momento, em qualquer lugar da web. Ela concede acesso a todos os desenvolvedores para a mesma infraestrutura altamente dimensionável, confiável, segura, rápida e econômica que a Amazon utiliza para rodar a sua própria rede global de sites da web. O serviço visa maximizar os benefícios de escala e poder passar esses benefícios para os desenvolvedores.

Este guia explica os conceitos principais do Amazon S3, como buckets e objetos, e como trabalhar com esses recursos usando a interface de programação de aplicativo (API) do Amazon S3.

## Como eu faço?

Informações	Seções relevantes
Visão geral e definição de preço do produto	<a href="#">Amazon S3</a>
Obtenha uma introdução rápida e prática ao Amazon S3	<a href="#">Guia de conceitos básicos do Amazon Simple Storage Service</a>
Saiba mais sobre terminologia e conceitos principais do Amazon S3	<a href="#">Introdução ao Amazon S3 (p. 2)</a>
Como faço para trabalhar com buckets?	<a href="#">Trabalho com buckets do Amazon S3 (p. 54)</a>
Como faço para trabalhar com objetos?	<a href="#">Trabalho com objetos do Amazon S3 (p. 101)</a>
Como faço solicitações?	<a href="#">Fazer solicitações (p. 10)</a>
Como faço para gerenciar o acesso aos recursos?	<a href="#">Gerenciamento de permissões de acesso aos recursos do Amazon S3 (p. 282)</a>

# Introdução ao Amazon S3

Essa introdução ao Amazon Simple Storage Service tem o objetivo de fornecer um resumo detalhado desse serviço da web. Depois de ler esta seção, você deverá ter uma boa ideia do que ele oferece e de como ele pode ser ajustado em sua empresa.

## Tópicos

- [Visão geral do Amazon S3 e este guia \(p. 2\)](#)
- [Vantagens do Amazon S3 \(p. 2\)](#)
- [Conceitos do Amazon S3 \(p. 3\)](#)
- [Recursos do Amazon S3 \(p. 6\)](#)
- [Interfaces de programação de aplicativos \(APIs\) do Amazon S3 \(p. 8\)](#)
- [Pagar pelo Amazon S3 \(p. 9\)](#)
- [Serviços relacionados \(p. 9\)](#)

## Visão geral do Amazon S3 e este guia

O Amazon S3 tem uma interface simples de serviços da web que você pode usar para armazenar e recuperar qualquer quantidade de dados, a qualquer momento, em qualquer lugar da web.

Este guia descreve como você envia solicitações para criar buckets, armazenar e recuperar objetos e gerenciar permissões em seus recursos. O guia também descreve o controle de acesso e o processo de autenticação. O controle de acesso define quem pode acessar objetos e buckets no Amazon S3 e o tipo de acesso (por exemplo, LEITURA e GRAVAÇÃO). O processo de autenticação verifica a identidade de um usuário que está tentando acessar o Amazon Web Services (AWS).

## Vantagens do Amazon S3

O Amazon S3 foi desenvolvido intencionalmente com um conjunto mínimo de recursos com foco em simplicidade e robustez. As seguintes são algumas das vantagens do serviço do Amazon S3:

- Criar buckets – crie e nomeie um bucket que armazena dados. Os buckets são o contêiner fundamental no Amazon S3 para armazenamento de dados físico.
- Armazenar dados em buckets – armazene uma quantidade infinita de dados em um bucket. Carregue quantos objetos desejar em um bucket do Amazon S3. Cada objeto pode conter até 5 TB de dados. Cada objeto é armazenado e recuperado usando uma chave exclusiva atribuída pelo desenvolvedor.
- Baixar dados – baixe seus dados ou permita que outras pessoas os baixem. Baixe seus dados a qualquer momento ou permita que outros façam o mesmo.
- Permissões – conceda ou negue permissões a outras pessoas que desejam carregar ou baixar dados no bucket do Amazon S3. Conceda permissões para upload ou para download a três tipos de usuário. Os mecanismos de autenticação podem ajudar a manter os dados protegidos contra acesso não autorizado.
- Interfaces padrão – use as interfaces REST e SOAP baseadas em padrão desenvolvidas para funcionar com qualquer toolkit de desenvolvimento da Internet.

### Note

O suporte de SOAP via HTTP está obsoleto, mas continua disponível via HTTPS. Novos recursos do Amazon S3 não serão compatíveis com o SOAP. Recomendamos que você use o REST API ou os SDKs da AWS.

## Conceitos do Amazon S3

### Tópicos

- [Buckets \(p. 3\)](#)
- [Objetos \(p. 3\)](#)
- [Chaves \(p. 3\)](#)
- [Regiões \(p. 4\)](#)
- [Modelo de consistência de dados do Amazon S3 \(p. 4\)](#)

Esta seção descreve os conceitos básicos e a terminologia que você precisa entender para usar o Amazon S3 de maneira efetiva. Eles são apresentados na ordem em que você muito provavelmente os encontrará.

## Buckets

Um bucket é um contêiner para objetos armazenados no Amazon S3. Cada objeto está contido em um bucket. Por exemplo, se o objeto chamado `photos/puppy.jpg` estiver armazenado no bucket `johndoe`, ele poderá ser endereçado usando a URL `http://johndoe.s3.amazonaws.com/photos/puppy.jpg`

Os buckets têm várias finalidades: eles organizam o namespace do Amazon S3 no nível mais alto, identificam a conta responsável pelas cobranças de armazenamento e transferência de dados, têm uma função no controle de acesso e servem como a unidade de agregação para relatórios de uso.

Você pode configurar os buckets para que sejam criados em uma região específica. Para obter mais informações, consulte [Buckets e regiões \(p. 56\)](#). Você também pode configurar um bucket para que sempre que um objeto for adicionado, o Amazon S3 gere um ID exclusivo de versão e o atribua ao objeto. Para obter mais informações, consulte [Versionamento \(p. 448\)](#).

Para obter mais informações sobre buckets, consulte [Trabalho com buckets do Amazon S3 \(p. 54\)](#).

## Objetos

Os objetos são as entidades fundamentais armazenadas no Amazon S3. Os objetos consistem em metadados e dados de objeto. A porção de dados não é visível para o Amazon S3. Os metadados são um conjunto de pares de nome e valor que descrevem o objeto. Incluem alguns metadados padrão, como a data da última modificação e metadados HTTP padrão, como Content-Type. Você também pode especificar metadados personalizados no momento em que o objeto é armazenado.

Um objeto é identificado exclusivamente em um bucket por uma chave (nome) e um ID de versão. Para obter mais informações, consulte [Chaves \(p. 3\)](#) e [Versionamento \(p. 448\)](#).

## Chaves

Uma chave é um identificador exclusivo de um objeto em um bucket. Cada objeto em um bucket tem exatamente uma chave. Como a combinação de um bucket, chave e ID de versão identifica

exclusivamente cada objeto, o Amazon S3 pode ser considerado como um mapa de dados básico entre "bucket + chave + versão" e o próprio objeto. Cada objeto no Amazon S3 pode ser endereçado exclusivamente por meio da combinação do endpoint de serviço da web, do nome de bucket, da chave e, opcionalmente, de uma versão. Por exemplo, na URL do <http://doc.s3.amazonaws.com/2006-03-01/AmazonS3.wsdl>, "doc" é o nome do bucket e "2006-03-01/AmazonS3.wsdl" é a chave.

Para obter mais informações sobre chaves de objeto, consulte [Chaves de objeto](#).

## Regiões

Você pode escolher a região geográfica onde o Amazon S3 armazenará os buckets que você cria. É possível escolher uma região para otimizar a latência, minimizar os custos ou atender a requisitos regulatórios. Os objetos armazenados em uma região nunca saem dela, a não ser que você os transfira explicitamente para outra região. Por exemplo, os objetos armazenados na região UE (Irlanda) nunca saem dela.

Para obter uma lista de endpoints e regiões do Amazon S3 disponíveis, consulte [Regiões e endpoints](#) na Referência geral da AWS.

## Modelo de consistência de dados do Amazon S3

O Amazon S3 fornece consistência de leitura após gravação para PUTS de novos objetos em seu bucket do S3 em todas as regiões com uma advertência. A advertência é que se você fizer uma solicitação HEAD ou GET para o nome da chave (para saber se o objeto existe) antes de criar o objeto, o Amazon S3 fornecerá consistência eventual para leitura após gravação.

O Amazon S3 oferece consistência eventual para substituir PUTS e DELETES em todas as regiões.

As atualizações em uma única chave são atômicas. Por exemplo, se você executar PUT para uma chave existente, uma leitura subsequente poderá retornar os dados antigos ou os dados atualizados, mas nunca retornará dados corrompidos ou parciais.

O Amazon S3 atinge alta disponibilidade replicando dados entre vários servidores nos datacenters da Amazon. Se uma solicitação PUT for bem-sucedida, os dados serão armazenados com segurança. No entanto, as informações sobre as alterações devem ser replicadas no Amazon S3, o que pode demorar algum tempo e, portanto, você pode observar os seguintes comportamentos:

- Um processo grava um novo objeto no Amazon S3 e imediatamente lista as chaves em seu bucket. Até que a alteração seja totalmente propagada, o objeto poderá não aparecer na lista.
- Um processo substitui um objeto existente e imediatamente tenta lê-lo. Até que a alteração seja totalmente propagada, o Amazon S3 poderá retornar os dados anteriores.
- Um processo exclui um objeto existente e imediatamente tenta lê-lo. Até que a exclusão seja totalmente propagada, o Amazon S3 poderá retornar os dados excluídos.
- Um processo exclui um objeto existente e imediatamente lista as chaves em seu bucket. Até que a exclusão seja totalmente propagada, o Amazon S3 poderá listar o objeto excluído.

### Note

No momento, o Amazon S3 não oferece suporte a bloqueio de objeto. Se duas solicitações PUT forem realizadas simultaneamente na mesma chave, a solicitação com o time stamp mais recente será a escolhida. Se isso for um problema, você precisará criar um mecanismo de bloqueio de objetos em seu aplicativo.

As atualizações são baseadas em chaves. Não há possibilidade de realizar atualizações atômicas entre chaves. Por exemplo, você não pode tornar a atualização de uma chave dependente da atualização de outra chave a menos que você desenvolva essa funcionalidade em seu aplicativo.

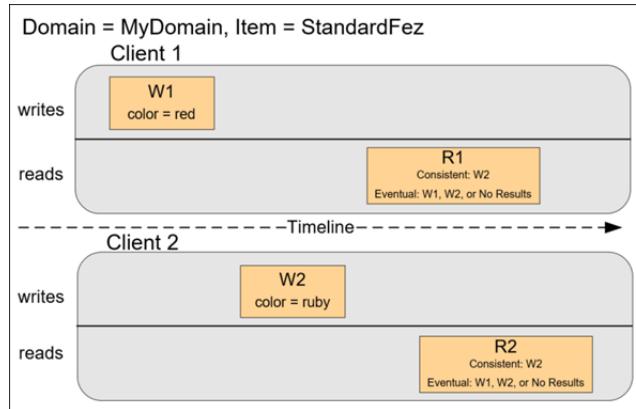
A tabela a seguir descreve as características de leitura eventualmente consistente e de leitura consistente.

Leitura eventualmente consistente	Leitura consistente
Leituras obsoletas possíveis	Sem leituras obsoletas
Menor latência de leitura	Potencial latência mais alta de leitura
Throughput mais alto de leitura	Potencial throughput mais baixo de leitura

## Aplicativos simultâneos

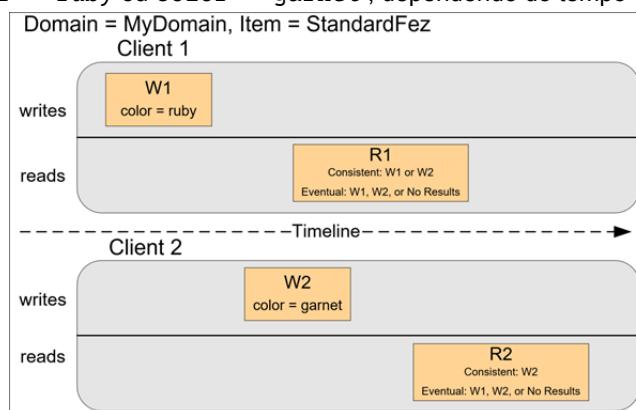
Esta seção fornece exemplos de solicitações de leitura eventualmente consistente e de leitura consistente quando vários clientes estão gravando nos mesmos itens.

Neste exemplo, W1 (gravação 1) e W2 (gravação 2) são concluídas antes do início de R1 (leitura 1) e R2 (leitura 2). Para uma leitura consistente, R1 e R2 retornam `color = ruby`. Para uma leitura eventualmente consistente, R1 e R2 podem retornar `color = red` ou `color = ruby`, dependendo do tempo decorrido.

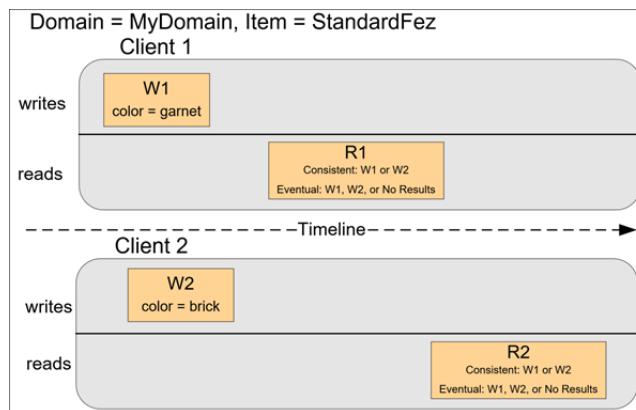


No próximo exemplo, a W2 não é encerrada antes do início da R1. Portanto, a R1 pode retornar `color = ruby` ou `color = garnet` para uma leitura consistente ou para uma leitura eventualmente consistente. Além disso, dependendo do tempo decorrido, uma leitura eventualmente consistente poderá retornar nenhum resultado.

Para uma leitura consistente, R2 retorna `color = garnet`. Para uma leitura eventualmente consistente, R2 pode retornar `color = ruby` ou `color = garnet`, dependendo do tempo decorrido.



No último exemplo, o cliente 2 realiza a W2 antes de o Amazon S3 retornar um êxito para a W1, portanto, o resultado do valor final é desconhecido (color = garnet ou color = brick). Todas as leituras subsequentes (leitura consistente ou eventualmente consistente) podem retornar qualquer um dos valores. Além disso, dependendo do tempo decorrido, uma leitura eventualmente consistente poderá retornar nenhum resultado.



## Recursos do Amazon S3

### Tópicos

- [Classes de armazenamento \(p. 6\)](#)
- [Políticas de buckets \(p. 6\)](#)
- [AWS Identity and Access Management \(p. 7\)](#)
- [Listas de controle de acesso \(p. 7\)](#)
- [Versionamento \(p. 8\)](#)
- [Operações \(p. 8\)](#)

Esta seção descreve recursos importantes do Amazon S3.

## Classes de armazenamento

O Amazon S3 oferece diversas classes de armazenamento criadas para diferentes casos de uso. Elas incluem o Amazon S3 STANDARD para armazenamento de finalidades gerais para dados acessados frequentemente, Amazon S3 STANDARD\_IA para dados de longa duração, mas acessados com menos frequência, e GLACIER para armazenamentos a longo prazo.

Para obter mais informações, consulte [Classes de armazenamento \(p. 107\)](#).

## Políticas de buckets

As políticas de bucket fornecem controle de acesso centralizado aos buckets e objetos com base em várias condições, incluindo operações do Amazon S3, solicitantes, recursos e aspectos da solicitação (por exemplo, endereço IP). As políticas são expressas em nossa linguagem de políticas de acesso e permitem gerenciamento centralizado de permissões. Permissões anexadas a um bucket aplicam-se a todos os objetos nesse bucket.

Usuários individuais e empresas podem usar políticas de bucket. Quando as empresas se registram no Amazon S3, elas criam uma conta. Depois disso, a empresa se torna um sinônimo da conta. As contas

são financeiramente responsáveis pelos recursos da Amazon que elas (e seus funcionários) criam. As contas têm a capacidade de conceder permissões às políticas de buckets e de atribuir permissões aos funcionários com base em uma variedade de condições. Por exemplo, uma conta pode criar uma política que dá a um usuário acesso de gravação:

- Para um bucket específico do S3
- Na rede corporativa de uma conta
- Durante o horário comercial

Uma conta pode conceder a um usuário acesso limitado de leitura e gravação, mas permitir que outro crie e exclua buckets também. Uma conta pode permitir que vários escritórios armazenem relatórios diários em um único bucket, permitindo que cada escritório grave apenas em um determinado conjunto de nomes (por exemplo, "Nevada/\*" ou "Utah/\*") e somente no intervalo de endereços IP do escritório.

Ao contrário das listas de controle de acesso (descritas a seguir), que podem adicionar (conceder) permissões somente em objetos individuais, as políticas podem adicionar ou negar permissões em todos os objetos (ou em um subconjunto) em um bucket. Com uma solicitação uma conta pode definir as permissões de qualquer número de objetos em um bucket. Uma conta pode usar os caracteres curinga (semelhantes a operadores de expressão regular) em nomes de recursos da Amazon (ARNs) e outros valores, para que uma conta possa controlar o acesso a grupos de objetos que começam com um prefixo ou um final comum com uma determinada extensão como html.

Só o proprietário do bucket tem permissão para associar uma política a um bucket. As políticas, escritas na linguagem de políticas de acesso, permitem ou negam solicitações com base em:

- Operações em buckets do Amazon S3 (como `PUT ?acl`) e operações em objetos (como `PUT Object` ou `GET Object`)
- Solicitante
- Condições especificadas na política

Uma conta pode controlar o acesso com base em operações específicas do Amazon S3, como `GetObject`, `GetObjectVersion`, `DeleteObject` ou `DeleteBucket`.

As condições podem ser coisas como endereços IP, intervalos de endereços IP em notação CIDR, datas, agentes de usuário, referenciador HTTP e transportes (HTTP e HTTPS).

Para obter mais informações, consulte [Uso de políticas de bucket e políticas de usuário \(p. 326\)](#).

## AWS Identity and Access Management

Por exemplo, você pode usar IAM com Amazon S3 para controlar o tipo de acesso de um usuário ou grupo de usuários a partes específicas do bucket Amazon S3 que sua conta da AWS possui.

Para obter mais informações sobre IAM, consulte o seguinte:

- [AWS Identity and Access Management \(IAM\)](#)
- [Conceitos básicos](#)
- [Guia do usuário do IAM](#)

## Listas de controle de acesso

Para obter mais informações, consulte [Gerenciar o acesso com ACLs \(p. 390\)](#)

## Versionamento

Para obter mais informações, consulte [Versionamento de objeto \(p. 111\)](#).

## Operações

As seguintes são as operações mais comuns que você executará por meio da API.

### Operações comuns

- Criar um bucket – Crie e nomeie seu próprio bucket para armazenar seus objetos.
- Gravar um objeto – armazene dados criando ou substituindo um objeto. Quando você grava um objeto, você especifica uma chave exclusiva no namespace de seu bucket. Essa também é uma boa hora para especificar qualquer controle de acesso desejado no objeto.
- Ler um objeto – leia os dados de volta. Você pode baixar os dados via HTTP ou BitTorrent.
- Excluir um objeto – exclua alguns de seus dados.
- Listar chaves – liste as chaves contidas em um de seus buckets. Você pode filtrar a lista de chaves com base em um prefixo.

Detalhes sobre isso e todas as outras funcionalidades são descritos em detalhes posteriormente neste guia.

## Interfaces de programação de aplicativos (APIs) do Amazon S3

A arquitetura do Amazon S3 foi desenvolvida para ser neutra em termos de linguagem de programação que usa nossas interfaces compatíveis para armazenar e recuperar objetos.

O Amazon S3 fornece uma interface REST e uma interface SOAP. Elas são semelhantes, mas há algumas diferenças. Por exemplo, na interface REST, os metadados são retornados em cabeçalhos HTTP. Como só oferecemos suporte a solicitações HTTP de até 4 KB (sem incluir o corpo), a quantidade de metadados que você pode fornecer é restrita.

### Note

O suporte de SOAP via HTTP está obsoleto, mas continua disponível via HTTPS. Novos recursos do Amazon S3 não serão compatíveis com o SOAP. Recomendamos que você use o REST API ou os SDKs da AWS.

## A interface REST

A API REST é uma interface HTTP para o Amazon S3. Usando REST, você usa solicitações HTTP padrão para criar, buscar e excluir bucket e objetos.

Você pode usar qualquer toolkit compatível com HTTP para usar a API REST. Você pode até usar um navegador para buscar objetos, desde que eles possam ser lidos anonimamente.

A API REST usa os cabeçalhos padrão e os códigos de status HTTP, para que os navegadores e os toolkits padrão funcionem como esperado. Em algumas áreas, adicionamos funcionalidade ao HTTP (por exemplo, adicionamos cabeçalhos para oferecer suporte ao controle de acesso). Nesses casos, fizemos o melhor para adicionar nova funcionalidade de uma forma que corresponesse ao estilo de uso padrão do HTTP.

## A interface SOAP

### Note

O suporte de SOAP via HTTP está obsoleto, mas continua disponível via HTTPS. Novos recursos do Amazon S3 não serão compatíveis com o SOAP. Recomendamos que você use o REST API ou os SDKs da AWS.

A API SOAP fornece uma interface SOAP 1.1 usando a codificação literal de documentos. A maneira mais comum de usar o SOAP é baixar o WSDL (consulte <http://doc.s3.amazonaws.com/2006-03-01/AmazonS3.wsdl>), usar um toolkit do SOAP, como o Apache Axis ou o Microsoft .NET, para criar associações e, em seguida, escrever código que use as associações para chamar o Amazon S3.

## Pagar pelo Amazon S3

A definição de preço para o Amazon S3 foi desenvolvida para que você não precise planejar para os requisitos de armazenamento de seu aplicativo. A maioria dos provedores de armazenamento força você a comprar uma quantidade predeterminada de armazenamento e capacidade de transferência de rede: Se você exceder essa capacidade, o serviço é desligado e você é cobrado por altas taxas excedentes. Se você não exceder essa capacidade, você paga como se tivesse usado tudo.

O Amazon S3 cobra apenas pelo que você realmente usa, sem taxas ocultas e nenhuma taxa excedente. Isso fornece aos desenvolvedores um serviço de custo variável que pode aumentar com seus negócios e ao mesmo tempo usufruir das vantagens de custos de infraestrutura da Amazon.

Antes de armazenar qualquer coisa no Amazon S3, você precisa se registrar com o serviço e fornecer um modo de pagamento que será cobrado no final de cada mês. Não há encargos de configuração para começar a usar o serviço. No final do mês, seu modo de pagamento é automaticamente cobrado pelo uso daquele mês.

Para obter informações sobre como pagar pelo armazenamento do Amazon S3, consulte [Definição de preço do Amazon S3](#).

## Serviços relacionados

Após carregar os dados no Amazon S3, você pode usá-los com outros serviços que fornecemos. Os seguintes serviços são os que podem ser usados com mais frequência:

- Amazon Elastic Compute Cloud – este serviço da web fornece recursos virtuais de computação na nuvem. Para obter mais informações, visite a [página de detalhes do produto Amazon EC2](#).
- Amazon EMR – este serviço da Web permite que empresas, pesquisadores, analistas de dados e desenvolvedores processem de maneira fácil e econômica grandes quantidades de dados. Ele usa uma estrutura Hadoop hospedada que é executada na infraestrutura de escala da web do Amazon EC2 e do Amazon S3. Para obter mais informações, visite a [página de detalhes do produto Amazon EMR](#).
- O AWS Import/Export – AWS Import/Export permite que você envie um dispositivo de armazenamento por e-mail, como um disco RAID, para a Amazon para que possamos carregar seus (terabytes) de dados no Amazon S3. Para obter mais informações, acesse o [AWS Import/Export Developer Guide](#).

# Fazer solicitações

## Tópicos

- [Sobre as chaves de acesso \(p. 10\)](#)
- [Endpoints de solicitações \(p. 11\)](#)
- [Fazer solicitações ao Amazon S3 por meio do IPv6 \(p. 12\)](#)
- [Fazer solicitações usando os SDKs da AWS \(p. 18\)](#)
- [Fazer solicitações usando a API REST \(p. 45\)](#)

O Amazon S3 é um serviço REST. Você pode enviar solicitações para o Amazon S3 usando a API REST ou as bibliotecas wrapper do AWS SDK (consulte [Código de exemplo e bibliotecas](#)) que envolvem a API REST estrutural do Amazon S3, simplificando as tarefas de programação.

Toda interação com o Amazon S3 é autenticada ou anônima. A autenticação é um processo de verificação da identidade do solicitante que está tentando acessar um produto da Amazon Web Services (AWS). As solicitações autenticadas devem incluir um valor de assinatura que autentique o remetente da solicitação. O valor de assinatura é, em parte, gerado a partir das chaves de acesso da AWS do solicitante (ID de chave de acesso e chave de acesso secreta). Para obter mais informações sobre a obtenção de chaves de acesso, consulte [Como obter credenciais de segurança?](#) no AWS General Reference.

Se você estiver usando o SDK da AWS, as bibliotecas calcularão a assinatura a partir das chaves fornecidas. No entanto, se fizer chamadas diretas da API REST no aplicativo, você deverá escrever o código para calcular a assinatura e adicioná-la à solicitação.

## Sobre as chaves de acesso

As seções a seguir avaliam os tipos de chaves de acesso que você pode usar para fazer solicitações autenticadas.

## Chaves de acesso da conta da AWS

As chaves de acesso da conta fornecem acesso total aos recursos da AWS que são de propriedade da conta. Veja a seguir exemplos de chaves de acesso:

- ID de chave de acesso (uma string de 20 caracteres alfanuméricos). Por exemplo:  
AKIAIOSFODNN7EXAMPLE
- Chave de acesso secreta (uma string de 40 caracteres). Por exemplo: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

O ID de chave de acesso identifica uma conta da AWS de maneira única. Use essas chaves de acesso para enviar solicitações autenticadas para o Amazon S3.

## Chaves de acesso do usuário do IAM

Você pode criar uma conta da AWS para sua empresa. No entanto, podem existir vários funcionários na organização que precisam de acesso aos recursos da AWS da organização. Compartilhar as chaves de acesso da conta da AWS reduz a segurança e criar contas individuais da AWS para cada funcionário pode não ser prático. Além disso, não é fácil compartilhar recursos como buckets e objetos, pois eles pertencem a contas diferentes. Para compartilhar recursos, você deve conceder permissões, o que gera trabalho adicional.

Em tais cenários, use o AWS Identity and Access Management (IAM) para criar usuários na conta da AWS com suas próprias chaves de acesso e anexe políticas de usuário do IAM concedendo as permissões de acesso aos recursos apropriados. Para gerenciar melhor esses usuários, o IAM permite que você crie grupos de usuários e conceda permissões no nível de grupo que se aplicam a todos os usuários do grupo.

Esses usuários são conhecidos como usuários do IAM criados e gerenciados dentro da AWS. A conta pai controla a capacidade que um usuário tem de acessar a AWS. Quaisquer recursos que um usuário do IAM cria estão sob o controle da conta pai da AWS e são pagos para ela. Esses usuários do IAM podem enviar solicitações autenticadas para o Amazon S3 usando as próprias credenciais de segurança. Para obter mais informações sobre a criação e o gerenciamento de usuário na conta da AWS, acesse a [página de detalhes do produto do AWS Identity and Access Management](#).

## Credenciais de segurança temporárias

Além de criar usuários do IAM com suas próprias chaves de acesso, o IAM também permite que você conceda credenciais de segurança temporárias (chaves de acesso temporárias e um token de segurança) a qualquer usuário do IAM permitindo que eles acessem serviços e recursos da AWS. Você também pode gerenciar usuários no sistema fora da AWS. Eles são conhecidos como usuários federados. Além disso, usuários podem ser aplicativos criados para acessar os recursos da AWS.

O IAM fornece a API do AWS Security Token Service para a solicitação de credenciais de segurança temporárias. Use a API da AWS STS ou o SDK da AWS para solicitar essas credenciais. A API retorna as credenciais de segurança temporárias (ID de chave de acesso e chave de acesso secreta) e um token de segurança. Essas credenciais são válidas apenas pela duração especificada ao solicitá-las. Use o ID de chave de acesso e a chave secreta da mesma forma que os usa ao enviar solicitações usando a conta da AWS ou as chaves de acesso do usuário do IAM. Além disso, é necessário incluir o token em cada solicitação enviada para o Amazon S3.

Um usuário do IAM pode solicitar essas credenciais de segurança temporárias para seu próprio uso ou enviá-las para usuários federados ou aplicativos. Ao solicitar credenciais de segurança temporárias para usuários federados, você deve fornecer um nome de usuário e uma política do IAM definindo as permissões que deseja associar a essas credenciais. O usuário federado não pode obter mais permissões que o usuário pai do IAM que solicitou as credenciais temporárias.

Use as credenciais de segurança temporárias para fazer solicitações ao Amazon S3. As bibliotecas de API calculam o valor de assinatura necessário usando essas credenciais para autenticar sua solicitação. Se você enviar solicitações usando credenciais vencidas, o Amazon S3 negará a solicitação.

Para obter informações sobre a assinatura de solicitações usando credenciais de segurança temporárias nas solicitações da API REST, consulte [Assinar e autenticar as solicitações REST \(p. 659\)](#). Para obter informações sobre o envio de solicitações usando SDKs da AWS, consulte [Fazer solicitações usando os SDKs da AWS \(p. 18\)](#).

Para obter mais informações sobre suporte do IAM para credenciais de segurança temporárias, consulte [Credenciais de segurança temporárias](#) no Guia do usuário do IAM.

Para maior segurança, é possível exigir autenticação multifator (MFA) ao acessar os recursos do Amazon S3 configurando uma política do bucket. Para obter mais informações, consulte [Adição de uma política de bucket para exigir MFA \(p. 362\)](#). Depois de exigir a MFA para acesso aos recursos do Amazon S3, a única maneira de acessar esses recursos é fornecendo credenciais temporárias criadas com uma chave MFA. Para obter mais informações, consulte a página de detalhes [AWS Multi-Factor Authentication e Configurar o acesso à API com proteção MFA](#) no Guia do usuário do IAM.

## Endpoints de solicitações

Envie solicitações REST para o endpoint predefinido do serviço. Para obter uma lista de todos os serviços da AWS e os seus respectivos endpoints, acesse [Regiões e endpoints](#) na AWS General Reference.

# Fazer solicitações ao Amazon S3 por meio do IPv6

O Amazon Simple Storage Service (Amazon S3) oferece suporte à capacidade de acessar buckets do S3 usando o Protocolo de Internet versão 6 (IPv6), além do protocolo IPv4. Os endpoints de pilha dupla do Amazon S3 oferecem suporte a buckets do S3 por meio do IPv6 e do IPv4. Não há custo adicional para acessar o Amazon S3 por meio do IPv6. Para obter mais informações sobre definição de preço, consulte [Definição de preço do Amazon S3](#).

## Tópicos

- [Conceitos básicos para fazer solicitações por meio do IPv6 \(p. 12\)](#)
- [Usar endereços do IPv6 em políticas do IAM \(p. 13\)](#)
- [Testar a compatibilidade com endereços IP \(p. 14\)](#)
- [Usar endpoints de pilha dupla do Amazon S3 \(p. 14\)](#)

## Conceitos básicos para fazer solicitações por meio do IPv6

Para fazer uma solicitação para um bucket do S3 por meio do IPv6, você precisa usar um endpoint de pilha dupla. A próxima seção descreve como fazer solicitações por meio do IPv6 usando endpoints de pilha dupla.

Estas são algumas coisas sobre as quais você deve estar ciente antes de tentar acessar um bucket por meio do IPv6:

- O cliente e a rede que estão acessando o bucket devem ter permissão para usar o IPv6.
- As solicitações de estilo hospedado virtual e de estilo de caminho são compatíveis para acessarem o IPv6. Para obter mais informações, consulte [Endpoints de pilha dupla do Amazon S3 \(p. 15\)](#).
- Se você usar a filtragem de endereços IP de origem nas políticas de usuário ou de bucket do AWS Identity and Access Management (IAM), será necessário atualizar as políticas para incluir intervalos de endereços IPv6. Para obter mais informações, consulte [Usar endereços do IPv6 em políticas do IAM \(p. 13\)](#).
- Ao usar o IPv6, os arquivos de log de acesso ao servidor fornecem endereços IP em um formato do IPv6. Você precisa atualizar as ferramentas, os scripts e o software existentes que usa para analisar os arquivos de log do Amazon S3 para que eles possam analisar os endereços Remote IP formatados para IPv6. Para obter mais informações, consulte [Formato do log de acesso ao servidor \(p. 631\)](#) e [Registro em log de acesso ao servidor Amazon S3 \(p. 625\)](#).

### Note

Se você tiver problemas relacionados à presença de endereços IPv6 nos arquivos de log, entre em contato com o [AWS Support](#).

## Fazer solicitações por meio do IPv6 usando endpoints de pilha dupla

Você faz solicitações com chamadas da API do Amazon S3 por meio do IPv6 usando endpoints de pilha dupla. As operações da API do Amazon S3 funcionam da mesma forma, quer você esteja acessando o Amazon S3 por meio do IPv6 ou do IPv4. O desempenho deve ser o mesmo também.

Ao usar a API REST, você acessa um endpoint de pilha dupla diretamente. Para obter mais informações, consulte [Endpoints de pilha dupla do \(p. 15\)](#).

Ao usar a AWS Command Line Interface (AWS CLI) e os SDKs da AWS, você pode usar um parâmetro ou um sinalizador para mudar para um endpoint de pilha dupla. Você também pode especificar o endpoint de pilha dupla diretamente como uma substituição do endpoint do Amazon S3 no arquivo de configuração.

Você pode usar um endpoint de pilha dupla para acessar um bucket por meio do IPv6 de qualquer um dos seguintes:

- A AWS CLI, consulte [Usar endpoints de pilha dupla da AWS CLI \(p. 15\)](#).
- Os AWS SDKs, consulte [Usar endpoints de pilha dupla dos SDKs da AWS \(p. 16\)](#).
- A API REST, consulte [Fazer solicitações para endpoints de pilha dupla usando a API REST \(p. 46\)](#).

## Recursos não disponíveis por meio do IPv6

No momento, os seguintes recursos não são compatíveis ao acessar um bucket do S3 por meio do IPv6:

- Hospedagem de site estático em um bucket do S3
- BitTorrent

## Usar endereços do IPv6 em políticas do IAM

Antes de tentar acessar um bucket usando o IPv6, você deve garantir que todas as políticas de usuário do IAM ou de bucket do S3 usadas para filtragem de endereços IP estejam atualizadas para incluir intervalos de endereços do IPv6. As políticas de filtragem de endereços IP que não estiverem atualizadas para lidar com endereços do IPv6 podem resultar na perda ou no ganho de acesso de clientes ao bucket quando começarem a usar o IPv6. Para obter mais informações sobre como gerenciar permissões de acesso com o IAM, consulte [Gerenciamento de permissões de acesso aos recursos do Amazon S3 \(p. 282\)](#).

As políticas do IAM que filtram endereços IP usam [Operadores de condição de endereço IP](#). A política de bucket a seguir identifica o intervalo 54.240.143.\* de endereços IPv4 permitidos usando operadores de condição de endereço IP. Todos os endereços IP fora deste intervalo terão o acesso ao bucket negado (examplebucket). Como todos os endereços do IPv6 estão fora do intervalo permitido, essa política impede que os endereços do IPv6 possam acessar o examplebucket.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "IPAllow",  
            "Effect": "Allow",  
            "Principal": "*",  
            "Action": "s3:*",  
            "Resource": "arn:aws:s3:::examplebucket/*",  
            "Condition": {  
                "IpAddress": {"aws:SourceIp": "54.240.143.0/24"}  
            }  
        }  
    ]  
}
```

Você pode modificar o elemento `Condition` da política do bucket para permitir os intervalos de endereços do IPv4 (54.240.143.0/24) e do IPv6 (2001:DB8:1234:5678::/64), conforme mostrado no exemplo a seguir. Você pode usar o mesmo tipo de bloqueio de `Condition` mostrado no exemplo para atualizar as políticas de usuário e de bucket do IAM.

```
"Condition": {  
    "IpAddress": {
```

```
    "aws:SourceIp": [
        "54.240.143.0/24",
        "2001:DB8:1234:5678::/64"
    ]
}
```

Antes de usar o IPv6, você deve atualizar todas as políticas de usuário e de bucket do IAM que usam a filtragem de endereços IP para permitir os intervalos de endereços do IPv6. Recomendamos que você atualize as políticas do IAM com os intervalos de endereços do IPv6 de sua organização além dos intervalos de endereços do IPv4 existentes. Para obter um exemplo de uma política de bucket que permite acesso por meio do IPv6 e do IPv4, consulte [Restringir o acesso a endereços IP específicos \(p. 359\)](#).

Você pode analisar suas políticas de usuário do IAM usando o console do IAM no <https://console.aws.amazon.com/iam/>. Para obter mais informações sobre o IAM, consulte o [Guia do usuário do IAM](#). Para obter informações sobre como editar políticas de bucket do S3, consulte [Como adicionar uma política de bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

## Testar a compatibilidade com endereços IP

Se estiver usando o Linux/Unix ou o Mac OS X, você poderá testar se é possível acessar um endpoint de pilha dupla por meio do IPv6 usando o comando `curl` conforme mostrado no exemplo a seguir:

Example

```
curl -v http://s3.dualstack.us-west-2.amazonaws.com/
```

Você recebe de volta informações semelhantes ao exemplo a seguir. Se estiver conectado por meio do IPv6, o endereço IP conectado será um endereço do IPv6.

```
* About to connect() to s3-us-west-2.amazonaws.com port 80 (#0)
*   Trying IPv6 address... connected
* Connected to s3.dualstack.us-west-2.amazonaws.com (IPv6 address) port 80 (#0)
> GET / HTTP/1.1
> User-Agent: curl/7.18.1 (x86_64-unknown-linux-gnu) libcurl/7.18.1 OpenSSL/1.0.1t
zlib/1.2.3
> Host: s3.dualstack.us-west-2.amazonaws.com
```

Se estiver usando o Microsoft Windows 7 ou 10, você poderá testar se é possível acessar um endpoint de pilha dupla por meio do IPv6 ou do IPv4 usando o comando `ping` conforme mostrado no exemplo a seguir.

```
ping ipv6.s3.dualstack.us-west-2.amazonaws.com
```

## Usar endpoints de pilha dupla do Amazon S3

Os endpoints de pilha dupla do Amazon S3 dão suporte a solicitações para buckets do S3 por meio do IPv6 e do IPv4. Esta seção descreve como usar os endpoints de pilha dupla.

### Tópicos

- [Endpoints de pilha dupla do Amazon S3 \(p. 15\)](#)
- [Usar endpoints de pilha dupla da AWS CLI \(p. 15\)](#)
- [Usar endpoints de pilha dupla dos SDKs da AWS \(p. 16\)](#)
- [Usar endpoints de pilha dupla da API REST \(p. 17\)](#)

## Endpoints de pilha dupla do Amazon S3

Quando você faz uma solicitação para um endpoint de pilha dupla, o URL do bucket resolve para um endereço IPv6 ou IPv4. Para obter mais informações sobre como acessar um bucket por meio do IPv6, consulte [Fazer solicitações ao Amazon S3 por meio do IPv6 \(p. 12\)](#).

Ao usar a API REST, você acessa diretamente um endpoint do Amazon S3 usando o nome do endpoint (URI). Você pode acessar um bucket do S3 por meio de um endpoint de pilha dupla usando um nome de endpoint de estilo de hospedagem virtual ou de estilo de caminho. O Amazon S3 dá suporte apenas a nomes regionais de endpoint de pilha dupla, o que significa que você deve especificar a região como parte do nome.

Use as seguintes convenções de atribuição de nomes para os nomes de endpoint de estilo de hospedagem virtual e de estilo de caminho de pilha dupla:

- Endpoint de pilha dupla de estilo de hospedagem virtual

`bucketname.s3.dualstack.aws-region.amazonaws.com`

- Endpoint de pilha dupla de estilo de caminho:

`s3.dualstack.aws-region.amazonaws.com/bucketname`

Para obter mais informações sobre estilo de nome de endpoints, consulte [Acesso a um bucket \(p. 56\)](#). Para obter uma lista dos endpoints do Amazon S3, consulte [Regiões e endpoints](#) no AWS General Reference.

### Important

Você pode usar a aceleração de transferência com endpoints de pilha dupla. Para obter mais informações, consulte [Conceitos básicos do Amazon S3 Transfer Acceleration \(p. 76\)](#).

Ao usar a AWS Command Line Interface (AWS CLI) e os SDKs da AWS, você pode usar um parâmetro ou um sinalizador para mudar para um endpoint de pilha dupla. Você também pode especificar o endpoint de pilha dupla diretamente como uma substituição do endpoint do Amazon S3 no arquivo de configuração. As seções a seguir descrevem como usar endpoints de pilha dupla da AWS CLI e dos SDKs da AWS.

## Usar endpoints de pilha dupla da AWS CLI

Esta seção fornece exemplos de comandos da AWS CLI usados para fazer solicitações a um endpoint de pilha dupla. Para obter instruções de configuração da AWS CLI, consulte [Configurar a CLI da AWS \(p. 645\)](#).

Você define o valor de configuração `use_dualstack_endpoint` para `true` em um perfil no seu arquivo do AWS Config para direcionar todas as solicitações do Amazon S3 feitas pelos comandos `s3` e `s3api` da AWS CLI ao endpoint de pilha dupla para a região especificada. Especifique a região no arquivo de configuração ou em um comando usando a opção `--region`.

Quando se usa endpoints de pilha dupla com a AWS CLI, os estilos de endereçamento `path` e `virtual` são compatíveis. O estilo de endereçamento, definido no arquivo de configuração, controla se o nome do bucket está no hostname ou em parte do URL. Por padrão, a CLI tentará usar o estilo `virtual` sempre que possível, mas voltará ao estilo de caminho se necessário. Para obter mais informações, consulte [Configuração da AWS CLI do Amazon S3](#).

Você também pode fazer alterações de configuração usando um comando, conforme mostrado no exemplo a seguir que define `use_dualstack_endpoint` para `true` e `addressing_style` para `virtual` no perfil padrão.

```
$ aws configure set default.s3.use_dualstack_endpoint true
$ aws configure set default.s3.addressing_style virtual
```

Se quiser usar um endpoint de pilha dupla apenas para comandos especificados da AWS CLI, (nem todos os comandos), você pode usar qualquer um dos métodos a seguir:

- Você pode usar o endpoint de pilha dupla por comando, definindo o parâmetro `--endpoint-url` como `https://s3.dualstack.aws-region.amazonaws.com` ou `http://s3.dualstack.aws-region.amazonaws.com` para qualquer comando s3 ou s3api.

```
$ aws s3api list-objects --bucket bucketname --endpoint-url https://s3.dualstack.aws-region.amazonaws.com
```

- Você pode configurar perfis separados em seu arquivo do AWS Config. Por exemplo, crie um perfil que defina `use_dualstack_endpoint` como `true` e um perfil que não defina `use_dualstack_endpoint`. Quando executar um comando, especifique qual perfil deseja usar, dependendo de querer ou não usar o endpoint de pilha dupla.

#### Note

Atualmente, ao usar a AWS CLI, você não pode usar a aceleração de transferência com endpoints de pilha dupla. Contudo, o suporte para a CLI da AWS estará disponível em breve. Para obter mais informações, consulte [Usar o Transfer Acceleration na AWS Command Line Interface \(AWS CLI\) \(p. 79\)](#).

## Usar endpoints de pilha dupla dos SDKs da AWS

Esta seção fornece exemplos de como acessar um endpoint de pilha dupla usando os SDKs da AWS.

### Exemplo do endpoint de pilha dupla do AWS SDK for Java

O exemplo a seguir mostra como habilitar endpoints de pilha dupla ao criar um cliente do Amazon S3 usando o AWS SDK for Java.

Para obter instruções sobre criar e testar um exemplo funcional Java, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;

public class DualStackEndpoints {

    public static void main(String[] args) {
        String clientRegion = "**** Client region ****";
        String bucketName = "**** Bucket name ****";

        try {
            // Create an Amazon S3 client with dual-stack endpoints enabled.
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(clientRegion)
                .withDualstackEnabled(true)
        }
    }
}
```

```
        .build();

        s3Client.listObjects(bucketName);
    }
    catch(AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    }
    catch(SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
```

Se estiver usando o AWS SDK for Java no Windows, você talvez tenha de definir a seguinte propriedade da máquina virtual Java (JVM):

```
java.net.preferIPv6Addresses=true
```

## Exemplo do endpoint de pilha dupla do AWS SDK para .NET

Ao usar o AWS SDK para .NET você, você usa a classe `AmazonS3Config` para permitir o uso de um endpoint de pilha dupla, como mostrado no exemplo a seguir.

```
var config = new AmazonS3Config
{
    UseDualstackEndpoint = true,
    RegionEndpoint = RegionEndpoint.USWest2
};

using (var s3Client = new AmazonS3Client(config))
{
    var request = new ListObjectsRequest
    {
        BucketName = "myBucket"
    };

    var response = await s3Client.ListObjectsAsync(request);
}
```

Para uma amostra completa da .NET para objetos de listagem, consulte [Listagem de chaves usando o AWS SDK para .NET \(p. 234\)](#).

Para obter informações sobre como criar e testar um exemplo funcional .NET, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

## Usar endpoints de pilha dupla da API REST

Para obter informações sobre fazer solicitação para endpoints de pilha dupla usando a API REST, consulte [Fazer solicitações para endpoints de pilha dupla usando a API REST \(p. 46\)](#).

# Fazer solicitações usando os SDKs da AWS

## Tópicos

- [Fazer solicitações usando credenciais de usuário do IAM ou da conta da AWS \(p. 18\)](#)
- [Fazer solicitações usando credenciais temporárias de usuário do IAM \(p. 25\)](#)
- [Fazer solicitações usando as credenciais temporárias de usuário federado \(p. 34\)](#)

Você pode enviar solicitações autenticadas para o Amazon S3 usando os SDKs da AWS ou fazendo as chamadas de API REST diretamente em seu aplicativo. A API do SDK da AWS usa as credenciais que você fornece para computar a assinatura para autenticação. Se você usar a API REST diretamente em seus aplicativos, deverá gravar o código necessário para computar a assinatura para autenticar sua solicitação. Para uma lista de SDKs da AWS disponíveis, acesse [Código de exemplo e bibliotecas](#).

## Fazer solicitações usando credenciais de usuário do IAM ou da conta da AWS

Você pode usar suas credenciais de segurança de usuário do IAM ou da conta da AWS para enviar solicitações autenticadas para o Amazon S3. Esta seção fornece exemplos de como você pode enviar solicitações autenticadas usando o AWS SDK for Java, o AWS SDK para .NET e o AWS SDK para PHP. Para uma lista de SDKs da AWS disponíveis, acesse [Código de exemplo e bibliotecas](#).

## Tópicos

- [Fazer solicitações usando a conta da AWS ou as credenciais de usuário do IAM - AWS SDK for Java \(p. 19\)](#)
- [Fazer solicitações usando a conta da AWS ou as credenciais de usuário do IAM - AWS SDK para .NET \(p. 20\)](#)
- [Fazer solicitações usando a conta da AWS ou as credenciais de usuário do IAM - AWS SDK para PHP \(p. 22\)](#)
- [Fazer solicitações usando a conta da AWS ou as credenciais de usuário do IAM - AWS SDK para Ruby \(p. 23\)](#)

Cada um desses SDKs da AWS usa uma cadeia de provedor de credenciais específicas do SDK para encontrar e usar credenciais, além de realizar ações em nome do proprietário das credenciais. O que todas essas cadeias de provedor de credenciais têm em comum é que elas procuram por seu arquivo local de credenciais da AWS.

A forma mais fácil de configurar credenciais para os SDKs da AWS é usar um arquivo de credenciais da AWS. Caso utilize a AWS Command Line Interface (AWS CLI), você já deve ter um arquivo de credenciais local da AWS configurado. Caso contrário, use o procedimento a seguir para configurar um arquivo de credenciais:

### Para criar um arquivo de credenciais local da AWS

1. Faça login no Console de gerenciamento da AWS e abra o console da IAM em <https://console.aws.amazon.com/iam/>.
2. Crie um novo usuário com permissões limitadas aos serviços e ações aos quais você deseja que seu código tenha acesso. Para obter mais informações sobre como criar um usuário do IAM, consulte [Criação de usuários do IAM \(console\)](#) e siga as instruções até a etapa 8.
3. Escolha Fazer download do arquivo .csv para salvar uma cópia de suas credenciais da AWS.
4. Em seu computador, navegue para seu diretório inicial e crie um diretório .aws. Nos sistemas baseados em Unix, como Linux ou OS X, isso fica no seguinte local:

```
~/ .aws
```

No Windows, isso está no seguinte local:

```
%HOMEPATH%\ .aws
```

5. No diretório `.aws`, crie um novo arquivo chamado `credentials`.
6. Abra o arquivo de credenciais `.csv` que você baixou do console do IAM e copie o conteúdo dele para o arquivo `credentials` usando o seguinte formato:

```
[default]
aws_access_key_id = your_access_key_id
aws_secret_access_key = your_secret_access_key
```

7. Salve o arquivo `credentials` e exclua o arquivo `.csv` que você baixou na etapa 3.

Seu arquivo de credenciais compartilhado agora está configurado em seu computador local, e ele está pronto para ser usado com os SDKs da AWS.

## Fazer solicitações usando a conta da AWS ou as credenciais de usuário do IAM - AWS SDK for Java

Para enviar solicitações autenticadas para o Amazon S3 usando as credenciais de conta da AWS ou de usuário do IAM, faça o seguinte:

- Use a classe `AmazonS3ClientBuilder` class para criar uma instância `AmazonS3Client`.
- Executar um dos métodos do `AmazonS3Client` para enviar solicitações para o Amazon S3. O cliente gera a assinatura necessária a partir das credenciais que você fornece e a inclui na solicitação.

O exemplo a seguir realiza as tarefas anteriores. Para obter informações sobre como criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

### Example

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-
developer-guide/blob/master/LICENSE-SAMPLECODE.)

import java.io.IOException;
import java.util.List;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ListObjectsRequest;
import com.amazonaws.services.s3.model.ObjectListing;
import com.amazonaws.services.s3.model.S3ObjectSummary;

public class MakingRequests {

    public static void main(String[] args) throws IOException {
        String clientRegion = "**** Client region ****";
        String bucketName = "**** Bucket name ****";
    }
}
```

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

    // Get a list of objects in the bucket, two at a time, and
    // print the name and size of each object.
    ListObjectsRequest listRequest = new
ListObjectsRequest().withBucketName(bucketName).withMaxKeys(2);
    ObjectListing objects = s3Client.listObjects(listRequest);
    while(true) {
        List<S3ObjectSummary> summaries = objects.getObjectSummaries();
        for(S3ObjectSummary summary : summaries) {
            System.out.printf("Object \"%s\" retrieved with size %d\n",
summary.getKey(), summary.getSize());
        }
        if(objects.isTruncated()) {
            objects = s3Client.listNextBatchOfObjects(objects);
        }
        else {
            break;
        }
    }
}
catch(AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
}
catch(SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

## Recursos relacionados

- [Usar os AWS SDKs, a CLI e os Explorers \(p. 639\)](#)

## Fazer solicitações usando a conta da AWS ou as credenciais de usuário do IAM - AWS SDK para .NET

Para enviar solicitações autenticadas usando as credenciais de conta da AWS ou de usuário IAM:

- Crie uma instância da classe `AmazonS3Client`.
- Executar um dos métodos do `AmazonS3Client` para enviar solicitações para o Amazon S3. O cliente gera a assinatura necessária a partir das credenciais que você fornece e a inclui na solicitação enviada para o Amazon S3.

O exemplo de C# a seguir mostra como realizar as tarefas anteriores. Para obter informações sobre como executar exemplos .NET neste guia e para instruções sobre como armazenar suas credenciais em um arquivo de configuração, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

### Example

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
```

```
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)  
  
using Amazon.S3;  
using Amazon.S3.Model;  
using System;  
using System.Threading.Tasks;  
  
namespace Amazon.DocSamples.S3  
{  
    class MakeS3RequestTest  
    {  
        private const string bucketName = "*** bucket name ***";  
        // Specify your bucket region (an example region is shown).  
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;  
        private static IAmazonS3 client;  
  
        public static void Main()  
        {  
            using (client = new AmazonS3Client(bucketRegion))  
            {  
                Console.WriteLine("Listing objects stored in a bucket");  
                ListingObjectsAsync().Wait();  
            }  
        }  
  
        static async Task ListingObjectsAsync()  
        {  
            try  
            {  
                ListObjectsRequest request = new ListObjectsRequest  
                {  
                    BucketName = bucketName,  
                    MaxKeys = 2  
                };  
                do  
                {  
                    ListObjectsResponse response = await client.ListObjectsAsync(request);  
                    // Process the response.  
                    foreach (S3Object entry in response.S3Objects)  
                    {  
                        Console.WriteLine("key = {0} size = {1}",  
                            entry.Key, entry.Size);  
                    }  
  
                    // If the response is truncated, set the marker to get the next  
                    // set of keys.  
                    if (response.IsTruncated)  
                    {  
                        request.Marker = response.NextMarker;  
                    }  
                    else  
                    {  
                        request = null;  
                    }  
                } while (request != null);  
            }  
            catch (AmazonS3Exception e)  
            {  
                Console.WriteLine("Error encountered on server. Message:'{0}' when writing  
an object", e.Message);  
            }  
            catch (Exception e)  
            {  
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when  
writing an object", e.Message);  
            }  
        }  
    }  
}
```

```
        }
    }
}
```

#### Note

Você pode criar o cliente `AmazonS3Client` sem fornecer suas credenciais de segurança. As solicitações que são enviadas usando esse cliente são anônimas, sem assinatura. O Amazon S3 retornará um erro se você enviar solicitações anônimas para um recurso não disponível publicamente.

Para obter exemplos funcionais, consulte [Trabalho com objetos do Amazon S3 \(p. 101\)](#) e [Trabalho com buckets do Amazon S3 \(p. 54\)](#). Você pode testar esses exemplos usando sua conta da AWS ou as credenciais de um usuário do IAM.

Por exemplo, para listar todas as chaves de objetos em seu bucket, consulte [Listagem de chaves usando o AWS SDK para .NET \(p. 234\)](#).

#### Recursos relacionados

- [Usar os AWS SDKs, a CLI e os Explorers \(p. 639\)](#)

## Fazer solicitações usando a conta da AWS ou as credenciais de usuário do IAM - AWS SDK para PHP

Esta seção explica como usar uma classe da versão 3 do AWS SDK para PHP para enviar solicitações autenticadas usando suas credenciais da conta da AWS ou de usuário do IAM. Pressupõe-se que você já esteja seguindo as instruções para [Usar o AWS SDK para PHP e executar exemplos do PHP \(p. 649\)](#) e tenha o AWS SDK para PHP devidamente instalado.

O exemplo de PHP a seguir mostra como o cliente faz uma solicitação usando suas credenciais de segurança para listar todos os buckets para a sua conta.

#### Example

```
<?php

require 'vendor/autoload.php';

use Aws\Sts\StsClient;
use Aws\S3\S3Client;
use Aws\S3\Exception\S3Exception;

$bucket = '*** Your Bucket Name ***';

$s3 = new S3Client([
    'region' => 'us-east-1',
    'version' => 'latest',
]);

// Retrieve the list of buckets.
$result = $s3->listBuckets();

try {
    // Retrieve a paginator for listing objects.
    $objects = $s3->getPaginator('ListObjects', [
        'Bucket' => $bucket
    ]);
}
```

```
]);
echo "Keys retrieved!" . PHP_EOL;

// Print the list of objects to the page.
foreach ($objects as $object) {
    echo $object['Key'] . PHP_EOL;
}
} catch (S3Exception $e) {
    echo $e->getMessage() . PHP_EOL;
}
```

#### Note

Você pode criar o cliente `S3Client` sem fornecer suas credenciais de segurança. As solicitações enviadas usando esse cliente são solicitações anônimas, sem uma assinatura. O Amazon S3 retorna um erro se você enviar solicitações anônimas para um recurso que não esteja disponível publicamente.

Para ver um exemplo funcional, consulte [Operações em objetos \(p. 165\)](#). Você pode testar esses exemplos usando suas credenciais de usuário do IAM ou da conta da AWS.

Para um exemplo de listagem de chaves de objeto em um bucket, consulte [Listagem de chaves usando o AWS SDK para PHP \(p. 235\)](#).

#### Recursos relacionados

- [AWS SDK para PHP para classe Aws\S3\S3Client do Amazon S3](#)
- [Documentação do AWS SDK para PHP](#)

## Fazer solicitações usando a conta da AWS ou as credenciais de usuário do IAM - AWS SDK para Ruby

Antes de poder usar a versão 3 do AWS SDK para Ruby para fazer chamadas para o Amazon S3, você deve definir as credenciais de acesso da AWS que o SDK usa para verificar seu acesso aos seus buckets e objetos. Se você compartilhou credenciais configuradas no perfil de credenciais da AWS em seu sistema local, a versão 3 do SDK para Ruby poderá usar essas credenciais sem você ter que declará-las em seu código. Para obter mais informações sobre como configurar credenciais compartilhadas, consulte [Fazer solicitações usando credenciais de usuário do IAM ou da conta da AWS \(p. 18\)](#).

O trecho de código seguinte do Ruby usa as credenciais de um arquivo de credenciais da AWS compartilhado em um computador local para autenticar uma solicitação a fim de obter todos os nomes de chaves de objeto em um bucket específico. Ela faz o seguinte:

1. Cria uma instância da classe `Aws::S3::Resource`.
2. Faz uma solicitação para o Amazon S3 enumerando objetos em um bucket usando o método `bucket` do `Aws::S3::Resource`. O cliente gera o valor de assinatura necessário com base nas credenciais do arquivo de credenciais da AWS em seu computador e o inclui na solicitação que envia ao Amazon S3.
3. Imprime o array de nomes de chaves de objeto no terminal.

#### Example

```
# Use the Amazon S3 modularized gem for version 3 of the AWS Ruby SDK.
require 'aws-sdk-s3'

# Get an Amazon S3 resource.
```

```
s3 = Aws::S3::Resource.new(region: 'us-west-2')

# Create an array of up to the first 100 object keynames in the bucket.
bucket = s3.bucket('example_bucket').objects.collect(&:key)

# Print the array to the terminal.
puts bucket
```

Se você não tiver um arquivo de credenciais da AWS local, ainda poderá criar o recurso `Aws::S3::Resource` e executar o código nos buckets e objetos do Amazon S3. As solicitações que são enviadas usando a versão 3 do SDK para Ruby são anônimas, sem assinatura por padrão. O Amazon S3 retornará um erro se você enviar solicitações anônimas para um recurso não disponível publicamente.

Você pode usar e expandir o trecho de código anterior para aplicativos do SDK para Ruby, como no seguinte exemplo mais robusto. As credenciais que são usadas para este exemplo vêm de um arquivo de credenciais local da AWS no computador que está executando o aplicativo. As credenciais são para um usuário do IAM que pode listar objetos no bucket que o usuário especifica quando executa o aplicativo.

```
# auth_request_test.rb
# Use the Amazon S3 modularized gem for version 3 of the AWS Ruby SDK.
require 'aws-sdk-s3'

# Usage: ruby auth_request_test.rb list BUCKET

# Set the name of the bucket on which the operations are performed.
# This argument is required
bucket_name = nil

# The operation to perform on the bucket.
operation = 'list' # default
operation = ARGV[0] if (ARGV.length > 0)

if ARGV.length > 1
    bucket_name = ARGV[1]
else
    exit 1
end

# Get an Amazon S3 resource.
s3 = Aws::S3::Resource.new(region: 'us-west-2')

# Get the bucket by name.
bucket = s3.bucket(bucket_name)

case operation

when 'list'
    if bucket.exists?
        # Enumerate the bucket contents and object etags.
        puts "Contents of '%s':\n" % bucket_name
        puts '  Name => GUID\n'

        bucket.objects.limit(50).each do |obj|
            puts "    #{obj.key} => #{obj.etag}"
        end
    else
        puts "The bucket '%s' does not exist!" % bucket_name
    end

else
    puts "Unknown operation: '%s'! Only list is supported." % operation
end
```

# Fazer solicitações usando credenciais temporárias de usuário do IAM

## Tópicos

- [Fazer solicitações usando as credenciais temporárias do usuário do IAM - AWS SDK for Java \(p. 25\)](#)
- [Fazer solicitações usando as credenciais temporárias do usuário do IAM - AWS SDK para .NET \(p. 27\)](#)
- [Fazer solicitações usando credenciais temporárias de usuário do IAM ou da conta da AWS - AWS SDK para PHP \(p. 29\)](#)
- [Fazer solicitações usando as credenciais temporárias do usuário do IAM - AWS SDK para Ruby \(p. 31\)](#)

Uma conta da AWS ou um usuário do IAM pode solicitar credenciais de segurança temporárias e usá-las para enviar solicitações autenticadas para o Amazon S3. Esta seção fornece exemplos de como usar AWS SDK for Java, .NET e PHP para obter credenciais de segurança temporárias e usá-las para autenticar suas solicitações para o Amazon S3.

## Fazer solicitações usando as credenciais temporárias do usuário do IAM - AWS SDK for Java

Um usuário do IAM ou uma conta da AWS pode solicitar credenciais de segurança temporárias (consulte [Fazer solicitações \(p. 10\)](#)) usando o AWS SDK for Java e usá-las para acessar o Amazon S3. Essas credenciais expiram após a duração da sessão especificada. Para usar credenciais de segurança temporárias do IAM, faça o seguinte:

1. Crie uma instância da classe `AWSSecurityTokenServiceClient`. Para obter informações sobre como fornecer credenciais, consulte [Usar os AWS SDKs, a CLI e os Explorers \(p. 639\)](#).
2. Assuma a função desejada chamando o método `assumeRole()` do cliente Security Token Service (STS).
3. Inicie uma sessão chamando o método `getSessionToken()` do cliente STS. Forneça informações da sessão para esse método usando um objeto `GetSessionTokenRequest`.

O método retorna as credenciais de segurança temporárias.

4. Empacote as credenciais de segurança temporárias em um objeto `BasicSessionCredentials`. Você usa esse objeto para fornecer as credenciais de segurança temporárias para o cliente do Amazon S3.
5. Crie uma instância da classe `AmazonS3Client` usando as credenciais de segurança temporárias. Você envia solicitações ao Amazon S3 usando esse cliente. Se você enviar solicitações usando credenciais expiradas, o Amazon S3 retornará um erro.

### Note

Se você obtiver as credenciais de segurança temporárias usando suas credenciais de segurança de conta da AWS, as credenciais temporárias serão válidas somente por uma hora. Você poderá especificar a duração de sessão somente se usar credenciais de usuário do IAM para solicitar uma sessão.

O exemplo a seguir lista um conjunto de chaves de objeto no bucket especificado. O exemplo obtém credenciais de segurança temporárias para uma sessão de duas horas e usa essas credenciais para enviar uma solicitação autenticada para o Amazon S3.

Se você desejar testar o exemplo usando credenciais de usuário do IAM, precisará criar um usuário do IAM em sua conta da AWS. Para obter mais informações sobre como criar um usuário do IAM, consulte [Criar o primeiro usuário do IAM e o grupo de administradores](#) no Guia do usuário do IAM.

Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-
developer-guide/blob/master/LICENSE-SAMPLECODE.)

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.BasicSessionCredentials;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ObjectListing;
import com.amazonaws.services.securitytoken.AWSecurityTokenService;
import com.amazonaws.services.securitytoken.AWSecurityTokenServiceClientBuilder;
import com.amazonaws.services.securitytoken.model.AssumeRoleRequest;
import com.amazonaws.services.securitytoken.model.Credentials;
import com.amazonaws.services.securitytoken.model.GetSessionTokenRequest;
import com.amazonaws.services.securitytoken.model.GetSessionTokenResult;

public class MakingRequestsWithIAMTempCredentials {
    public static void main(String[] args) {
        String clientRegion = "**** Client region ****";
        String roleARN = "**** ARN for role to be assumed ****";
        String roleSessionName = "**** Role session name ****";
        String bucketName = "**** Bucket name ****";

        try {
            // Creating the STS client is part of your trusted code. It has
            // the security credentials you use to obtain temporary security credentials.
            AWSecurityTokenService stsClient =
                AWSecurityTokenServiceClientBuilder.standard()
                    .withCredentials(new
ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            // Assume the IAM role. Note that you cannot assume the role of an AWS root
account;
            // Amazon S3 will deny access. You must use credentials for an IAM user or an
IAM role.
            AssumeRoleRequest roleRequest = new AssumeRoleRequest()
                .withRoleArn(roleARN)
                .withRoleSessionName(roleSessionName);
            stsClient.assumeRole(roleRequest);

            // Start a session.
            GetSessionTokenRequest getSessionTokenRequest = new GetSessionTokenRequest();
            // The duration can be set to more than 3600 seconds only if temporary
            // credentials are requested by an IAM user rather than an account owner.
            getSessionTokenRequest.setDurationSeconds(7200);
            GetSessionTokenResult sessionTokenResult =
                stsClient.getSessionToken(getSessionTokenRequest);
            Credentials sessionCredentials = sessionTokenResult.getCredentials();

            // Package the temporary security credentials as a BasicSessionCredentials
object
            // for an Amazon S3 client object to use.
            BasicSessionCredentials basicSessionCredentials = new BasicSessionCredentials(

```

```
        sessionCredentials.getAccessKeyId(),
        sessionCredentials.getSecretAccessKey(),
        sessionCredentials.getSessionToken());

        // Provide temporary security credentials so that the Amazon S3 client
        // can send authenticated requests to Amazon S3. You create the client
        // using the basicSessionCredentials object.
        AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
            .withCredentials(new
        AWSStaticCredentialsProvider(basicSessionCredentials))
            .withRegion(clientRegion)
            .build();

        // Verify that assuming the role worked and the permissions are set correctly
        // by getting a set of object keys from the bucket.
        ObjectListing objects = s3Client.listObjects(bucketName);
        System.out.println("No. of Objects: " + objects.getObjectSummaries().size());
    }
    catch(AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    }
    catch(SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}
```

## Recursos relacionados

- [Usar os AWS SDKs, a CLI e os Explorers \(p. 639\)](#)

## Fazer solicitações usando as credenciais temporárias do usuário do IAM - AWS SDK para .NET

Um usuário do IAM ou uma conta da AWS pode solicitar credenciais de segurança temporárias usando o AWS SDK para .NET e usá-las para acessar o Amazon S3. Essas credenciais expiram após a duração da sessão. Para obter credenciais de segurança temporárias e acessar o Amazon S3, faça o seguinte:

1. Crie uma instância do cliente do AWS Security Token Service, `AmazonSecurityTokenServiceClient`. Para obter informações sobre a concessão de credenciais, consulte [Usar os AWS SDKs, a CLI e os Explorers \(p. 639\)](#).
2. Inicie uma sessão chamando o método `GetSessionToken` do cliente STS criado na etapa anterior. Forneça informações da sessão para esse método usando um objeto `GetSessionTokenRequest`.  
O método retorna as credenciais de segurança temporárias.
3. Empacote as credenciais de segurança temporárias em uma instância do objeto `SessionAWSCredentials`. Você usa esse objeto para fornecer as credenciais de segurança temporárias para o cliente do Amazon S3.
4. Crie uma instância da classe `AmazonS3Client` passando as credenciais de segurança temporárias. Você envia solicitações ao Amazon S3 usando esse cliente. Se você enviar solicitações usando credenciais expiradas, o Amazon S3 retornará um erro.

### Note

Se você obtiver as credenciais de segurança temporárias usando suas credenciais de segurança de conta da AWS, as credenciais serão válidas somente por uma hora. Você poderá especificar a duração da sessão somente se usar as credenciais de usuário do IAM para solicitar uma sessão.

O exemplo do C# a seguir lista chaves de objeto no bucket especificado. Como ilustração, o exemplo obtém credenciais de segurança temporárias para uma sessão padrão de uma hora e usa essas credenciais para enviar uma solicitação autenticada para o Amazon S3.

Se você desejar testar o exemplo usando credenciais de usuário do IAM, precisará criar um usuário do IAM em sua conta da AWS. Para obter mais informações sobre como criar um usuário do IAM, consulte [Criar o primeiro usuário do IAM e o grupo de administradores](#) no Guia do usuário do IAM. Para obter mais informações sobre fazer solicitações, consulte [Fazer solicitações \(p. 10\)](#).

Para obter instruções sobre criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

using Amazon.Runtime;
using Amazon.S3;
using Amazon.S3.Model;
using Amazon.SecurityToken;
using Amazon.SecurityToken.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class TempCredExplicitSessionStartTest
    {
        private const string bucketName = "**** bucket name ****";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
        public static void Main()
        {
            ListObjectsAsync().Wait();
        }

        private static async Task ListObjectsAsync()
        {
            try
            {
                // Credentials use the default AWS SDK for .NET credential search chain.
                // On local development machines, this is your default profile.
                Console.WriteLine("Listing objects stored in a bucket");
                SessionAWSCredentials tempCredentials = await
                    GetTemporaryCredentialsAsync();

                // Create a client by providing temporary security credentials.
                using (s3Client = new AmazonS3Client(tempCredentials, bucketRegion))
                {
                    var listObjectRequest = new ListObjectsRequest
                    {
                        BucketName = bucketName
                    };
                    // Send request to Amazon S3.
                    ListObjectsResponse response = await
                        s3Client.ListObjectsAsync(listObjectRequest);
                }
            }
        }
    }
}
```

```
        List<S3Object> objects = response.S3Objects;
        Console.WriteLine("Object count = {0}", objects.Count);
    }
}
catch (AmazonS3Exception s3Exception)
{
    Console.WriteLine(s3Exception.Message, s3Exception.InnerException);
}
catch (AmazonSecurityTokenServiceException stsException)
{
    Console.WriteLine(stsException.Message, stsException.InnerException);
}
}

private static async Task<SessionAWSCredentials> GetTemporaryCredentialsAsync()
{
    using (var stsClient = new AmazonSecurityTokenServiceClient())
    {
        var getSessionTokenRequest = new GetSessionTokenRequest
        {
            DurationSeconds = 7200 // seconds
        };

        GetSessionTokenResponse sessionTokenResponse =
            await stsClient.GetSessionTokenAsync(getSessionTokenRequest);

        Credentials credentials = sessionTokenResponse.Credentials;

        var sessionCredentials =
            new SessionAWSCredentials(credentials.AccessKeyId,
                                      credentials.SecretAccessKey,
                                      credentials.SessionToken);
        return sessionCredentials;
    }
}
}
```

## Recursos relacionados

- [Usar os AWS SDKs, a CLI e os Explorers \(p. 639\)](#)

## Fazer solicitações usando credenciais temporárias de usuário do IAM ou da conta da AWS - AWS SDK para PHP

Este tópico explica como usar de classes da versão 3 do AWS SDK para PHP para solicitar credenciais de segurança temporárias e usá-las para acessar o Amazon S3. Pressupõe-se que você já esteja seguindo as instruções para [Usar o AWS SDK para PHP e executar exemplos do PHP \(p. 649\)](#) e tenha o AWS SDK para PHP devidamente instalado.

Um usuário do IAM ou uma conta da AWS podem solicitar credenciais de segurança temporárias usando a versão 3 do AWS SDK para PHP. As credenciais temporárias podem então ser usadas para acessar o Amazon S3. As credenciais expiram quando a duração da sessão expira. Por padrão, a duração da sessão é de uma hora. Se usar credenciais de usuário do IAM, especifique a duração (de 1 a 36 horas) ao solicitar as credenciais de segurança temporárias. Para obter mais informações sobre credenciais de segurança temporárias, consulte Credenciais de segurança temporárias em . Para obter mais informações sobre fazer solicitações, consulte [Fazer solicitações \(p. 10\)](#).

### Note

Se você obtiver as credenciais de segurança temporárias usando suas credenciais de segurança de conta da AWS, as credenciais de segurança temporárias serão válidas somente por uma hora. Você poderá especificar a duração de sessão somente se usar credenciais de usuário do IAM para solicitar uma sessão.

### Example

O exemplo de PHP a seguir lista as chaves de objeto no bucket especificado usando credenciais de segurança temporárias. O exemplo obtém credenciais de segurança temporárias para uma sessão padrão de uma hora e usa essas credenciais para enviar uma solicitação autenticada para o Amazon S3. Para obter informações sobre a execução dos exemplos de PHP neste guia, consulte [Executar exemplos do PHP \(p. 649\)](#).

Se você desejar testar o exemplo usando credenciais de usuário do IAM, precisará criar um usuário do IAM em sua conta da AWS. Para obter informações sobre como criar um usuário do IAM, consulte [Criar o primeiro usuário do IAM e o grupo de administradores](#) no Guia do usuário do IAM. Para ver um exemplo de definição de duração de sessão ao usar credenciais de usuário do IAM para solicitar uma sessão, consulte [Fazer solicitações usando credenciais temporárias de usuário federado - AWS SDK para PHP \(p. 40\)](#).

```
<?php

require 'vendor/autoload.php';

use Aws\Sts\StsClient;
use Aws\S3\S3Client;
use Aws\S3\Exception\S3Exception;

$bucket = '*** Your Bucket Name ***';

$sts = new StsClient([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

$sessionToken = $sts->getSessionToken();

$s3 = new S3Client([
    'region'  => 'us-east-1',
    'version' => 'latest',
    'credentials' => [
        'key'      => $sessionToken['Credentials']['AccessKeyId'],
        'secret'   => $sessionToken['Credentials']['SecretAccessKey'],
        'token'    => $sessionToken['Credentials']['SessionToken']
    ]
]);

$result = $s3->listBuckets();

try {
    // Retrieve a paginator for listing objects.
    $objects = $s3->getPaginator('ListObjects', [
        'Bucket' => $bucket
    ]);

    echo "Keys retrieved!" . PHP_EOL;

    // List objects
    foreach ($objects as $object) {
        echo $object['Key'] . PHP_EOL;
    }
}
```

```
| } catch (S3Exception $e) {  
|     echo $e->getMessage() . PHP_EOL;  
| }
```

## Recursos relacionados

- AWS SDK para PHP para classe Aws\S3\S3Client do Amazon S3
- Documentação do AWS SDK para PHP

## Fazer solicitações usando as credenciais temporárias do usuário do IAM - AWS SDK para Ruby

Um usuário do IAM ou uma conta da AWS pode solicitar credenciais de segurança temporárias usando o AWS SDK para Ruby e usá-las para acessar o Amazon S3. Essas credenciais expiram após a duração da sessão. Por padrão, a duração da sessão é de uma hora. Se usar credenciais de usuário do IAM, especifique a duração (de 1 a 36 horas) ao solicitar as credenciais de segurança temporárias. Para obter informações sobre a solicitação de credenciais de segurança temporárias, consulte [Fazer solicitações \(p. 10\)](#).

### Note

Se você obtiver as credenciais de segurança temporárias usando suas credenciais de segurança de conta da AWS, as credenciais de segurança temporárias serão válidas somente por uma hora. Você poderá especificar a duração da sessão somente se usar as credenciais de usuário do IAM para solicitar uma sessão.

O seguinte exemplo de Ruby cria um usuário temporário para listar os itens em um bucket especificado por uma hora. Para usar esse exemplo, você deve ter credenciais da AWS com as permissões necessárias para criar novos clientes do AWS Security Token Service (AWS STS) e listar buckets do Amazon S3.

```
require 'aws-sdk-core'  
require 'aws-sdk-s3'  
require 'aws-sdk-iam'  
  
USAGE = <<DOC  
Usage: assumerole_create_bucket_policy.rb -b BUCKET -u USER [-r REGION] [-d] [-h]  
Assumes a role for USER to list items in BUCKET for one hour.  
BUCKET is required and must already exist.  
USER is required and if not found, is created.  
If REGION is not supplied, defaults to us-west-2.  
-d gives you extra (debugging) information.  
-h displays this message and quits.  
  
DOC  
$debug = false  
  
def print_debug(s)  
  if $debug  
    puts s  
  end
```

```
end

def get_user(region, user_name, create)
  user = nil
  iam = Aws::IAM::Client.new(region: 'us-west-2')

begin
  user = iam.create_user(user_name: user_name)
  iam.wait_until(:user_exists, user_name: user_name)
  print_debug("Created new user #{user_name}")
rescue Aws::IAM::Errors::EntityAlreadyExists
  print_debug("Found user #{user_name} in region #{region}")
end
end

# main
region = 'us-west-2'
user_name = ''
bucket_name = ''

i = 0

while i < ARGV.length
  case ARGV[i]

    when '-b'
      i += 1
      bucket_name = ARGV[i]

    when '-u'
      i += 1
      user_name = ARGV[i]

    when '-r'
      i += 1

      region = ARGV[i]

    when '-d'
      puts 'Debugging enabled'
      $debug = true

    when '-h'
      puts USAGE
      exit 0

    else
      puts 'Unrecognized option: ' + ARGV[i]
      puts USAGE
      exit 1

  end

  i += 1
end

if bucket_name == ''
  puts 'You must supply a bucket name'
  puts USAGE
  exit 1
end

if user_name == ''
  puts 'You must supply a user name'
  puts USAGE
  exit 1
```

```
end

#Identify the IAM user that is allowed to list Amazon S3 bucket items for an hour.
user = get_user(region, user_name, true)

# Create a new Amazon STS client and get temporary credentials. This uses a role that was
already created.
creds = Aws::AssumeRoleCredentials.new(
  client: Aws::STS::Client.new(region: region),
  role_arn: "arn:aws:iam::111122223333:role/assumedrolelist",
  role_session_name: "assumerole-s3-list"
)

# Create an Amazon S3 resource with temporary credentials.
s3 = Aws::S3::Resource.new(region: region, credentials: creds)

puts "Contents of '%s':" % bucket_name
puts '  Name => GUID'

  s3.bucket(bucket_name).objects.limit(50).each do |obj|
    puts "    #{obj.key} => #{obj.etag}"
end
```

## Fazer solicitações usando as credenciais temporárias de usuário federado

Solicite credenciais de segurança temporárias e forneça-as aos aplicativos ou aos usuários federados que precisam de acesso aos recursos da AWS. Esta seção fornece exemplos de como usar o SDK da AWS para obter credenciais de segurança temporárias para os aplicativos ou usuários federados e enviar solicitações autenticadas para o Amazon S3 usando essas credenciais. Para uma lista de SDKs da AWS disponíveis, consulte [Código de exemplo e bibliotecas](#).

### Note

Tanto a conta da AWS quanto um usuário do IAM podem solicitar credenciais de segurança temporárias para usuários federados. No entanto, para maior segurança, somente um usuário do IAM com as permissões necessárias deve solicitar essas credenciais temporárias para garantir que o usuário federado consiga, no máximo, as mesmas permissões do usuário do IAM. Em alguns aplicativos, pode ser apropriado criar um usuário do IAM com permissões específicas com o único propósito de conceder credenciais de segurança temporárias aos aplicativos e aos usuários federados.

## Fazer solicitações usando credenciais temporárias de usuário federado - AWS SDK for Java

Forneça credenciais de segurança temporárias para os aplicativos e os usuários federados a fim de que eles possam enviar solicitações autenticadas para acessar os recursos da AWS. Ao solicitar essas credenciais temporárias, é necessário fornecer um nome de usuário e uma política do IAM que descreve as permissões de recurso que deseja conceder. Por padrão, a duração da sessão é de uma hora. Defina explicitamente um valor de duração diferente ao solicitar as credenciais de segurança temporárias para aplicativos e usuários federados.

### Note

Para garantir segurança adicional ao solicitar credenciais de segurança temporárias para usuários federados e aplicativos, recomendamos que você use um usuário dedicado do IAM apenas com as permissões de acesso necessárias. O usuário temporário criado nunca pode ter mais permissões que o usuário do IAM que solicitou as credenciais de segurança temporárias. Para obter mais informações, consulte [Perguntas frequentes de AWS Identity and Access Management](#).

Para fornecer credenciais de segurança e enviar solicitações autenticadas para acessar recursos, faça o seguinte:

- Crie uma instância da classe `AWSecurityTokenServiceClient`. Para obter informações sobre como fornecer credenciais, consulte [Usar o AWS SDK for Java \(p. 646\)](#).
- Inicie uma sessão chamando o método `getFederationToken()` do cliente Security Token Service (STS). Forneça informações da sessão, incluindo o nome de usuário e uma política do IAM que deseja anexar às credenciais temporárias. Forneça uma duração de sessão opcional. Esse método retorna suas credenciais de segurança temporárias.
- Empacote as credenciais de segurança temporárias em uma instância do objeto `BasicSessionCredentials`. Você usa esse objeto para fornecer as credenciais de segurança temporárias para o cliente do Amazon S3.
- Crie uma instância da classe `AmazonS3Client` usando as credenciais de segurança temporárias. Você envia solicitações ao Amazon S3 usando esse cliente. Se você enviar solicitações usando credenciais expiradas, o Amazon S3 retornará um erro.

## Example

O exemplo lista chaves no bucket especificado do S3. No exemplo, você obtém credenciais de segurança temporárias para uma sessão de duas horas para o seu usuário federado e usa as credenciais para enviar solicitações autenticadas ao Amazon S3. Para executar o exemplo, você precisa criar um usuário do IAM com a política anexada que permite ao usuário solicitar as credenciais de segurança temporárias e listar os recursos da AWS. A política seguinte faz isso:

```
{  
    "Statement": [ {  
        "Action": [ "s3>ListBucket",  
                  "sts:GetFederationToken*"  
                ],  
        "Effect": "Allow",  
        "Resource": "*"  
    }  
]  
}
```

Para obter mais informações sobre como criar um usuário do IAM, consulte [Criar o primeiro usuário do IAM e o grupo de administradores](#) no Guia do usuário do IAM.

Após criar um usuário do IAM e anexar a política anterior, você poderá executar o exemplo a seguir. Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.  
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)  
  
import java.io.IOException;  
  
import com.amazonaws.AmazonServiceException;  
import com.amazonaws.SdkClientException;  
import com.amazonaws.auth.AWSStaticCredentialsProvider;  
import com.amazonaws.auth.BasicSessionCredentials;  
import com.amazonaws.auth.policy.Policy;  
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.S3Actions;  
import com.amazonaws.auth.profile.ProfileCredentialsProvider;  
import com.amazonaws.services.s3.AmazonS3;  
import com.amazonaws.services.s3.AmazonS3ClientBuilder;  
import com.amazonaws.services.securitytoken.AWSIdentityTokenService;  
import com.amazonaws.services.securitytoken.AWSIdentityTokenServiceClientBuilder;  
import com.amazonaws.services.securitytoken.model.Credentials;  
import com.amazonaws.services.securitytoken.model.GetFederationTokenRequest;  
import com.amazonaws.services.securitytoken.model.GetFederationTokenResult;  
import com.amazonaws.services.s3.model.ObjectListing;  
  
public class MakingRequestsWithFederatedTempCredentials {  
  
    public static void main(String[] args) throws IOException {  
        String clientRegion = "*** Client region ***";  
        String bucketName = "*** Specify bucket name ***";  
        String federatedUser = "*** Federated user name ***";  
        String resourceARN = "arn:aws:s3:::" + bucketName;  
  
        try {  
            AWSIdentityTokenService stsClient = AWSIdentityTokenServiceClientBuilder  
                .standard()  
                .withCredentials(new ProfileCredentialsProvider())
```

```
.withRegion(clientRegion)
.build();

GetFederationTokenRequest getFederationTokenRequest = new
GetFederationTokenRequest();
getFederationTokenRequest.setDurationSeconds(7200);
getFederationTokenRequest.setName(federatedUser);

// Define the policy and add it to the request.
Policy policy = new Policy();
policy.withStatements(new Statement(Effect.Allow)
    .withActions(S3Actions.ListObjects)
    .withResources(new Resource(resourceARN)));
getFederationTokenRequest.setPolicy(policy.toJson());

// Get the temporary security credentials.
GetFederationTokenResult federationTokenResult =
stsClient.getFederationToken(getFederationTokenRequest);
Credentials sessionCredentials = federationTokenResult.getCredentials();

// Package the session credentials as a BasicSessionCredentials
// object for an Amazon S3 client object to use.
BasicSessionCredentials basicSessionCredentials = new BasicSessionCredentials(
    sessionCredentials.getAccessKeyId(),
    sessionCredentials.getSecretAccessKey(),
    sessionCredentials.getSessionToken());
    AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
        .withCredentials(new
AWSStaticCredentialsProvider(basicSessionCredentials))
        .withRegion(clientRegion)
        .build();

// To verify that the client works, send a listObjects request using
// the temporary security credentials.
ObjectListing objects = s3Client.listObjects(bucketName);
System.out.println("No. of Objects = " + objects.getObjectSummaries().size());
}

catch(AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
}
catch(SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

## Recursos relacionados

- [Usar os AWS SDKs, a CLI e os Explorers \(p. 639\)](#)

## Fazer solicitações usando credenciais temporárias de usuário federado - AWS SDK para .NET

Forneça credenciais de segurança temporárias para os aplicativos e os usuários federados a fim de que eles possam enviar solicitações autenticadas para acessar os recursos da AWS. Ao solicitar essas

credenciais temporárias, é necessário fornecer um nome de usuário e uma política do IAM que descreve as permissões de recurso que deseja conceder. Por padrão, a duração de uma sessão é de uma hora. Defina explicitamente um valor de duração diferente ao solicitar as credenciais de segurança temporárias para aplicativos e usuários federados. Para obter informações sobre o envio de solicitações autenticadas, consulte [Fazer solicitações \(p. 10\)](#).

**Note**

Ao solicitar credenciais de segurança temporárias para usuários federados e aplicativos a fim de garantir segurança adicional, recomendamos que você use um usuário dedicado do IAM apenas com as permissões de acesso necessárias. O usuário temporário criado nunca pode ter mais permissões que o usuário do IAM que solicitou as credenciais de segurança temporárias. Para obter mais informações, consulte [Perguntas frequentes de AWS Identity and Access Management](#).

Faça o seguinte:

- Crie uma instância de cliente AWS Security Token Service, classe `AmazonSecurityTokenServiceClient`. Para obter informações sobre a concessão de credenciais, consulte [Usar o AWS SDK para .NET \(p. 647\)](#).
- Inicie uma sessão chamando o método `GetFederationToken` do cliente STS. Você deverá fornecer informações da sessão, incluindo o nome de usuário e uma política do IAM que deseja anexar às credenciais temporárias. Como opção, você pode fornecer uma duração de sessão. Esse método retorna suas credenciais de segurança temporárias.
- Empacote as credenciais de segurança temporárias em uma instância do objeto `SessionAWSCredentials`. Você usa esse objeto para fornecer as credenciais de segurança temporárias para o cliente do Amazon S3.
- Crie uma instância da classe `AmazonS3Client` enviando as credenciais de segurança temporárias. Use este cliente para enviar solicitações ao Amazon S3. Se você enviar solicitações usando credenciais expiradas, o Amazon S3 retornará um erro.

**Example**

O exemplo do C# a seguir lista as chaves no bucket especificado. No exemplo, você obtém credenciais de segurança temporárias para uma sessão de duas horas para o seu usuário federado (User1), e usa as credenciais para enviar solicitações autenticadas ao Amazon S3.

- Neste exercício, você criará um usuário do IAM com permissões mínimas. Usando as credenciais desse usuário do IAM, solicite credenciais temporárias para terceiros. Este exemplo lista somente os objetos em um bucket específico. Crie um usuário do IAM com a política a seguir anexada:

```
{  
    "Statement": [  
        {  
            "Action": ["s3>ListBucket",  
                      "sts:GetFederationToken*"  
                    ],  
            "Effect": "Allow",  
            "Resource": "*"  
        }  
    ]  
}
```

A política permite que o usuário do IAM solicite credenciais de segurança temporárias e permissão de acesso apenas para listar os recursos da AWS. Para obter mais informações sobre como criar um usuário do IAM, consulte [Criar o primeiro usuário do IAM e o grupo de administradores](#) no Guia do usuário do IAM.

- Use as credenciais de segurança do usuário do IAM para testar o exemplo a seguir. O exemplo envia solicitação autenticada para o Amazon S3 usando credenciais de segurança temporárias. O exemplo

especifica a política a seguir ao solicitar credenciais de segurança temporárias para o usuário federado (User1), que restringe o acesso aos objetos de lista em um bucket específico (YourBucketName). É necessário atualizar a política e fornecer um nome de bucket existente.

```
{  
    "Statement": [  
        {  
            "Sid": "1",  
            "Action": ["s3>ListBucket"],  
            "Effect": "Allow",  
            "Resource": "arn:aws:s3:::YourBucketName"  
        }  
    ]  
}
```

- **Example**

Atualize o exemplo a seguir e forneça o nome de bucket especificado na política de acesso do usuário federado anterior. Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.  
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)  
  
using Amazon.Runtime;  
using Amazon.S3;  
using Amazon.S3.Model;  
using Amazon.SecurityToken;  
using Amazon.SecurityToken.Model;  
using System;  
using System.Collections.Generic;  
using System.Threading.Tasks;  
  
namespace Amazon.DocSamples.S3  
{  
    class TempFederatedCredentialsTest  
    {  
        private const string bucketName = "*** bucket name ***";  
        // Specify your bucket region (an example region is shown).  
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;  
        private static IAmazonS3 client;  
  
        public static void Main()  
        {  
            ListObjectsAsync().Wait();  
        }  
  
        private static async Task ListObjectsAsync()  
        {  
            try  
            {  
                Console.WriteLine("Listing objects stored in a bucket");  
                // Credentials use the default AWS SDK for .NET credential search chain.  
                // On local development machines, this is your default profile.  
                SessionAWSCredentials tempCredentials =  
                    await GetTemporaryFederatedCredentialsAsync();  
  
                // Create a client by providing temporary security credentials.  
                using (client = new AmazonS3Client(bucketRegion))  
                {  
                    ListObjectsRequest listObjectRequest = new ListObjectsRequest();  
                    listObjectRequest.BucketName = bucketName;  
                }  
            }  
        }  
    }  
}
```

```
        ListObjectsResponse response = await
client.ListObjectsAsync(listObjectRequest);
List<S3Object> objects = response.S3Objects;
Console.WriteLine("Object count = {0}", objects.Count);

        Console.WriteLine("Press any key to continue...");
        Console.ReadKey();
    }
}
catch (AmazonS3Exception e)
{
    Console.WriteLine("Error encountered ***. Message:{0}' when writing an
object", e.Message);
}
catch (Exception e)
{
    Console.WriteLine("Unknown encountered on server. Message:{0}' when
writing an object", e.Message);
}
}

private static async Task<SessionAWSCredentials>
GetTemporaryFederatedCredentialsAsync()
{
    AmazonSecurityTokenServiceConfig config = new
AmazonSecurityTokenServiceConfig();
    AmazonSecurityTokenServiceClient stsClient =
        new AmazonSecurityTokenServiceClient(
            config);

    GetFederationTokenRequest federationTokenRequest =
        new GetFederationTokenRequest();
    federationTokenRequest.DurationSeconds = 7200;
    federationTokenRequest.Name = "User1";
    federationTokenRequest.Policy = @{
        ""Statement"":
        [
            {
                ""Sid"":"""Stmt1311212314284""",
                ""Action"":[""s3>ListBucket""],
                ""Effect"":""Allow"",
                ""Resource"":""arn:aws:s3:::" + bucketName + @"""
            }
        ]
    };
};

GetFederationTokenResponse federationTokenResponse =
    await stsClient.GetFederationTokenAsync(federationTokenRequest);
Credentials credentials = federationTokenResponse.Credentials;

SessionAWSCredentials sessionCredentials =
    new SessionAWSCredentials(credentials.AccessKeyId,
                               credentials.SecretAccessKey,
                               credentials.SessionToken);
return sessionCredentials;
}
}
```

## Recursos relacionados

- [Usar os AWS SDKs, a CLI e os Explorers \(p. 639\)](#)

## Fazer solicitações usando credenciais temporárias de usuário federado - AWS SDK para PHP

Este tópico explica como usar classes da versão 3 do AWS SDK para PHP para solicitar credenciais de segurança temporárias para aplicativos e usuários federados, e usá-las para acessar recursos armazenados no Amazon S3. Pressupõe-se que você já esteja seguindo as instruções para [Usar o AWS SDK para PHP e executar exemplos do PHP \(p. 649\)](#) e tenha o AWS SDK para PHP devidamente instalado.

Forneça credenciais de segurança temporárias para os aplicativos e os usuários federados para que eles possam enviar solicitações autenticadas para acessar os recursos da AWS. Ao solicitar essas credenciais temporárias, é necessário fornecer um nome de usuário e uma política do IAM que descreve as permissões de recurso que deseja conceder. Essas credenciais expiram quando a duração da sessão expira. Por padrão, a duração da sessão é de uma hora. Defina explicitamente um valor de duração diferente ao solicitar as credenciais de segurança temporárias para aplicativos e usuários federados. Para obter mais informações sobre credenciais de segurança temporárias, consulte [Credenciais de segurança temporárias](#) em Guia do usuário do IAM. Para obter informações sobre como fornecer credenciais de segurança temporárias para aplicativos e usuários federados, consulte [Fazer solicitações \(p. 10\)](#).

Para garantir segurança adicional ao solicitar credenciais de segurança temporárias para usuários federados e aplicativos, recomendamos que você use um usuário dedicado do IAM apenas com as permissões de acesso necessárias. O usuário temporário criado nunca pode ter mais permissões que o usuário do IAM que solicitou as credenciais de segurança temporárias. Para obter informações sobre a federação de identidades, consulte [Perguntas frequentes de AWS Identity and Access Management](#).

Para obter informações sobre a execução dos exemplos de PHP neste guia, consulte [Executar exemplos do PHP \(p. 649\)](#).

### Example

O exemplo PHP a seguir lista chaves no bucket especificado. No exemplo, você obtém credenciais de segurança temporárias para uma sessão de uma hora para o usuário federado (User1). Em seguida, você usa as credenciais de segurança temporárias para enviar solicitações autenticadas para o Amazon S3.

Para maior segurança, ao solicitar credenciais temporárias para outros, use as credenciais de segurança de um usuário do IAM com permissões para solicitar credenciais de segurança temporárias. Para garantir que o usuário do IAM conceda apenas as permissões mínimas específicas do aplicativo ao usuário federado, você pode limitar as permissões de acesso desse usuário do IAM. Este exemplo lista somente objetos em um bucket específico. Crie um usuário do IAM com a política a seguir anexada:

```
{  
    "Statement": [ {  
        "Action": [ "s3>ListBucket",  
                  "sts:GetFederationToken*"  
                ],  
        "Effect": "Allow",  
        "Resource": "*"  
    }  
]
```

A política permite que o usuário do IAM solicite credenciais de segurança temporárias e permissão de acesso apenas para listar os recursos da AWS. Para obter mais informações sobre como criar um usuário do IAM, consulte [Criar o primeiro usuário do IAM e o grupo de administradores](#) no Guia do usuário do IAM.

Agora use as credenciais de segurança do usuário do IAM para testar o exemplo a seguir. O exemplo envia uma solicitação autenticada para o Amazon S3 usando credenciais de segurança temporárias. Ao solicitar credenciais de segurança temporárias para o usuário federado (User1), o exemplo especifica a política a seguir, que restringe o acesso aos objetos de lista em um bucket específico. Atualizar a política com o nome do seu bucket.

```
{  
    "Statement": [  
        {  
            "Sid": "1",  
            "Action": ["s3>ListBucket"],  
            "Effect": "Allow",  
            "Resource": "arn:aws:s3:::YourBucketName"  
        }  
    ]  
}
```

No exemplo a seguir, ao especificar o recurso de política, substitua *YourBucketName* pelo seu próprio nome de bucket:

```
<?php  
  
require 'vendor/autoload.php';  
  
use Aws\Sts\StsClient;  
use Aws\S3\S3Client;  
use Aws\S3\Exception\S3Exception;  
  
$bucket = '*** Your Bucket Name ***';  
  
// In real applications, the following code is part of your trusted code. It has  
// the security credentials that you use to obtain temporary security credentials.  
$sts = new StsClient(  
    [  
        'version' => 'latest',  
        'region' => 'us-east-1'  
];  
  
// Fetch the federated credentials.  
$sessionToken = $sts->getFederationToken([  
    'Name'          => 'User1',  
    'DurationSeconds' => '3600',  
    'Policy'         => json_encode([  
        'Statement' => [  
            'Sid'           => 'randomstatementid' . time(),  
            'Action'        => ['s3>ListBucket'],  
            'Effect'        => 'Allow',  
            'Resource'      => 'arn:aws:s3:::' . $bucket  
        ]  
    ])  
]);  
  
// The following will be part of your less trusted code. You provide temporary  
// security credentials so the code can send authenticated requests to Amazon S3.  
  
$s3 = new S3Client([  
    'region' => 'us-east-1',  
    'version' => 'latest',  
    'credentials' => [  
        'key'    => $sessionToken['Credentials']['AccessKeyId'],  
        'secret' => $sessionToken['Credentials']['SecretAccessKey'],  
        'token'  => $sessionToken['Credentials']['SessionToken']  
    ]
```

```
]);
try {
    $result = $s3->listObjects([
        'Bucket' => $bucket
    ]);
} catch (S3Exception $e) {
    echo $e->getMessage() . PHP_EOL;
}
```

## Recursos relacionados

- [AWS SDK para PHP para classe Aws\S3\S3Client do Amazon S3](#)
- [Documentação do AWS SDK para PHP](#)

## Fazer solicitações usando credenciais temporárias de usuário federado - AWS SDK para Ruby

Forneça credenciais de segurança temporárias para os aplicativos e os usuários federados a fim de que eles possam enviar solicitações autenticadas para acessar os recursos da AWS. Ao solicitar essas credenciais temporárias do serviço do IAM, forneça um nome de usuário e uma política do IAM que descreva as permissões de recurso que você deseja conceder. Por padrão, a duração da sessão é de uma hora. No entanto, se estiver solicitando credenciais temporárias usando credenciais de usuário do IAM, defina explicitamente um valor de duração diferente ao solicitar as credenciais de segurança temporárias para usuários federados e aplicativos. Para obter informações sobre credenciais de segurança temporárias para aplicativos e usuários federados, consulte [Fazer solicitações \(p. 10\)](#).

### Note

Para garantir segurança adicional ao solicitar credenciais de segurança temporárias para usuários federados e aplicativos, use um usuário dedicado do IAM apenas com as permissões de acesso necessárias. O usuário temporário criado nunca pode ter mais permissões que o usuário do IAM que solicitou as credenciais de segurança temporárias. Para obter mais informações, consulte [Perguntas frequentes de AWS Identity and Access Management](#).

### Example

O exemplo de código Ruby a seguir permite que um usuário federado com um conjunto limitado de permissões listar as chaves no bucket específico.

```
require 'aws-sdk-s3'
require 'aws-sdk-iam'

USAGE = <<DOC

Usage: federated_create_bucket_policy.rb -b BUCKET -u USER [-r REGION] [-d] [-h]

Creates a federated policy for USER to list items in BUCKET for one hour.

BUCKET is required and must already exist.

USER is required and if not found, is created.

If REGION is not supplied, defaults to us-west-2.

-d gives you extra (debugging) information.

-h displays this message and quits.

DOC

$debug = false

def print_debug(s)
  if $debug
    puts s
  end
end

def get_user(region, user_name, create)
  user = nil
  iam = Aws::IAM::Client.new(region: 'us-west-2')

begin
  user = iam.create_user(user_name: user_name)
  iam.wait_until(:user_exists, user_name: user_name)
end

  user
end
```

```
    print_debug("Created new user #{user_name}")
rescue Aws::IAM::Errors::EntityAlreadyExists
    print_debug("Found user #{user_name} in region #{region}")
end
end

# main
region = 'us-west-2'
user_name = ''
bucket_name = ''

i = 0

while i < ARGV.length
  case ARGV[i]

    when '-b'
      i += 1
      bucket_name = ARGV[i]

    when '-u'
      i += 1
      user_name = ARGV[i]

    when '-r'
      i += 1
      region = ARGV[i]

    when '-d'
      puts 'Debugging enabled'
      $debug = true

    when '-h'
      puts USAGE
      exit 0

    else
      puts 'Unrecognized option: ' + ARGV[i]
      puts USAGE
      exit 1
  end

  i += 1
end

if bucket_name == ''
  puts 'You must supply a bucket name'
  puts USAGE
  exit 1
end

if user_name == ''
  puts 'You must supply a user name'
  puts USAGE
  exit 1
end

#Identify the IAM user we allow to list Amazon S3 bucket items for an hour.
user = get_user(region, user_name, true)

# Create a new STS client and get temporary credentials.
sts = Aws::STS::Client.new(region: region)

creds = sts.get_federation_token({
```

```
duration_seconds: 3600,  
name: user_name,  
policy: "{\"Version\":\"2012-10-17\", \"Statement\":[{\"Sid\":\"Stmt1\", \"Effect\":\"Allow\", \"Action\":\"s3>ListBucket\", \"Resource\":\"arn:aws:s3:::{bucket_name}\"]}]}",  
}  
  
# Create an Amazon S3 resource with temporary credentials.  
s3 = Aws::S3::Resource.new(region: region, credentials: creds)  
  
puts "Contents of '%s':" % bucket_name  
puts '  Name => GUID'  
  
s3.bucket(bucket_name).objects.limit(50).each do |obj|  
  puts "    #{obj.key} => #{obj.etag}"  
end
```

## Fazer solicitações usando a API REST

Esta seção contém informações sobre como fazer solicitações para endpoints do Amazon S3 usando a API REST. Para obter uma lista dos endpoints do Amazon S3, consulte [Regiões e endpoints](#) no AWS General Reference.

### Tópicos

- [Fazer solicitações para endpoints de pilha dupla usando a API REST \(p. 46\)](#)
- [Hospedagem virtual de buckets \(p. 46\)](#)
- [Redirecionamento de solicitação e a API REST \(p. 51\)](#)

Ao fazer solicitações usando a API REST, use URIs no estilo de hospedagem virtual ou de caminho para os endpoints do Amazon S3. Para obter mais informações, consulte [Trabalho com buckets do Amazon S3 \(p. 54\)](#).

### Example Solicitação no estilo de hospedagem virtual

Veja a seguir um exemplo de solicitação no estilo de hospedagem virtual para excluir o arquivo – do bucket chamado examplebucket.

```
DELETE /puppy.jpg HTTP/1.1  
Host: examplebucket.s3-us-west-2.amazonaws.com  
Date: Mon, 11 Apr 2016 12:00:00 GMT  
x-amz-date: Mon, 11 Apr 2016 12:00:00 GMT  
Authorization: authorization string
```

### Example Solicitação no estilo de caminho

Veja a seguir um exemplo com a versão no estilo de caminho da mesma solicitação.

```
DELETE /examplebucket/puppy.jpg HTTP/1.1  
Host: s3-us-west-2.amazonaws.com  
Date: Mon, 11 Apr 2016 12:00:00 GMT  
x-amz-date: Mon, 11 Apr 2016 12:00:00 GMT  
Authorization: authorization string
```

O Amazon S3 oferece suporte ao acesso no estilo de caminho e no estilo hospedado virtual em todas as regiões. A sintaxe de caminho- estilo, porém, requer que você use o endpoint específico da região ao

tentar acessar um bucket. Por exemplo, se houver um bucket chamado `mybucket` que resida na região UE (Irlanda) você querer usar a sintaxe de caminho- estilo e o objeto for chamado `puppy.jpg`, o URI correto será `http://s3-eu-west-1.amazonaws.com/mybucket/puppy.jpg`.

Você receberá um erro de redirecionamento temporário com código 307 de resposta HTTP e uma mensagem indicando o URI correto para o seu recurso, caso tente acessar um bucket fora da região Leste dos EUA (N. Virginia) com sintaxe de caminho- estilo que use qualquer um dos seguintes:

- `http://s3.amazonaws.com`
- Um endpoint para uma região diferente daquela na qual reside o bucket. Por exemplo, se você usar `http://s3-eu-west-1.amazonaws.com` para um bucket que foi criado na região Oeste dos EUA (Norte da Califórnia).

## Fazer solicitações para endpoints de pilha dupla usando a API REST

Ao usar a API REST, acesse um endpoint de pilha dupla diretamente usando um nome de endpoint (URI) de hospedagem virtual ou de caminho. Todos os nomes de endpoint de pilha dupla do Amazon S3 incluem a região. Diferente dos endpoints somente-IPv4 padrão, os endpoints de hospedagem virtual e de caminho usam nomes de endpoint específicos para a região.

Example Solicitação de endpoint de pilha dupla do estilo de hospedagem virtual

Conforme mostrado no exemplo a seguir, use um endpoint no estilo de hospedagem virtual na solicitação REST que recupera o objeto do bucket chamado `puppy.jpg` do bucket `examplebucket`.

```
GET /puppy.jpg HTTP/1.1
Host: examplebucket.s3.dualstack.us-west-2.amazonaws.com
Date: Mon, 11 Apr 2016 12:00:00 GMT
x-amz-date: Mon, 11 Apr 2016 12:00:00 GMT
Authorization: authorization string
```

Example Solicitação de endpoint de pilha dupla no estilo de caminho

Ou use um endpoint no de estilo caminho na solicitação, conforme mostrado no exemplo a seguir.

```
GET /examplebucket/puppy.jpg HTTP/1.1
Host: s3.dualstack.us-west-2.amazonaws.com
Date: Mon, 11 Apr 2016 12:00:00 GMT
x-amz-date: Mon, 11 Apr 2016 12:00:00 GMT
Authorization: authorization string
```

Para obter mais informações sobre endpoints de pilha dupla, consulte [Usar endpoints de pilha dupla do Amazon S3 \(p. 14\)](#).

## Hospedagem virtual de buckets

### Tópicos

- [Especificação de bucket do cabeçalho de host HTTP \(p. 47\)](#)
- [Exemplos \(p. 48\)](#)
- [Personalizar URLs do Amazon S3 com CNAMEs \(p. 49\)](#)
- [Limitações \(p. 51\)](#)

- [Compatibilidade retroativa \(p. 51\)](#)

Em geral, hospedagem virtual é a prática de atender vários sites a partir de um único servidor web. Uma maneira de diferenciar sites é usar o nome de host aparente da solicitação em vez de apenas a parte do caminho do URI correspondente ao nome. Uma solicitação REST comum do Amazon S3 especifica um bucket usando o primeiro componente delimitado por barra do caminho do URI da solicitação. Como alternativa, use a hospedagem virtual do Amazon S3 para endereçar um bucket em uma chamada de API REST usando o cabeçalho Host HTTP. Na prática, o Amazon S3 interpreta Host como um aviso de que a maioria dos buckets é acessível automaticamente (para tipos limitados de solicitações) em `http://bucketname.s3.amazonaws.com`. Além disso, ao nomear o bucket com o nome do domínio registrado e ao tornar esse nome um alias do DNS para o Amazon S3, você pode personalizar totalmente o URL dos recursos do Amazon S3, por exemplo, `http://my.bucketname.com/`.

Além do poder atrativo dos URLs personalizados, um segundo benefício da hospedagem virtual é a habilidade de publicação no "diretório raiz" do servidor virtual do bucket. Esta habilidade pode ser importante pois muitos aplicativos buscam arquivos nesse local padrão. Por exemplo, `favicon.ico`, `robots.txt`, `crossdomain.xml` serão todos encontrados na raiz.

#### Important

O Amazon S3 oferece suporte ao acesso no estilo de caminho e no estilo hospedado virtual em todas as regiões. A sintaxe de caminho- estilo, porém, requer que você use o endpoint específico da região ao tentar acessar um bucket. Por exemplo, se houver um bucket chamado `mybucket` que resida na região UE (Irlanda) você querer usar a sintaxe de caminho- estilo e o objeto for chamado `puppy.jpg`, o URI correto será `http://s3-eu-west-1.amazonaws.com/mybucket/puppy.jpg`.

Você receberá um erro de redirecionamento temporário com código 307 de resposta HTTP e uma mensagem indicando o URI correto para o seu recurso, caso tente acessar um bucket fora da região Leste dos EUA (N. Virginia) com sintaxe de caminho- estilo que use qualquer um dos seguintes:

- `http://s3.amazonaws.com`
- Um endpoint para uma região diferente daquela na qual reside o bucket. Por exemplo, se você usar `http://s3-eu-west-1.amazonaws.com` para um bucket que foi criado na região Oeste dos EUA (Norte da Califórnia).

#### Note

O Amazon S3 encaminha todas as solicitações hospedadas virtualmente para a região Leste dos EUA (Norte da Virgínia), por padrão, se você usar o endpoint Leste dos EUA (Norte da Virgínia) (`s3.amazonaws.com`), em vez do endpoint específico da região (por exemplo, `s3-eu-west-1.amazonaws.com`). Quando você cria um bucket, em qualquer região, o Amazon S3 atualiza o DNS para encaminhar novamente a solicitação para o local correto, o que pode levar algum tempo. Enquanto isso, as regras padrão se aplicam e a solicitação hospedada virtualmente vai para a região Leste dos EUA (Norte da Virgínia), e o Amazon S3 a redireciona para a região correta com HTTP 307. Para obter mais informações, consulte [Redirecionamento de solicitação e a API REST \(p. 590\)](#).

Ao usar bucket hospedados virtualmente com SSL, o certificado curinga SSL corresponde apenas buckets que não contêm pontos. Para contornar isso, use HTTP ou escreva a sua própria lógica de verificação do certificado.

## Especificação de bucket do cabeçalho de host HTTP

Desde que a solicitação GET não use o endpoint SSL, você pode especificar o bucket para a solicitação usando o cabeçalho Host HTTP. O cabeçalho Host em uma solicitação REST é interpretado da seguinte forma:

- Se o cabeçalho `Host` estiver omitido ou se o seu valor for 's3.amazonaws.com', o bucket para a solicitação será o primeiro componente delimitado por barra no URI da solicitação e a chave da solicitação será o restante do URI. Este é o método comum, conforme ilustrado pelos dois primeiros exemplos desta seção. Omitir o cabeçalho `Host` é válido apenas para solicitações HTTP 1.0.
- Caso contrário, se o valor do cabeçalho `Host` terminar com 's3.amazonaws.com', o nome do bucket será o componente inicial do valor do cabeçalho `Host` até 's3.amazonaws.com'. A chave da solicitação será o seu URI. Essa interpretação expõe buckets como subdomínios do s3.amazonaws.com, conforme ilustrado pelos exemplos 3 e 4 nesta seção.
- Caso contrário, o bucket da solicitação é o valor em letras minúsculas do cabeçalho `Host`, e a chave da solicitação é o seu URI. Essa interpretação é útil se você tiver registrado o mesmo nome DNS que o nome do bucket e se tiver configurado o nome para ser um alias CNAME para o Amazon S3. O procedimento para registrar nomes de domínios e configurar o DNS está além do escopo deste guia, mas o resultado é ilustrado pelo último exemplo desta seção.

## Exemplos

Esta seção fornece exemplos de URLs e solicitações.

### Example Método de estilo de caminho

Este exemplo usa `johndoe.net` como nome do bucket e `homepage.html` como nome da chave.

O URL é o seguinte:

```
http://s3.amazonaws.com/johndoe.net/homepage.html
```

A solicitação é a seguinte:

```
GET /johndoe.net/homepage.html HTTP/1.1
Host: s3.amazonaws.com
```

A solicitação com HTTP 1.0 e a omissão do cabeçalho `host` é a seguinte:

```
GET /johndoe.net/homepage.html HTTP/1.0
```

Para obter informações sobre nomes compatíveis do DNS, consulte [Limitações \(p. 51\)](#). Para obter mais informações sobre chaves, consulte [Chaves \(p. 3\)](#).

### Example Método do estilo hospedagem virtual

Este exemplo usa `johndoe.net` como nome do bucket e `homepage.html` como nome da chave.

O URL é o seguinte:

```
http://johndoe.net.s3.amazonaws.com/homepage.html
```

A solicitação é a seguinte:

```
GET /homepage.html HTTP/1.1
Host: johndoe.net.s3.amazonaws.com
```

O método do estilo hospedagem virtual requer que o nome do bucket esteja em conformidade com o DNS.

Example Método do estilo hospedagem virtual para um bucket que não esteja na região Leste dos EUA (Norte da Virgínia)

Este exemplo usa `johnsmith.eu` como nome do bucket na região UE (Irlanda) e `homepage.html` como nome da chave.

O URL é o seguinte:

```
http://johnsmith.eu.s3-eu-west-1.amazonaws.com/homepage.html
```

A solicitação é a seguinte:

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

Observe que, em vez de usar o endpoint específico para a região, também é possível usar o endpoint da região Leste dos EUA (Norte da Virgínia), não importando a região de residência do bucket.

```
http://johnsmith.eu.s3.amazonaws.com/homepage.html
```

A solicitação é a seguinte:

```
GET /homepage.html HTTP/1.1
Host: johnsmith.eu.s3.amazonaws.com
```

Example Método CNAME

Este exemplo usa `www.johnsmith.net` como nome do bucket e `homepage.html` como nome da chave. Para usar este método, é necessário configurar o nome DNS como um alias CNAME para `bucketname.s3.amazonaws.com`.

O URL é o seguinte:

```
http://www.johnsmith.net/homepage.html
```

O exemplo é o seguinte:

```
GET /homepage.html HTTP/1.1
Host: www.johnsmith.net
```

## Personalizar URLs do Amazon S3 com CNAMEs

Dependendo da necessidade, é possível que você não queira que "s3.amazonaws.com" apareça no site ou serviço. Por exemplo, se você hospedar as imagens do site no Amazon S3, talvez prefira `http://images.johnsmith.net/` em vez de `http://johnsmith-images.s3.amazonaws.com/`.

O nome do bucket deve ser o mesmo que o CNAME. Portanto, `http://images.johnsmith.net/filename` seria o mesmo que `http://images.johnsmith.net.s3.amazonaws.com/filename` se um CNAME fosse criado para mapear `images.johnsmith.net` para `images.johnsmith.net.s3.amazonaws.com`.

Qualquer bucket com um nome compatível com o DNS pode ser mencionado da seguinte forma: `http://[BucketName].s3.amazonaws.com/[Filename]`, por exemplo, `http://images.johnsmith.net.s3.amazonaws.com/mydog.jpg`. Ao usar o CNAME, mapeie

`images.johnsmith.net` para um nome de host do Amazon S3 para que o URL anterior se torne `http://images.johnsmith.net/mydog.jpg`.

O registro DNS do CNAME deve apelidar o nome do domínio para o nome de host apropriado do estilo hospedagem virtual. Por exemplo, se o nome do bucket e o nome do domínio são `images.johnsmith.net`, o alias do registro do CNAME deve ser `images.johnsmith.net.s3.amazonaws.com`.

```
images.johnsmith.net CNAME images.johnsmith.net.s3.amazonaws.com.
```

Configurar o destino de alias para `s3.amazonaws.com` também funciona, mas pode resultar em redirecionamentos HTTP extras.

O Amazon S3 usa o nome de host para determinar o nome do bucket. Por exemplo, suponha que você tenha configurado `www.example.com` como um CNAME para `www.example.com.s3.amazonaws.com`. Quando você acessa `http://www.example.com`, o Amazon S3 recebe uma solicitação semelhante à seguinte:

#### Example

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

Como o Amazon S3 enxerga apenas o nome de host original `www.example.com` e não tem conhecimento sobre o mapeamento CNAME usado para resolver a solicitação, o CNAME e o nome do bucket devem ser os mesmos.

Qualquer endpoint do Amazon S3 pode ser usado em um CNAME. Por exemplo, `s3-ap-southeast-1.amazonaws.com` pode ser usado em CNAMEs. Para obter mais informações sobre endpoints, consulte [Endpoints de solicitações \(p. 11\)](#).

Para associar um nome de host a um bucket do Amazon S3 usando CNAMEs

1. Selecione um nome de host que pertença a um domínio controlado por você. Este exemplo usa o subdomínio `images` do domínio `johnsmith.net`.
2. Crie um bucket que corresponda ao nome de host. Neste exemplo, os nomes de host e do bucket são `images.johnsmith.net`.

#### Note

O nome do bucket deve ser exatamente igual ao nome de host.

3. Crie um registro do CNAME que define o nome de host como um alias para o bucket do Amazon S3. Por exemplo:

```
images.johnsmith.net CNAME images.johnsmith.net.s3.amazonaws.com
```

#### Important

Por questões de encaminhamento de solicitações, o registro do CNAME deve estar definido exatamente como mostrado no exemplo anterior. Caso contrário, a operação pode parecer correta, mas, eventualmente, resultará em comportamento imprevisível.

#### Note

O procedimento para configurar o DNS depende do servidor DNS ou do provedor DNS. Para obter informações específicas, consulte a documentação do servidor ou entre em contato com o provedor.

## Limitações

Especificar o bucket para a solicitação usando o cabeçalho `Host` HTTP é compatível com solicitações que não sejam SSL e ao usar a API REST. Você não pode especificar o bucket no SOAP usando um endpoint diferente.

### Note

O suporte de SOAP via HTTP está obsoleto, mas continua disponível via HTTPS. Novos recursos do Amazon S3 não serão compatíveis com o SOAP. Recomendamos que você use o REST API ou os SDKs da AWS.

## Compatibilidade retroativa

As versões anteriores do Amazon S3 ignoravam de maneira incorreta o cabeçalho `Host` HTTP. Os aplicativos que dependem desse comportamento não documentado devem ser atualizados para definir o cabeçalho `Host` corretamente. Como o Amazon S3 determina o nome do bucket a partir do `Host` quando está presente, o sintoma mais comum desse problema é receber um código de resultado de erro `NoSuchBucket` inesperado.

## Redirecionamento de solicitação e a API REST

### Tópicos

- [Redirecionamentos e agentes de usuário de HTTP \(p. 51\)](#)
- [Redirecionamentos e 100-Continue \(p. 52\)](#)
- [Exemplo de redirecionamento \(p. 52\)](#)

Esta seção descreve como processar redirecionamentos HTTP usando a API REST do Amazon S3. Para obter informações gerais sobre os redirecionamentos do Amazon S3, consulte [Redirecionamento de solicitação e a API REST \(p. 590\)](#) no Amazon Simple Storage Service API Reference.

## Redirecionamentos e agentes de usuário de HTTP

Os programas que usam a API REST do Amazon S3 devem processar os redirecionamentos na camada de aplicativo ou na camada HTTP. Muitas bibliotecas de clientes HTTP e agentes de usuário podem ser configurados para processar redirecionamentos de modo correto e automático; contudo, muitas outras têm implementações de redirecionamento incorretas ou incompletas.

Antes de confiar em uma biblioteca para atender aos requisitos de redirecionamento, teste os seguintes casos:

- Verifique se todos os cabeçalhos de solicitações HTTP estão incluídos corretamente na solicitação redirecionada (a segunda solicitação depois de receber um redirecionamento), incluindo os padrões HTTP como Autorização e Data.
- Verifique se redirecionamentos não GET, como PUT e DELETE, funcionam corretamente.
- Verifique se grandes solicitações PUT seguem o redirecionamento corretamente.
- Verifique se as solicitações PUT seguem redirecionamentos corretamente se a resposta 100-continue demorar muito tempo para chegar.

Os agentes de usuário HTTP que se conformam estritamente a RFC 2616 podem exigir confirmação explícita antes de seguir um redirecionamento quando o método de solicitação HTTP não for GET nem HEAD. Em geral, é seguro seguir redirecionamentos gerados pelo Amazon S3 automaticamente, pois o sistema emitirá redirecionamentos somente para hosts no domínio `amazonaws.com` e o efeito da solicitação redirecionada será igual ao da solicitação original.

## Redirecionamentos e 100-Continue

Para simplificar o processamento de redirecionamentos, aumentar a eficiência e evitar custos associados com o envio de um corpo de solicitação redirecionado duas vezes, configure seu aplicativo para usar 100-continues para operações PUT. Quando seu aplicativo usa 100-continue, ele não envia o corpo da solicitação até receber uma confirmação. Se a mensagem for rejeitada com base nos cabeçalhos, o corpo da mensagem não será enviado. Para obter mais informações sobre 100-continue, acesse [RFC 2616 Section 8.2.3](#).

### Note

De acordo com o RFC 2616, ao usar `Expect: Continue` com um servidor HTTP desconhecido, você não deve esperar um período indefinido antes de enviar o corpo da solicitação. Isso porque alguns servidores HTTP não reconhecem 100-continue. Porém, o Amazon S3 reconhecerá se sua solicitação contiver um `Expect: Continue` e responderá com um status 100-continue temporário ou um código de status final. Além disso, um erro de não redirecionamento ocorrerá depois de receber o 100-continue temporário. Isto ajudará a evitar que você receba uma resposta de redirecionamento enquanto ainda estiver escrevendo o corpo da solicitação.

## Exemplo de redirecionamento

Esta seção fornece um exemplo de interação de servidor cliente usando redirecionamento HTTP e 100-continue.

A seguir está um exemplo PUT para o bucket `quotes.s3.amazonaws.com`.

```
PUT /nelson.txt HTTP/1.1
Host: quotes.s3.amazonaws.com
Date: Mon, 15 Oct 2007 22:18:46 +0000

Content-Length: 6
Expect: 100-continue
```

O Amazon S3 retorna o seguinte:

```
HTTP/1.1 307 Temporary Redirect
Location: http://quotes.s3-4c25d83b.amazonaws.com/nelson.txt?rk=8d47490b
Content-Type: application/xml
Transfer-Encoding: chunked
Date: Mon, 15 Oct 2007 22:18:46 GMT

Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<Error>
<Code>TemporaryRedirect</Code>
<Message>Please re-send this request to the
specified temporary endpoint. Continue to use the
original request endpoint for future requests.
</Message>
<Endpoint>quotes.s3-4c25d83b.amazonaws.com</Endpoint>
<Bucket>quotes</Bucket>
</Error>
```

O cliente segue a resposta de redirecionamento e emite uma nova solicitação ao endpoint temporário `quotes.s3-4c25d83b.amazonaws.com`.

```
PUT /nelson.txt?rk=8d47490b HTTP/1.1
```

```
Host: quotes.s3-4c25d83b.amazonaws.com
Date: Mon, 15 Oct 2007 22:18:46 +0000

Content-Length: 6
Expect: 100-continue
```

O Amazon S3 retorna um 100-continue indicando que o cliente deve continuar com o envio de corpo da solicitação.

```
HTTP/1.1 100 Continue
```

O cliente envia o corpo da solicitação.

```
ha ha\n
```

O Amazon S3 retorna a resposta final.

```
HTTP/1.1 200 OK
Date: Mon, 15 Oct 2007 22:18:48 GMT

ETag: "a2c8d6b872054293af41061e93bc289"
Content-Length: 0
Server: AmazonS3
```

# Trabalho com buckets do Amazon S3

O Amazon S3 é o armazenamento na nuvem para a Internet. Para fazer upload dos dados (fotos, vídeos, documentos etc.), você primeiro cria um bucket em uma das regiões da AWS. Você pode fazer upload de um número ilimitado de objetos para o bucket.

Em termos de implementação, os buckets e objetos são recursos, e o Amazon S3 fornece APIs para você gerenciá-los. Por exemplo, você pode criar um bucket e fazer upload de objetos usando a API do Amazon S3. Você também pode usar o console do Amazon S3 para executar essas operações. O console usa as APIs do Amazon S3 para enviar solicitações para o Amazon S3.

Esta seção explica como trabalhar com buckets. Para obter mais informações sobre como trabalhar com objetos, consulte [Trabalho com objetos do Amazon S3 \(p. 101\)](#).

Um nome de bucket do Amazon S3 é globalmente exclusivo, e o namespace é compartilhado por todas as contas da AWS. Isso significa que, após a criação de um bucket, o nome dele não pode ser usado por outra conta da AWS em nenhuma região da AWS até que ele seja excluído. Você não pode depender de convenções de nomenclatura de buckets específicos para fins de disponibilidade ou verificação de segurança. Para ver as diretrizes de nomeação de bucket, consulte [Restrições e limitações do bucket \(p. 59\)](#).

O Amazon S3 cria buckets na região que você especifica. Para otimizar a latência, minimizar os custos ou atender a requisitos regulatórios, você pode escolher qualquer região da AWS geograficamente próxima para otimizar a latência. Por exemplo, se você residir na Europa, poderá considerar vantajoso criar buckets nas regiões UE (Irlanda) ou UE (Frankfurt). Para obter uma lista de regiões do Amazon S3, consulte [Regiões e endpoints](#) na Referência geral da AWS.

## Note

Os objetos que pertencem a um bucket criado em uma região da AWS específica jamais saem dela, a menos que você os transfira explicitamente para outra região. Por exemplo, os objetos armazenados na região UE (Irlanda) nunca saem dela.

## Tópicos

- [Criação de um bucket \(p. 54\)](#)
- [Acesso a um bucket \(p. 56\)](#)
- [Opções de configuração de bucket \(p. 57\)](#)
- [Restrições e limitações do bucket \(p. 59\)](#)
- [Exemplos de criação de um bucket \(p. 60\)](#)
- [Exclusão ou esvaziamento do bucket \(p. 63\)](#)
- [Criptografia padrão do Amazon S3 para buckets do S3 \(p. 68\)](#)
- [Gerenciamento de configuração de website de bucket \(p. 71\)](#)
- [Amazon S3 Transfer Acceleration \(p. 75\)](#)
- [Buckets de Pagamento pelo solicitante \(p. 83\)](#)
- [Controle de acesso e buckets \(p. 86\)](#)
- [Relatórios de uso e faturamento dos buckets do S3 \(p. 86\)](#)

## Criação de um bucket

O Amazon S3 fornece APIs para criar e gerenciar buckets. Por padrão, você pode criar até 100 buckets em cada conta da AWS. Se você precisar de mais buckets, poderá aumentar o limite do bucket enviando um

aumento de limite de serviço. Para saber como enviar um aumento de limite de bucket, consulte [Limites de serviço da AWS](#) na Referência geral da AWS.

Quando você cria um bucket, fornece um nome e a região da AWS onde deseja criar o bucket. Para obter informações sobre nomeação de buckets, consulte [Regras para nomeação de bucket \(p. 59\)](#).

Você pode armazenar qualquer número de objetos no bucket.

Você pode criar um bucket usando qualquer um dos seguintes métodos:

- Com o console.
- Usar os AWS SDKs de maneira programática.

Note

Se necessário, você também pode chamar a API REST do Amazon S3 diretamente do seu código. Contudo, isso pode ser complicado, porque exige que você grave código para autenticar suas solicitações. Para obter mais informações, consulte [Bucket PUT](#) no Amazon Simple Storage Service API Reference.

Ao usar os SDKs da AWS, você primeiro cria um cliente e então usa o cliente para enviar uma solicitação para criar um bucket. Ao criar o cliente, você pode especificar uma região da AWS. Leste dos EUA (Norte da Virgínia) é a região padrão. Observe o seguinte:

- Se você criar um cliente especificando a região Leste dos EUA (Norte da Virgínia), o cliente usará o seguinte endpoint para se comunicar com o Amazon S3:

s3.amazonaws.com

Você pode usar esse cliente para criar um bucket em qualquer região da AWS. Na sua solicitação de criação de bucket:

- Se você não especificar uma região, o Amazon S3 criará o bucket na região Leste dos EUA (Norte da Virgínia).
- Se você especificar uma região da AWS, o Amazon S3 criará o bucket na região especificada.
- Se você criar um cliente especificando qualquer outra região da AWS, cada uma dessas regiões será mapeada para o endpoint específico da região:

s3-<region>.amazonaws.com

Por exemplo, se você criar um cliente especificando a região eu-west-1, ele será mapeado para o seguinte endpoint específico da região:

s3-eu-west-1.amazonaws.com

Nesse caso, você pode usar o cliente para criar um bucket somente na região eu-west-1. O Amazon S3 retornará um erro se você especificar qualquer outra região em sua solicitação para criar um bucket.

- Se você criar um cliente para acessar um endpoint de pilha dupla, deverá especificar uma região da AWS. Para obter mais informações, consulte [Endpoints de pilha dupla do \(p. 15\)](#).

Para obter uma lista das regiões da AWS disponíveis, consulte [Regiões e endpoints](#) no AWS General Reference.

---

Para ver exemplos, consulte [Exemplos de criação de um bucket \(p. 60\)](#).

## Sobre permissões

Você pode usar suas credenciais raiz da conta da AWS para criar um bucket e executar qualquer outra operação do Amazon S3. Contudo, a AWS recomenda que você não use as credenciais raiz da sua conta da AWS para fazer solicitações tais como a criação de um bucket. Em vez disso, crie um usuário do IAM e conceda a esse usuário acesso total (por padrão, os usuários não têm nenhuma permissão). Esses usuários são conhecidos como usuários administradores. As credenciais do usuário administrador podem ser usadas em vez das credenciais raiz da conta para interagir com a AWS e executar tarefas, tais como criar um bucket, criar usuários e conceder permissões a eles.

Para obter mais informações, consulte [Credenciais de conta raiz vs. credenciais do usuário do IAM](#) na Referência geral da AWS e nas [Melhores práticas do IAM](#) no Guia do usuário do IAM.

A conta da AWS que cria um recurso é proprietária daquele recurso. Por exemplo, se você criar um usuário do IAM na sua conta da AWS e conceder permissões para esse usuário criar um bucket, o usuário poderá criar um bucket. Mas o usuário não é proprietário do bucket; a conta da AWS à qual o usuário pertence é proprietária do bucket. O usuário precisará de permissão adicional do proprietário do recurso para executar qualquer outra operação de bucket. Para obter mais informações sobre o gerenciamento de permissões para recursos do Amazon S3, consulte [Gerenciamento de permissões de acesso aos recursos do Amazon S3 \(p. 282\)](#).

## Acesso a um bucket

Você pode acessar seu bucket usando o console do Amazon S3. Usando a interface do console, você pode executar quase todas as operações de bucket sem ter que gravar nenhum código.

Se você acessar um bucket por programação, observe que o Amazon S3 oferece suporte à arquitetura RESTful na qual seus buckets e objetos são recursos, cada um com um URI de recurso que identifica exclusivamente o recurso.

O Amazon S3 oferece suporte aos URLs de estilo hospedado virtual e estilo de caminho para acessar um bucket.

- Em um URL de estilo hospedado virtual, o nome do bucket faz parte do nome do domínio no URL. Por exemplo:
  - `http://bucket.s3.amazonaws.com`
  - `http://bucket.s3-aws-region.amazonaws.com`

Em um URL de estilo hospedado virtual, você pode usar qualquer um desses endpoints. Se você fizer uma solicitação ao endpoint `http://bucket.s3.amazonaws.com`, o DNS terá informações suficientes para direcionar sua solicitação diretamente à região onde seu bucket reside.

Para obter mais informações, consulte [Hospedagem virtual de buckets \(p. 46\)](#).

- Em um URL de estilo de caminho, o nome do bucket não faz parte do domínio (a menos que você use um endpoint específico da região). Por exemplo:
  - Endpoint da região Leste dos EUA (Norte da Virgínia), `http://s3.amazonaws.com/bucket`
  - Endpoint específico da região, `http://s3-aws-region.amazonaws.com/bucket`

Em um URL de estilo de caminho, o endpoint que você usa deve corresponder à região na qual o bucket reside. Por exemplo, se seu bucket estiver na região América do Sul (São Paulo), você deverá usar o endpoint `http://s3-sa-east-1.amazonaws.com/bucket`. Se seu bucket estiver na região Leste dos EUA (Norte da Virgínia), você deverá usar o endpoint `http://s3.amazonaws.com/bucket`.

### Important

Como os buckets podem ser acessados usando URLs de estilo hospedado virtual e estilo de caminho, recomendamos que você crie buckets com nomes compatíveis com DNS. Para obter mais informações, consulte [Restrições e limitações do bucket \(p. 59\)](#).

### Acesso a um bucket do S3 via IPv6

O Amazon S3 tem um conjunto de endpoints de pilha dupla, que oferece suporte a solicitações para buckets do S3 via Protocolo de Internet versão 6 (IPv6) e IPv4. Para obter mais informações, consulte [Fazer solicitações por meio do IPv6 \(p. 12\)](#).

## Opções de configuração de bucket

O Amazon S3 oferece suporte a várias opções para que você configure seu bucket. Por exemplo, você pode configurar seu bucket para hospedagem de sites, adicionar configuração para gerenciar o ciclo de vida dos objetos no bucket e configurar o bucket para registrar todo o acesso a ele. O Amazon S3 oferece suporte a sub-recursos para armazenar e gerenciar as informações de configuração do bucket. Isto é, usando a API do Amazon S3, você pode criar e gerenciar esses sub-recursos. Você também pode usar o console ou os SDKs da AWS.

### Note

Há também configurações no nível do objeto. Por exemplo, você pode configurar permissões no nível do objeto configurando uma lista de controle de acesso (ACL) específica para aquele objeto.

São chamados de sub-recursos porque existem no contexto de um bucket ou objeto específico. A tabela a seguir lista os sub-recursos que permitem gerenciar configurações específicas de bucket.

Sub-recurso	Descrição
location	Ao criar um bucket, especifique a região da AWS onde deseja que o Amazon S3 crie o bucket. O Amazon S3 armazena essas informações no sub-recurso do local e fornece uma API para recuperar essas informações.
política e ACL (lista de controle de acesso)	Todos os seus recursos (como buckets e objetos) são privados por padrão. O Amazon S3 oferece suporte às opções de política de bucket e lista de controle de acesso (ACL) para conceder e gerenciar permissões no nível do bucket. O Amazon S3 armazena as informações de permissão nos sub-recursos política e acl.  Para obter mais informações, consulte <a href="#">Gerenciamento de permissões de acesso aos recursos do Amazon S3 (p. 282)</a> .
cors (compartilhamento de recurso de origem cruzada)	Você pode configurar seu bucket para autorizar solicitações de origem cruzada.  Para obter mais informações, consulte <a href="#">Habilitar compartilhamento de recursos de origem cruzada</a> .
website	É possível configurar seu bucket para hospedagem de sites estáticos. O Amazon S3 armazena essa configuração criando um sub-recurso website.  Para obter mais informações, consulte <a href="#">Hospedar um site estático no Amazon S3</a> .
registro em log	O registro em log permite que você rastreie solicitações de acesso ao seu bucket. Cada registro de log de acesso fornece detalhes sobre uma única solicitação de acesso, como solicitante, nome do bucket, horário da solicitação, ação da solicitação, status de resposta e código de erro, se houver. As informações de log

Sub-recurso	Descrição
	<p>de acesso podem ser úteis em auditorias de segurança e acesso. Isso também pode ajudar a conhecer sua base de clientes e entender sua fatura do Amazon S3.</p> <p>Para obter mais informações, consulte <a href="#">Registro em log de acesso ao servidor Amazon S3 (p. 625)</a>.</p>
notificação de evento	<p>Você pode permitir que seu bucket envie notificações de eventos do bucket especificado.</p> <p>Para obter mais informações, consulte <a href="#">Configurar notificações de evento do Amazon S3 (p. 522)</a>.</p>
versionamento	<p>O versionamento ajuda a recuperar substituições e exclusões acidentais.</p> <p>Recomendamos o versionamento como melhor prática para impedir a exclusão ou substituição de objetos por engano.</p> <p>Para obter mais informações, consulte <a href="#">Usar versionamento (p. 448)</a>.</p>
ciclo de vida	<p>Você pode definir regras de ciclo de vida para objetos em seu bucket que têm um ciclo de vida bem definido. Por exemplo, você pode definir uma regra para arquivar objetos um ano após a criação ou excluir um objeto 10 anos após a criação.</p> <p>Para obter mais informações, consulte <a href="#">Gerenciamento de ciclo de vida de objetos</a>.</p>
replicação entre regiões	<p>A replicação entre regiões é a cópia assíncrona automática de objetos em buckets, em diferentes regiões da AWS. Para obter mais informações, consulte <a href="#">Replicação entre regiões (p. 544)</a>.</p>
marcação	<p>Você pode adicionar tags de alocação de custos ao seu bucket para categorizar e monitorar seus custos da AWS. O Amazon S3 fornece o sub-recurso marcação para armazenar e gerenciar tags em um bucket. Com o uso de tags em seu bucket, a AWS gera um relatório de alocação de custos com o uso e custos agregados por suas tags.</p> <p>Para obter mais informações, consulte <a href="#">Relatórios de uso e faturamento dos buckets do S3 (p. 86)</a>.</p>
requestPayment	<p>Por padrão, a conta da AWS que cria o bucket (o proprietário do bucket) paga pelos downloads do bucket. Usando esse sub-recurso, o proprietário do bucket pode especificar que a pessoa que solicita o download será cobrada pelo download. O Amazon S3 fornece uma API para gerenciar esse sub-recurso.</p> <p>Para obter mais informações, consulte <a href="#">Buckets de Pagamento pelo solicitante (p. 83)</a>.</p>
aceleração de transferência	<p>O Transfer Acceleration possibilita transferências de arquivos rápidas, fáceis e seguras em longas distâncias entre o seu cliente e um bucket do S3. O Transfer Acceleration aproveita os pontos de presença distribuídos globalmente do Amazon CloudFront.</p> <p>Para obter mais informações, consulte <a href="#">Amazon S3 Transfer Acceleration (p. 75)</a>.</p>

## Restrições e limitações do bucket

Um bucket é de propriedade da conta da AWS que o criou. Por padrão, você pode criar até 100 buckets em cada conta da AWS. Se você precisar de buckets adicionais, poderá aumentar o limite do bucket enviando um aumento de limite de serviço. Para obter informações sobre como aumentar o limite do seu bucket, acesse [Limites de serviço da AWS](#) na Referência geral da AWS.

A propriedade do bucket não é transferível; contudo, se um bucket está vazio, você pode excluí-lo. Depois que um bucket é excluído, o nome se torna disponível para reutilização, mas, por várias razões, o nome pode não estar disponível para reutilização. Por exemplo, alguma outra conta pode criar um bucket com aquele nome. Observe também que pode levar algum tempo para que o nome possa ser reutilizado. Se você deseja continuar a usar o mesmo nome de bucket, não exclua o bucket.

Não há limite para o número de objetos que podem ser armazenados em um bucket, nem diferença de desempenho se você usa muitos buckets ou apenas alguns. Você pode armazenar todos os objetos em um único bucket, ou pode organizá-los em vários buckets.

Depois de criar um bucket, você não pode alterar sua região.

Se você especificar explicitamente uma região da AWS na solicitação de criação do bucket que é diferente da região especificada ao criar o cliente, poderá obter um erro.

Você não pode criar um bucket em outro bucket.

A engenharia de alta disponibilidade do Amazon S3 é focada nas operações get, put, list e delete. Como as operações de bucket funcionam em um espaço de recurso centralizado e global, não é apropriado criar ou excluir buckets no caminho de código de alta disponibilidade do seu aplicativo. É melhor criar ou excluir buckets em uma rotina de inicialização ou configuração separada que você executa com menor frequência.

### Note

Se o seu aplicativo cria buckets automaticamente, escolha um esquema de nomeação de bucket que não seja suscetível a causar conflitos de nomeação. Certifique-se de que a lógica do seu aplicativo escolha um nome de bucket diferente, caso um nome de bucket já esteja em uso.

## Regras para nomeação de bucket

Após criar um bucket do S3, você não pode alterar o nome do bucket. Portanto, escolha bem esse nome.

### Important

Em 1º de março de 2018, atualizamos nossas convenções de nomenclatura para buckets do S3 na região Leste dos EUA (Norte da Virgínia) para corresponder às convenções de nomenclatura que usamos em todas as outras regiões da AWS no mundo. O Amazon S3 não oferece mais suporte à criação de nomes de bucket que contêm letras maiúsculas e sublinhados. Essa alteração permite que cada bucket possa ser abordado usando o estilo virtual de endereçamento de host, como `https://myawsbucket.s3.amazonaws.com`. Recomendamos que você analise os processos existentes de criação de bucket para garantir que você está seguindo as convenções de nomenclatura compatíveis com DNS.

Veja a seguir as regras para nomear buckets do S3 em todas as regiões da AWS:

- Os nomes de bucket devem ser exclusivos para todos os nomes de bucket existentes no Amazon S3.
- Os nomes de bucket devem estar em conformidade com as convenções de nomenclatura do DNS.
- Os nomes de bucket devem ter no mínimo 3 e no máximo 63 caracteres de extensão.
- Os nomes de bucket não devem conter caracteres em letras maiúsculas ou sublinhadas.
- Os nomes de bucket devem começar com uma letra minúscula ou um número.

- Os nomes de bucket devem ser uma série de um ou mais rótulos. Os rótulos adjacentes são separados por um único ponto (.). Os nomes de bucket podem conter letras minúsculas, números e hífen. Cada rótulo deve começar e terminar com uma letra minúscula ou um número.
- Os nomes de bucket não devem ser formatados como um endereço IP (por exemplo, 192.168.5.4).
- Ao usar buckets hospedados virtualmente com Secure Sockets Layer (SSL), o certificado SSL curinga corresponde somente a buckets que não contenham pontos. Para contornar isso, use HTTP ou escreva a sua própria lógica de verificação do certificado. Recomendamos não usar pontos (".") em nomes de buckets ao usar buckets hospedados virtualmente.

## Nomes de bucket antigos que não estão em conformidade com DNS

A partir de 1º de março de 2018, atualizamos as convenções de nomenclatura para buckets do S3 na região Leste dos EUA (Norte da Virgínia) para exigir nomes compatíveis com DNS.

A região Leste dos EUA (Norte da Virgínia) permitia anteriormente padrões menos rigorosos para nomenclatura de bucket, o que pode ter resultado em um nome de bucket que não é compatível com DNS. Por exemplo, `MyAWSbucket` era um nome de bucket válido, mesmo que contivesse letras maiúsculas. Se você tentar acessar esse bucket usando uma solicitação de hospedagem virtual (`http://MyAWSbucket.s3.amazonaws.com/yourobject`), o URL levará ao bucket `myawsbucket` e não ao bucket `MyAWSbucket`. Em resposta, o Amazon S3 retornará um erro de "bucket não encontrado". Para obter mais informações sobre o acesso em estilo de hospedagem virtual ao seus buckets, consulte [Hospedagem virtual de buckets \(p. 46\)](#).

As regras antigas para nomes de bucket na região Leste dos EUA (Norte da Virgínia) permitiam que eles tivessem até 255 caracteres e qualquer combinação de caracteres maiúsculos e minúsculos, números, pontos (.), hífens (-) e sublinhados (\_).

O nome do bucket usado para o Amazon S3 Transfer Acceleration deve ser compatível com DNS e não deve conter pontos ("."). Para obter mais informações sobre aceleração de transferência, consulte [Amazon S3 Transfer Acceleration \(p. 75\)](#).

## Exemplos de criação de um bucket

### Tópicos

- [Usar o console do Amazon S3 \(p. 61\)](#)
- [Usar o AWS SDK for Java \(p. 61\)](#)
- [Usar o AWS SDK para .NET \(p. 62\)](#)
- [Uso do AWS SDK para Ruby Versão 3 \(p. 63\)](#)
- [Uso de outros AWS SDKs \(p. 63\)](#)

Os exemplos de código a seguir criam um bucket por programação usando os AWS SDKs para Java, .NET e Ruby. Os exemplos de código executam as seguintes tarefas:

- Criar um bucket, se ele não existir ainda — Os exemplos criam um bucket ao executar as seguintes tarefas:
  - Crie um cliente especificando explicitamente uma região da AWS (o exemplo usa a Região `s3-eu-west-1`). Conforme apropriado, o cliente comunica-se com o Amazon S3 usando o endpoint `s3-eu-west-1.amazonaws.com`. Você pode especificar qualquer outra região da AWS. Para obter uma lista das Regiões do AWS, consulte [Regiões e endpoints](#) na Referência geral da AWS.
  - Envie uma solicitação de criação de bucket especificando apenas um nome de bucket. A solicitação de criação de bucket não especifica outra Região da AWS. O cliente envia uma solicitação ao Amazon

S3 de criar o bucket na Região que você especificou ao criar o cliente. Assim que você criou um bucket, você não pode alterar sua Região.

Note

Se você especificar explicitamente uma região da AWS na solicitação de criação do bucket que é diferente da região especificada ao criar o cliente, poderá obter um erro. Para obter mais informações, consulte [Criação de um bucket \(p. 54\)](#).

As bibliotecas de SDK enviam ao Amazon S3 uma solicitação PUT para criar o bucket. Para obter mais informações, consulte [PUT Bucket](#).

- Recupere as informações sobre a localização do bucket — o Amazon S3 armazena as informações de localização de bucket no sub-recurso local que é associado ao bucket. As bibliotecas de SDK enviam uma solicitação GET Bucket location (consulte [GET localização do bucket](#)) para recuperar essas informações.

## Usar o console do Amazon S3

Para criar um bucket usando o console do Amazon S3 consulte [Como crio um bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

## Usar o AWS SDK for Java

### Example

Este exemplo mostra como criar um bucket do Amazon S3 usando o AWS SDK for Java. Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

import java.io.IOException;

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

public class CreateBucket {

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

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

            if (!s3Client.doesBucketExistV2(bucketName)) {
                // Because the CreateBucketRequest object doesn't specify a region, the
                // bucket is created in the region specified in the client.
                s3Client.createBucket(new CreateBucketRequest(bucketName));
            }
        } catch (AmazonServiceException | SdkClientException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

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

## Usar o AWS SDK para .NET

Para obter informações sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

### Example

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-
developer-guide/blob/master/LICENSE-SAMPLECODE.)

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

namespace Amazon.DocSamples.S3
{
    class CreateBucketTest
    {
        private const string bucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            CreateBucketAsync().Wait();
        }

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

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

## Uso do AWS SDK para Ruby Versão 3

Para obter informações sobre como criar e testar um exemplo funcional, consulte [Usar o AWS SDK para Ruby - versão 3 \(p. 650\)](#).

### Example

```
require 'aws-sdk-s3'

s3 = Aws::S3::Client.new(region: 'us-west-2')
s3.create_bucket(bucket: 'bucket-name')
```

## Uso de outros AWS SDKs

Para obter informações sobre como usar outros AWS SDKs, consulte [Código de exemplo e bibliotecas](#).

## Exclusão ou esvaziamento do bucket

É fácil excluir um bucket vazio. No entanto, em algumas situações, pode ser que você precise excluir ou esvaziar um bucket que contenha alguns objetos. Nesta seção, explicaremos como excluir objetos em um bucket sem versionamento, e também como excluir versões de objeto e marcadores de exclusão em um bucket com o versionamento habilitado. Para obter mais informações sobre versionamento, consulte [Usar versionamento \(p. 448\)](#). Em algumas situações, é possível esvaziar um bucket em vez de exclui-lo. Esta seção explica várias opções que você pode usar para excluir ou esvaziar um bucket que contenha objetos.

## Tópicos

- [Excluir um Bucket \(p. 64\)](#)
- [Esvaziar um bucket \(p. 66\)](#)

# Excluir um Bucket

Você pode excluir um bucket e seu conteúdo programaticamente usando os SDKs da AWS. Você também pode usar a configuração de ciclo de vida em um bucket para esvaziar o conteúdo e, em seguida, excluir o bucket. Há opções adicionais, como usar o console do Amazon S3 e a CLI da AWS, mas existem limitações nesses métodos com base no número de objetos em seu bucket e no status de versionamento do bucket.

## Excluir um bucket: usando o console do Amazon S3

O console do Amazon S3 oferece suporte à exclusão de um bucket que pode ou não estar vazio. Para obter informações sobre como usar o console do Amazon S3 para excluir um bucket, consulte [Como eu faço para excluir um bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

## Excluir um bucket: usando a AWS CLI

Você só pode excluir um bucket que contenha objetos usando a CLI da AWS se o bucket não tiver versionamento habilitado. Se seu bucket não tiver o versionamento habilitado, você poderá usar o comando `rb` (remove bucket) da CLI da AWS com o parâmetro `--force` para remover um bucket não vazio. Esse comando exclui todos os objetos primeiro e, em seguida, exclui o bucket.

```
$ aws s3 rb s3://bucket-name --force
```

Para obter informações, consulte [Usar comandos do S3 de alto nível com a interface da linha de comando da AWS](#) no Guia do usuário do AWS Command Line Interface.

Para excluir um bucket não vazio que não tenha o versionamento habilitado, você tem as seguintes opções:

- Excluir o bucket programaticamente usando o AWS SDK.
- Exclua todos os objetos usando a configuração de ciclo de vida do bucket e, em seguida, exclua o bucket vazio usando o console do Amazon S3.

## Excluir um bucket usando a configuração de ciclo de vida

Você pode configurar o ciclo de vida em seu bucket para que torne os objetos expirados, e o Amazon S3 então excluirá esses objetos. Você pode adicionar regras de configuração de ciclo de vida para tornar todos os objetos ou um subconjunto de objetos expirados com um prefixo de nome de chave específico. Por exemplo, para remover todos os objetos em um bucket, você pode definir uma regra de ciclo de vida para tornar os objetos expirados um dia após a criação.

Se seu bucket tem versionamento habilitado, você também pode configurar a regra para tornar versões desatualizadas de objetos expiradas.

Após o Amazon S3 excluir todos os objetos em seu bucket, você pode excluir o bucket ou mantê-lo.

### Important

Se você deseja somente esvaziar o bucket e não excluí-lo, certifique-se de remover a regra de configuração de ciclo de vida que adicionou para esvaziar o bucket para que os objetos novos que você criar no bucket permaneçam no bucket.

Para obter mais informações, consulte [Gerenciamento do ciclo de vida de objetos \(p. 122\)](#) e [Configurar a expiração de objeto \(p. 129\)](#).

## Excluir um bucket: usando AWS SDKs

Você pode usar AWS SDKs para excluir um bucket: As seções a seguir fornecem exemplos de como excluir um bucket usando AWS SDK para Java e .NET. Primeiro, o código exclui os objetos no bucket e, em seguida, exclui o bucket. Para obter mais informações sobre outros AWS SDKs, consulte [Ferramentas para o Amazon Web Services](#).

### Excluir um bucket usando o AWS SDK for Java

O exemplo de Java a seguir exclui um bucket que contém objetos. O exemplo exclui todos os objetos e, em seguida, exclui o bucket. O exemplo também funciona para buckets com ou sem versionamento habilitado.

#### Note

Para buckets sem versionamento habilitado, você pode excluir todos os objetos diretamente e, em seguida, excluir o bucket. Para buckets com versionamento habilitado, você deve excluir todas as versões do objeto antes de excluir o bucket.

Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

import java.util.Iterator;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ListVersionsRequest;
import com.amazonaws.services.s3.model.ObjectListing;
import com.amazonaws.services.s3.model.S3ObjectSummary;
import com.amazonaws.services.s3.model.S3VersionSummary;
import com.amazonaws.services.s3.model.VersionListing;

public class DeleteBucket {

    public static void main(String[] args) {
        String clientRegion = "**** Client region ****";
        String bucketName = "**** Bucket name ****";

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

            // Delete all objects from the bucket. This is sufficient
            // for unversioned buckets. For versioned buckets, when you attempt to delete
            objects, Amazon S3 inserts
            // delete markers for all objects, but doesn't delete the object versions.
            // To delete objects from versioned buckets, delete all of the object versions
            before deleting
            // the bucket (see below for an example).
            ObjectListing objectListing = s3Client.listObjects(bucketName);
        }
    }
}
```

```
        while (true) {
            Iterator<S3ObjectSummary> objIter =
objectListing.getObjectSummaries().iterator();
            while (objIter.hasNext()) {
                s3Client.deleteObject(bucketName, objIter.next().getKey());
            }

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

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

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

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

## Esvaziar um bucket

Você pode esvaziar o conteúdo de um bucket (ou seja, excluir todo conteúdo, mas manter o bucket) programaticamente usando o AWS SDK. Você também pode especificar a configuração de ciclo de vida em um bucket para expirar objetos para que o Amazon S3 possa excluí-los. Há opções adicionais, como usar o console do Amazon S3 e a CLI da AWS, mas existem limitações nesse método com base no número de objetos em seu bucket e no status de versionamento do bucket.

### Tópicos

- [Esvaziar um bucket: usando o console do Amazon S3 \(p. 67\)](#)

- [Esvaziar um bucket: usando a AWS CLI \(p. 67\)](#)
- [Esvaziar um bucket: usando a configuração de ciclo de vida \(p. 67\)](#)
- [Esvaziar um bucket usando AWS SDKs \(p. 68\)](#)

## Esvaziar um bucket: usando o console do Amazon S3

Para obter informações sobre como usar o console do Amazon S3 para esvaziar um bucket, consulte [Como eu faço para esvaziar um bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

### Esvaziar um bucket: usando a AWS CLI

Você só pode esvaziar um bucket usando a CLI da AWS se o bucket não tiver versionamento habilitado. Se seu bucket não tem o versionamento habilitado, você pode usar o comando `rm` (remove) da CLI da AWS com o parâmetro `--recursive` para esvaziar um bucket (ou remover um subconjunto de objetos com um prefixo de nome de chave específico).

O comando `rm` a seguir remove os objetos com o prefixo de nome de chave `doc`, por exemplo, `doc/doc1` e `doc/doc2`.

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

Use o comando a seguir para remover todos os objetos sem especificar um prefixo.

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

Para obter informações, consulte [Usar comandos do S3 de alto nível com a interface da linha de comando da AWS](#) no Guia do usuário do AWS Command Line Interface.

#### Note

Você não pode remover objetos de um bucket com o versionamento habilitado. O Amazon S3 adiciona um marcador de exclusão quando você exclui um objeto, que é o que este comando fará. Para obter mais informações sobre versionamento, consulte [Usar versionamento \(p. 448\)](#).

Para esvaziar um bucket com versionamento habilitado, você tem as seguintes opções:

- Excluir o bucket programaticamente usando o AWS SDK.
- Usar a configuração de ciclo de vida do bucket para solicitar que o Amazon S3 exclua objetos.
- Use o console do Amazon S3 (consulte [Como eu esvazio um bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service)

## Esvaziar um bucket: usando a configuração de ciclo de vida

Você pode configurar o ciclo de vida no bucket para expirar objetos e solicitar que o Amazon S3 exclua esses objetos. Você pode adicionar regras de configuração de ciclo de vida para tornar todos os objetos ou um subconjunto de objetos expirados com um prefixo de nome de chave específico. Por exemplo, para remover todos os objetos em um bucket, você pode definir uma regra de ciclo de vida para tornar objetos expirados um dia após a criação.

Se seu bucket tem versionamento habilitado, você também pode configurar a regra para tornar versões desatualizadas de objetos expiradas.

### Warning

Após seus objetos expirarem, o Amazon S3 os exclui. Se você deseja somente esvaziar o bucket e não excluí-lo, certifique-se de remover a regra de configuração de ciclo de vida que adicionou para esvaziar o bucket para que os objetos novos que você criar no bucket permaneçam no bucket.

Para obter mais informações, consulte [Gerenciamento do ciclo de vida de objetos \(p. 122\)](#) e [Configurar a expiração de objeto \(p. 129\)](#).

## Esvaziar um bucket usando AWS SDKs

Você pode usar os AWS SDKs para esvaziar um bucket ou para remover um subconjunto de objetos com um prefixo de nome de chave específico.

Para obter um exemplo de como esvaziar um bucket usando o AWS SDK for Java, consulte [Excluir um bucket usando o AWS SDK for Java \(p. 65\)](#). O código exclui todos os objetos, independentemente de o bucket ter versionamento habilitado ou não e, em seguida, exclui o bucket. Para só esvaziar o bucket, certifique-se de remover o comando que exclui o bucket.

Para obter mais informações sobre como usar outros AWS SDKs, consulte [Ferramentas para o Amazon Web Services](#).

# Criptografia padrão do Amazon S3 para buckets do S3

A criptografia padrão do Amazon S3 fornece uma forma de configurar o comportamento de criptografia padrão para um bucket do S3. Você pode configurar a criptografia padrão em um bucket para que todos os objetos sejam criptografados quando forem armazenados nele. Os objetos são criptografados usando a criptografia do lado do servidor com as chaves gerenciadas pelo Amazon S3 (SSE-S3) ou as chaves gerenciadas pelo AWS KMS (SSE-KMS).

Quando você usa a criptografia do lado do servidor, o Amazon S3 criptografa um objeto antes de salvá-lo no disco em seus respectivos datacenters e o descriptografa quando você faz download dele. Para obter mais informações sobre como proteger dados usando a criptografia do lado do servidor e o gerenciamento de chaves de criptografia, consulte [Proteção de dados usando criptografia \(p. 409\)](#).

A criptografia padrão funciona com todos os buckets do S3, existentes e novos. Para criptografar todos os objetos armazenados em um bucket sem a criptografia padrão, você precisa incluir informações de criptografia com cada solicitação de armazenamento de objeto. Você também precisa configurar uma política de bucket do S3 para rejeitar solicitações de armazenamento que não incluem informações de criptografia.

Não há novas cobranças para usar a criptografia padrão para buckets do S3. As solicitações de configuração do recurso de criptografia padrão geram cobranças padrão de solicitação do Amazon S3. Para obter mais informações sobre definição de preços, consulte [Definição de preço do Amazon S3](#). Para o armazenamento de chaves de criptografia SSE-KMS, aplicam-se taxas do AWS Key Management Service. Elas estão listadas em [Definição de preços do AWS KMS](#).

### Tópicos

- [Como configurar a criptografia padrão do Amazon S3 em um bucket do S3? \(p. 69\)](#)
- [Mudar das políticas de bucket para a criptografia padrão para realização de criptografia \(p. 69\)](#)
- [Usar a criptografia padrão com a replicação entre regiões \(p. 69\)](#)

- [Monitoramento da criptografia padrão com o CloudTrail e o CloudWatch](#) (p. 70)
- [Mais informações](#) (p. 71)

## Como configurar a criptografia padrão do Amazon S3 em um bucket do S3?

Esta seção descreve como configurar a criptografia padrão do Amazon S3. Você pode usar os SDKs da AWS, a API REST do Amazon S3, a AWS Command Line Interface (AWS CLI), ou o console do Amazon S3 para habilitar a criptografia padrão. A maneira mais fácil de configurar a criptografia padrão para um bucket do S3 é usar o Console de gerenciamento da AWS.

Você pode configurar a criptografia padrão em um bucket usando qualquer uma das seguintes maneiras:

- Use o console do Amazon S3. Para obter mais informações, consulte [Como habilitar a criptografia padrão para um bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.
- Use as seguintes APIs REST:
  - Use a operação da API REST [PUT Bucket encryption](#) para habilitar a criptografia padrão e configurar o tipo de criptografia do lado do servidor de modo a usar —SSE-S3 ou SSE-KMS.
  - Use a API REST [DELETE Bucket encryption](#) para desabilitar a criptografia padrão de objetos. Depois de desabilitar a criptografia padrão, o Amazon S3 criptografará objetos somente se as solicitações `PUT` incluírem informações de criptografia. Para obter mais informações, consulte [PUT Object](#) e [PUT Object – Cópia](#).
  - Use a API REST [GET Bucket encryption](#) para verificar a configuração da criptografia padrão atual.
- Use a AWS CLI e os SDKs da AWS. Para obter mais informações, consulte [Usar os AWS SDKs, a CLI e os Explorers](#) (p. 639).

Depois de habilitar a criptografia padrão para um bucket, o seguinte comportamento de criptografia será aplicado:

- Não há alteração na criptografia dos objetos que existiam no bucket antes da ativação da criptografia padrão.
- Quando você faz upload de objetos após a ativação da criptografia padrão:
  - Se seus cabeçalhos de solicitação `PUT` não incluírem informações de criptografia, o Amazon S3 usará as configurações de criptografia padrão do bucket para criptografar os objetos.
  - Se seus cabeçalhos de solicitação `PUT` incluírem informações de criptografia, o Amazon S3 usará as informações de criptografia da solicitação `PUT` para criptografar objetos antes de armazená-los no Amazon S3. Se `PUT` for concluído com sucesso, a resposta será `HTTP/1.1 200 OK` com as informações de criptografia nos cabeçalhos de resposta. Para obter mais informações, consulte [Objeto PUT](#).
- Se você usar a opção SSE-KMS na sua configuração de criptografia padrão, estará sujeito aos limites de RPS (solicitações por segundo) do AWS KMS. Para obter mais informações sobre os limites do AWS KMS e sobre como solicitar um aumento de limite, consulte [Limites do AWS KMS](#).

## Mudar das políticas de bucket para a criptografia padrão para realização de criptografia

Se você atualmente aplica a criptografia de objeto em um bucket do S3 usando uma política de bucket para rejeitar solicitações `PUT` sem cabeçalhos de criptografia, recomendamos que use o procedimento a seguir para começar a usar a criptografia padrão.

Para deixar de usar a política de bucket para rejeitar solicitações `PUT` sem cabeçalhos de criptografia e passar a usar criptografia padrão

1. Se você planeja especificar que a criptografia padrão utilizará SSE-KMS, verifique se todas as solicitações de objeto `PUT` e `GET` foram assinadas usando o Signature Version 4 e enviadas ao Amazon S3 por meio de uma conexão SSL. Para informações sobre como usar o AWS KMS, consulte [Proteção de dados usando criptografia do servidor com chaves gerenciadas pelo AWS KMS– \(SSE-KMS\) \(p. 410\)](#).

Note

Por padrão, o console do Amazon S3, a AWS CLI versão 1.11.108 e posterior e todos os SDKs da AWS lançados depois de maio de 2016 utilizam as solicitações assinadas do Signature Version 4 enviadas ao Amazon S3 por meio de uma conexão SSL.

2. Exclua as instruções da política de bucket que rejeitam solicitações `PUT` sem cabeçalhos de criptografia. (Recomendamos que você salve uma cópia de backup da política de bucket que está sendo substituída.)
3. Para garantir que o comportamento de criptografia esteja configurado como você deseja, teste várias solicitações `PUT` para simular com mais precisão sua carga de trabalho real.
4. Se você estiver usando a criptografia padrão com SSE-KMS, monitore seus clientes para identificar solicitações `PUT` e `GET` que apresentaram falhas depois das mudanças. Provavelmente, essas são as solicitações que você não atualizou de acordo com a Etapa 1. Altere as solicitações `PUT` ou `GET` com falha para serem assinadas com o Signature Version 4 da AWS e enviadas por meio de SSL.

Depois de habilitar a criptografia padrão para o seu bucket do S3, os objetos armazenados no Amazon S3 por meio de quaisquer solicitações `PUT` sem cabeçalhos de criptografia serão criptografados usando as configurações de criptografia padrão do nível de bucket.

## Usar a criptografia padrão com a replicação entre regiões

Depois de habilitar a criptografia padrão para um bucket de destino de replicação entre regiões, o seguinte comportamento de criptografia será aplicado:

- Se os objetos no bucket de origem não estiverem criptografados, os objetos de réplica no bucket de destino serão criptografados usando as configurações de criptografia padrão do bucket de destino. Isso faz com que a ETag do objeto de origem seja diferente da ETag do objeto de réplica. Você precisa atualizar os aplicativos que usam a ETag para acomodar essa diferença.
- Se os objetos no bucket de origem forem criptografados usando SSE-S3 ou SSE-KMS, os objetos de réplica no bucket de destino usarão a mesma criptografia que a criptografia do objeto de origem. As configurações de criptografia padrão do bucket de destino não são usadas.

## Monitoramento da criptografia padrão com o CloudTrail e o CloudWatch

Você pode acompanhar as solicitações de configuração de criptografia padrão por meio de eventos do AWS CloudTrail. Os nomes de eventos da API usados nos logs do CloudTrail são `PutBucketEncryption`, `GetBucketEncryption` e `DeleteBucketEncryption`. Você também pode criar o Eventos do Amazon CloudWatch com operações de nível de bucket do S3 como o tipo de evento. Para obter mais informações sobre os eventos do CloudTrail, consulte [Como habilitar o registro no nível do objeto para um bucket do S3 com eventos de dados do CloudWatch?](#)

Você pode usar os logs do CloudTrail para ações de nível de objeto do Amazon S3 para acompanhar solicitações PUT e POST ao Amazon S3 e verificar se a criptografia padrão está sendo usada para criptografar objetos quando as solicitações PUT recebidas não contiverem cabeçalhos de criptografia.

Quando o Amazon S3 criptografa um objeto usando as configurações de criptografia padrão, o log inclui o seguinte campo como o par de nome/valor: "SSEApplied": "Default\_SSE\_S3" or "SSEApplied": "Default\_SSE\_KMS".

Quando o Amazon S3 criptografa um objeto usando os cabeçalhos de criptografia PUT, o log inclui o seguinte campo como par de nome/valor: "SSEApplied": "SSE\_S3", "SSEApplied": "SSE\_KMS" ou "SSEApplied": "SSE\_C". Para multipart uploads, essas informações estão incluídas nas solicitações de API `InitiateMultipartUpload`. Para obter mais informações sobre o uso do CloudTrail e do CloudWatch, consulte [Monitoramento do Amazon S3 \(p. 597\)](#).

## Mais informações

- [PUT Bucket encryption](#)
- [DELETE Bucket encryption](#)
- [GET Bucket encryption](#)

# Gerenciamento de configuração de website de bucket

## Tópicos

- [Gerenciamento de sites com o Console de gerenciamento da AWS \(p. 71\)](#)
- [Gerenciamento de sites com o AWS SDK for Java \(p. 71\)](#)
- [Gerenciamento de sites com o AWS SDK para .NET \(p. 73\)](#)
- [Gerenciamento de sites com o AWS SDK para PHP \(p. 74\)](#)
- [Gerenciamento de sites com a API REST \(p. 75\)](#)

Você pode hospedar sites estáticos no Amazon S3 configurando seu bucket para hospedagem de sites. Para obter mais informações, consulte [Hospedagem de um site estático no Amazon S3 \(p. 494\)](#). Há várias formas de gerenciar a configuração de site do bucket. Você pode usar o Console de gerenciamento da AWS para gerenciar a configuração sem gravar nenhum código. Você pode você criar, atualizar e excluir, de maneira programática, a configuração de site usando os AWS SDKs. Os SDKs fornecem classes de wrapper na API REST do Amazon S3. Se seu aplicativo exigir, você pode enviar solicitações de API REST diretamente do seu aplicativo.

## Gerenciamento de sites com o Console de gerenciamento da AWS

Para obter mais informações, consulte [Configuração de bucket para hospedagem de site \(p. 496\)](#).

## Gerenciamento de sites com o AWS SDK for Java

O exemplo a seguir mostra como usar o AWS SDK for Java para gerenciar a configuração de site para um bucket. Para adicionar uma configuração de site a um bucket, forneça um nome de bucket e uma configuração de site. A configuração de site deve incluir um documento de índice e pode incluir um

documento de erro opcional. Esses documentos já devem existir no bucket. Para obter mais informações, consulte [PUT em site de bucket](#). Para obter mais informações sobre o recurso de site do Amazon S3, consulte [Hospedagem de um site estático no Amazon S3 \(p. 494\)](#).

### Example

O exemplo a seguir usa o AWS SDK for Java para adicionar uma configuração de site a um bucket, recuperar e imprimir a configuração e, em seguida, excluir a configuração e verificar a exclusão. Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.  
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-  
developer-guide/blob/master/LICENSE-SAMPLECODE.)  
  
import java.io.IOException;  
  
import com.amazonaws.AmazonServiceException;  
import com.amazonaws.SdkClientException;  
import com.amazonaws.auth.profile.ProfileCredentialsProvider;  
import com.amazonaws.services.s3.AmazonS3;  
import com.amazonaws.services.s3.AmazonS3ClientBuilder;  
import com.amazonaws.services.s3.model.BucketWebsiteConfiguration;  
  
public class WebsiteConfiguration {  
  
    public static void main(String[] args) throws IOException {  
        String clientRegion = "**** Client region ****";  
        String bucketName = "**** Bucket name ****";  
        String indexDocName = "**** Index document name ****";  
        String errorDocName = "**** Error document name ****";  
  
        try {  
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()  
                .withRegion(clientRegion)  
                .withCredentials(new ProfileCredentialsProvider())  
                .build();  
  
            // Print the existing website configuration, if it exists.  
            printWebsiteConfig(s3Client, bucketName);  
  
            // Set the new website configuration.  
            s3Client.setBucketWebsiteConfiguration(bucketName, new  
                BucketWebsiteConfiguration(indexDocName, errorDocName));  
  
            // Verify that the configuration was set properly by printing it.  
            printWebsiteConfig(s3Client, bucketName);  
  
            // Delete the website configuration.  
            s3Client.deleteBucketWebsiteConfiguration(bucketName);  
  
            // Verify that the website configuration was deleted by printing it.  
            printWebsiteConfig(s3Client, bucketName);  
        }  
        catch(AmazonServiceException e) {  
            // The call was transmitted successfully, but Amazon S3 couldn't process  
            // it, so it returned an error response.  
            e.printStackTrace();  
        }  
        catch(SdkClientException e) {  
            // Amazon S3 couldn't be contacted for a response, or the client  
            // couldn't parse the response from Amazon S3.  
            e.printStackTrace();  
        }  
    }  
}
```

```
}

private static void printWebsiteConfig(AmazonS3 s3Client, String bucketName) {
    System.out.println("Website configuration: ");
    BucketWebsiteConfiguration bucketWebsiteConfig =
s3Client.getBucketWebsiteConfiguration(bucketName);
    if (bucketWebsiteConfig == null) {
        System.out.println("No website config.");
    } else {
        System.out.println("Index doc: " +
bucketWebsiteConfig.getIndexDocumentSuffix());
        System.out.println("Error doc: " + bucketWebsiteConfig.getErrorDocument());
    }
}
```

## Gerenciamento de sites com o AWS SDK para .NET

O exemplo a seguir mostra como usar o AWS SDK para .NET para gerenciar a configuração de site para um bucket. Para adicionar uma configuração de site a um bucket, forneça um nome de bucket e uma configuração de site. A configuração de site deve incluir um documento de índice e pode conter um documento de erro opcional. Esses documentos devem ser armazenados no bucket. Para obter mais informações, consulte [PUT em site de bucket](#). Para obter mais informações sobre o recurso de site do Amazon S3, consulte [Hospedagem de um site estático no Amazon S3 \(p. 494\)](#).

### Example

O exemplo de código C# a seguir adiciona uma configuração de site ao bucket especificado. A configuração especifica o documento de índice e os nomes de documento de erros. Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-
developer-guide/blob/master/LICENSE-SAMPLECODE.)

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

namespace Amazon.DocSamples.S3
{
    class WebsiteConfigTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string indexDocumentSuffix = "*** index object key ***"; // For
example, index.html.
        private const string errorDocument = "*** error object key ***"; // For example,
error.html.
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 client;
        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            AddWebsiteConfigurationAsync(bucketName, indexDocumentSuffix,
errorDocument).Wait();
        }

        static async Task AddWebsiteConfigurationAsync(string bucketName,
                string indexDocumentSuffix,
```

```
        string errorDocument)
    {
        try
        {
            // 1. Put the website configuration.
            PutBucketWebsiteRequest putRequest = new PutBucketWebsiteRequest()
            {
                BucketName = bucketName,
                WebsiteConfiguration = new WebsiteConfiguration()
                {
                    IndexDocumentSuffix = indexDocumentSuffix,
                    ErrorDocument = errorDocument
                }
            };
            PutBucketWebsiteResponse response = await
client.PutBucketWebsiteAsync(putRequest);

            // 2. Get the website configuration.
            GetBucketWebsiteRequest getRequest = new GetBucketWebsiteRequest()
            {
                BucketName = bucketName
            };
            GetBucketWebsiteResponse getResponse = await
client.GetBucketWebsiteAsync(getRequest);
            Console.WriteLine("Index document: {0}",
getResponse.WebsiteConfiguration.IndexDocumentSuffix);
            Console.WriteLine("Error document: {0}",
getResponse.WebsiteConfiguration.ErrorDocument);
        }
        catch (AmazonS3Exception e)
        {
            Console.WriteLine("Error encountered on server. Message:'{0}' when writing
an object", e.Message);
        }
        catch (Exception e)
        {
            Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
        }
    }
}
```

## Gerenciamento de sites com o AWS SDK para PHP

Este tópico explica como usar classes do AWS SDK para PHP para configurar e gerenciar um bucket do Amazon S3 para hospedagem de sites. Pressupõe-se que você já esteja seguindo as instruções para [Usar o AWS SDK para PHP e executar exemplos do PHP \(p. 649\)](#) e tenha o AWS SDK para PHP devidamente instalado. Para obter mais informações sobre o recurso de site do Amazon S3, consulte [Hospedagem de um site estático no Amazon S3 \(p. 494\)](#).

O exemplo de PHP a seguir adiciona uma configuração de site ao bucket especificado. O método `create_website_config` fornece explicitamente os nomes de documentos de índice e de documentos de erros. O exemplo também recupera a configuração de site e imprime a resposta. Para obter mais informações sobre o recurso de site do Amazon S3, consulte [Hospedagem de um site estático no Amazon S3 \(p. 494\)](#).

Para obter instruções sobre criar e testar um exemplo funcional, consulte [Usar o AWS SDK para PHP e executar exemplos do PHP \(p. 649\)](#).

```
<?php
```

```
require 'vendor/autoload.php';

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);


// Add the website configuration.
$s3->putBucketWebsite([
    'Bucket'           => $bucket,
    'WebsiteConfiguration' => [
        'IndexDocument' => ['Suffix' => 'index.html'],
        'ErrorDocument'  => ['Key' => 'error.html']
    ]
]);


// Retrieve the website configuration.
$result = $s3->getBucketWebsite([
    'Bucket' => $bucket
]);
echo $result->getPath('IndexDocument/Suffix');


// Delete the website configuration.
$s3->deleteBucketWebsite([
    'Bucket' => $bucket
]);
```

## Recursos relacionados

- [AWS SDK para PHP para classe Aws\S3\S3Client do Amazon S3](#)
- [Documentação do AWS SDK para PHP](#)

## Gerenciamento de sites com a API REST

Você pode usar o Console de gerenciamento da AWS ou o AWS SDK para configurar um bucket como um site. Contudo, se o seu aplicativo exigir, você pode enviar solicitações REST diretamente. Para obter mais informações, consulte as seguintes seções no Amazon Simple Storage Service API Reference.

- [PUT em site de bucket](#)
- [GET em site de bucket](#)
- [Site de DELETE Bucket](#)

## Amazon S3 Transfer Acceleration

O Amazon S3 Transfer Acceleration possibilita transferências de arquivos rápidas, fáceis e seguras em longas distâncias entre o seu cliente e um bucket do S3. O Transfer Acceleration aproveita os pontos de presença distribuídos globalmente do Amazon CloudFront. Conforme os dados chegam em um ponto de presença, eles são roteados para o Amazon S3 por um caminho de rede otimizado.

Ao usar o Transfer Acceleration, cobranças de transferência de dados adicionais podem ser aplicadas. Para obter mais informações sobre definição de preço, consulte [Definição de preço do Amazon S3](#).

## Tópicos

- [Por que usar o Amazon S3 Transfer Acceleration? \(p. 76\)](#)
- [Conceitos básicos do Amazon S3 Transfer Acceleration \(p. 76\)](#)
- [Requisitos para usar o Amazon S3 Transfer Acceleration \(p. 77\)](#)
- [Exemplos do Amazon S3 Transfer Acceleration \(p. 78\)](#)

## Por que usar o Amazon S3 Transfer Acceleration?

Você pode querer usar o Transfer Acceleration em um bucket por vários motivos, incluindo o seguinte:

- Você tem clientes que fazem upload em um bucket centralizado do mundo todo.
- Você transfere gigabytes a terabytes de dados regularmente entre continentes.
- Não é possível utilizar toda a largura de banda disponível via Internet ao fazer upload para o Amazon S3.

Para obter mais informações sobre quando usar o Transfer Acceleration, consulte [Perguntas frequentes do Amazon S3](#).

## Uso da ferramenta de comparação de velocidade do Amazon S3 Transfer Acceleration

Você pode usar a [ferramenta de comparação de velocidade do Amazon S3 Transfer Acceleration](#) para comparar velocidades de upload aceleradas e não aceleradas entre regiões do Amazon S3. A ferramenta de comparação de velocidades usa multipart uploads para transferir um arquivo do seu navegador para várias regiões do Amazon S3 com e sem o uso do Transfer Acceleration.

Você pode acessar a ferramenta de comparação de velocidade usando qualquer um dos seguintes métodos:

- Copie o seguinte URL na janela do navegador, substituindo `region` pela região que você está usando (por exemplo, us-west-2) e `yourBucketName` pelo nome do bucket que deseja avaliar:

```
http://s3-accelerate-speedtest.s3-accelerate.amazonaws.com/en/accelerate-speed-comparsion.html?region=region&origBucketName=yourBucketName
```

Para obter uma lista das regiões compatíveis com o Amazon S3 consulte [Regiões e endpoints](#) no Referência geral do Amazon Web Services.

- Use o console do Amazon S3. Para obter detalhes, consulte [Habilitar o Transfer Acceleration](#) no Guia do usuário do console do Amazon Simple Storage Service.

## Conceitos básicos do Amazon S3 Transfer Acceleration

Para começar a usar o Amazon S3 Transfer Acceleration, realize as seguintes etapas:

1. Ative o Transfer Acceleration em um bucket – para que seu bucket trabalhe com Transfer Acceleration, o nome do bucket deve estar em conformidade com os requisitos de nomenclatura de DNS e não deve conter pontos (".").

Você pode habilitar o Transfer Acceleration em um bucket das seguintes formas:

- Use o console do Amazon S3. Para obter mais informações, consulte [Habilitar o Transfer Acceleration](#) no Guia do usuário do console do Amazon Simple Storage Service.
  - Use a operação [PUT Bucket accelerate](#) da API REST.
  - Use a CLI da AWS e os SDKs da AWS. Para obter mais informações, consulte [Usar os AWS SDKs, a CLI e os Explorers \(p. 639\)](#).
2. Transfira dados para e do bucket habilitado para aceleração, usando um dos seguintes nomes de domínio de endpoint do s3-accelerate:
- `bucketname.s3-accelerate.amazonaws.com` – para acessar um bucket habilitado para aceleração.
  - `bucketname.s3-accelerate.dualstack.amazonaws.com` – para acessar um bucket habilitado para aceleração por meio de IPv6. Os endpoints de pilha dupla do Amazon S3 oferecem suporte a buckets do S3 por meio de IPv6 e IPv4. O endpoint de pilha dupla do Transfer Acceleration usa somente o tipo virtual hospedado de nome do endpoint. Para obter mais informações, consulte [Conceitos básicos para fazer solicitações por meio do IPv6 \(p. 12\)](#) e [Usar endpoints de pilha dupla do Amazon S3 \(p. 14\)](#).

#### Important

O suporte para o endpoint acelerado de pilha dupla está atualmente disponível somente no SDK do Java da AWS. O suporte para a CLI da AWS e outros SDKs da AWS será disponibilizado em breve.

#### Note

Você pode continuar a usar o endpoint regular além dos endpoints de aceleração.

Você pode apontar as solicitações de objeto PUT e objeto GET do Amazon S3 para o nome de domínio do endpoint do s3-accelerate depois de habilitar o Transfer Acceleration. Por exemplo, digamos que, atualmente, você tem um aplicativo de API REST usando o [objeto PUT](#) que usa o nome de host `mybucket.s3.amazonaws.com` na solicitação `PUT`. Para acelerar o `PUT`, basta alterar o nome do host na solicitação para `mybucket.s3-accelerate.amazonaws.com`. Para voltar a usar a velocidade de upload padrão, basta alterar o nome novamente para `mybucket.s3.amazonaws.com`.

Depois que o Transfer Acceleration é ativado, pode demorar 20 minutos para você perceber o benefício de desempenho. Contudo, o endpoint de aceleração estará disponível assim que você habilitar o Transfer Acceleration.

Você pode usar o endpoint de aceleração na CLI da AWS, SDKs da AWS e outras ferramentas que transferem dados para e do Amazon S3. Se você estiver usando SDKs da AWS, algumas linguagens compatíveis usam uma sinalização de configuração de cliente do endpoint de aceleração para que você não precise definir explicitamente o endpoint do Transfer Acceleration como `bucketname.s3-accelerate.amazonaws.com`. Para ver exemplos de como usar uma sinalização de configuração de cliente do endpoint de aceleração, consulte [Exemplos do Amazon S3 Transfer Acceleration \(p. 78\)](#).

Você pode usar todas as operações do Amazon S3 por meio dos endpoints de aceleração de transação, exceto para as seguintes operações: [GET Service \(listar buckets\)](#), [PUT Bucket \(criar bucket\)](#) e [DELETE Bucket](#). Além disso, o Amazon S3 Transfer Acceleration não oferece suporte para cópias entre regiões usando [Objeto PUT - Copiar](#).

## Requisitos para usar o Amazon S3 Transfer Acceleration

---

Veja a seguir os requisitos para usar o Transfer Acceleration em um bucket do S3:

Versão da API 2006-03-01

- O Transfer Acceleration é compatível somente com solicitações de estilo virtual. Para obter mais informações sobre solicitações de estilo virtual, consulte [Fazer solicitações usando a API REST \(p. 45\)](#).
- O nome do bucket usado para o Transfer Acceleration deve ser compatível com DNS e não deve conter pontos (".").
- O Transfer Acceleration deve ser ativado no bucket. Depois de habilitar o Transfer Acceleration em um bucket, pode demorar até trinta minutos para que a velocidade de transferência de dados para o bucket aumente.
- Para acessar o bucket habilitado para Transfer Acceleration, você deve usar o endpoint `bucketname.s3-accelerate.amazonaws.com` ou o endpoint de pilha dupla `bucketname.s3-accelerate.dualstack.amazonaws.com` para se conectar ao bucket habilitado por meio de IPv6.
- Você deve ser o proprietário do bucket para configurar o estado de aceleração de transferência. O proprietário do bucket pode designar permissões para outros usuários para permitir que eles definam o estado de aceleração em um bucket. A permissão `s3:PutAccelerateConfiguration` autoriza os usuários a habilitar ou desabilitar o Transfer Acceleration em um bucket. A permissão `s3:GetAccelerateConfiguration` autoriza os usuários a retornar o estado do Transfer Acceleration de um bucket, que é `Enabled` ou `Suspended`. Para obter mais informações sobre essas permissões, consulte [Permissões relacionadas a operações de sub-recurso de bucket \(p. 332\)](#) e [Gerenciamento de permissões de acesso aos recursos do Amazon S3 \(p. 282\)](#).

## Mais informações

- [GET Aceleração do bucket](#)
- [Aceleração do bucket PUT](#)

## Exemplos do Amazon S3 Transfer Acceleration

Esta seção fornece exemplos de como habilitar o Amazon S3 Transfer Acceleration em um bucket e usar o endpoint de aceleração para o bucket ativado. Algumas linguagens compatíveis com o AWS SDK (por exemplo, Java e .NET) usam uma sinalização de configuração de cliente do endpoint de aceleração para que você não precise definir explicitamente o endpoint do Transfer Acceleration como `bucketname.s3-accelerate.amazonaws.com`. Para obter mais informações sobre Transfer Acceleration, consulte [Amazon S3 Transfer Acceleration \(p. 75\)](#).

### Tópicos

- [Usar o console do Amazon S3 \(p. 78\)](#)
- [Usar o Transfer Acceleration na AWS Command Line Interface \(AWS CLI\) \(p. 79\)](#)
- [Usar o Transfer Acceleration com o AWS SDK for Java \(p. 80\)](#)
- [Uso do Transfer Acceleration no AWS SDK para .NET \(p. 81\)](#)
- [Uso do Transfer Acceleration no AWS SDK para JavaScript \(p. 82\)](#)
- [Usar o Transfer Acceleration na AWS SDK for Python \(Boto\) \(p. 82\)](#)
- [Uso de outros AWS SDKs \(p. 82\)](#)

## Usar o console do Amazon S3

Para obter informações sobre como habilitar o Transfer Acceleration em um bucket usando o console do Amazon S3, consulte [Habilitar o Transfer Acceleration](#) no Guia do usuário do console do Amazon Simple Storage Service.

## Usar o Transfer Acceleration na AWS Command Line Interface (AWS CLI)

Esta seção fornece exemplos de comandos da AWS CLI usados para o Transfer Acceleration. Para obter instruções de configuração da AWS CLI, consulte [Configurar a CLI da AWS \(p. 645\)](#).

### Ativação do Transfer Acceleration em um bucket usando a AWS CLI

Use o comando [put-bucket-accelerate-configuration](#) da AWS CLI para habilitar ou suspender o Transfer Acceleration em um bucket. O exemplo a seguir define Status=Enabled para habilitar o Transfer Acceleration em um bucket. Use Status=Suspended para suspender o Transfer Acceleration.

Example

```
$ aws s3api put-bucket-accelerate-configuration --bucket bucketname --accelerate-configuration Status=Enabled
```

### Usar o Transfer Acceleration na AWS CLI

Definir o valor de configuração use\_accelerate\_endpoint como true em um perfil em seu arquivo AWS Config direcionará todas as solicitações do Amazon S3 feitas pelos comandos do s3 e da AWS CLI do s3api para o endpoint de aceleração: s3-accelerate.amazonaws.com. O Transfer Acceleration deve ser habilitado em seu bucket para usar o endpoint de aceleração.

Todas as solicitações são enviadas, usando o estilo virtual de endereçamento de bucket: my-bucket.s3-accelerate.amazonaws.com. As solicitações ListBuckets, CreateBucket e DeleteBucket não serão enviadas ao endpoint de aceleração porque o endpoint não oferece suporte a essas operações. Para obter mais informações sobre use\_accelerate\_endpoint, consulte [Configuração da AWS CLI S3](#).

O exemplo a seguir define use\_accelerate\_endpoint como true no perfil padrão.

Example

```
$ aws configure set default.s3.use_accelerate_endpoint true
```

Se você quiser usar o endpoint de aceleração para alguns comandos da AWS CLI, mas não para outros, use qualquer um destes dois métodos:

- Você pode usar o endpoint de aceleração por comando, definindo o parâmetro --endpoint-url como <https://s3-accelerate.amazonaws.com> ou <http://s3-accelerate.amazonaws.com> para qualquer comando s3 ou s3api.
- Você pode configurar perfis separados em seu arquivo AWS Config. Por exemplo, crie um perfil que defina use\_accelerate\_endpoint como true e um perfil que não defina use\_accelerate\_endpoint. Quando você executar um comando, especifique qual perfil deseja usar, caso queira ou não usar o endpoint de aceleração.

### Exemplos da AWS CLI de upload de um objeto em um bucket habilitado para o Transfer Acceleration

O exemplo a seguir faz upload de um arquivo em um bucket habilitado para o Transfer Acceleration usando o perfil padrão que foi configurado para usar o endpoint de aceleração.

Example

```
$ aws s3 cp file.txt s3://bucketname/keyname --region region
```

O exemplo a seguir faz upload de um arquivo em um bucket habilitado para o Transfer Acceleration usando o parâmetro `--endpoint-url` para especificar o endpoint de aceleração.

Example

```
$ aws configure set s3.addressing_style virtual
$ aws s3 cp file.txt s3://bucketname/keyname --region region --endpoint-url http://s3-accelerate.amazonaws.com
```

## Usar o Transfer Acceleration com o AWS SDK for Java

Example

O exemplo a seguir mostra como usar um endpoint de aceleração para fazer upload de um objeto no Amazon S3. O exemplo faz o seguinte:

- Cria um `AmazonS3Client` que é configurado para usar endpoints de aceleração. Todos os buckets acessados pelo cliente devem ter a aceleração da transferência habilitada.
- Permite a aceleração da transferência em um bucket especificado. Essa etapa é necessária somente se o bucket que você especificar não tiver a aceleração de transferência habilitada ainda.
- Verifica se a aceleração da transferência está habilitada para o bucket especificado.
- Faz upload de um novo objeto para o bucket especificado usando o endpoint de aceleração do bucket.

Para obter mais informações sobre o uso de Transfer Acceleration, consulte [Conceitos básicos do Amazon S3 Transfer Acceleration \(p. 76\)](#). Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.BucketAccelerateConfiguration;
import com.amazonaws.services.s3.model.BucketAccelerateStatus;
import com.amazonaws.services.s3.model.GetBucketAccelerateConfigurationRequest;
import com.amazonaws.services.s3.model.SetBucketAccelerateConfigurationRequest;

public class TransferAcceleration {
    public static void main(String[] args) {
        String clientRegion = "**** Client region ****";
        String bucketName = "**** Bucket name ****";
        String keyName = "**** Key name ****";

        try {
            // Create an Amazon S3 client that is configured to use the accelerate endpoint.
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .withRegion(clientRegion)
                .withCredentials(new ProfileCredentialsProvider())
                .enableAccelerateMode()
        }
    }
}
```

```
.build();

// Enable Transfer Acceleration for the specified bucket.
s3Client.setBucketAccelerateConfiguration(
    new SetBucketAccelerateConfigurationRequest(bucketName,
        new
BucketAccelerateConfiguration(
    BucketAccelerateStatus.Enabled)));

// Verify that transfer acceleration is enabled for the bucket.
String accelerateStatus = s3Client.getBucketAccelerateConfiguration(
    new
GetBucketAccelerateConfigurationRequest(bucketName))
    .getStatus();
System.out.println("Bucket accelerate status: " + accelerateStatus);

// Upload a new object using the accelerate endpoint.
s3Client.putObject(bucketName, keyName, "Test object for transfer
acceleration");
System.out.println("Object \\" + keyName + "\\" uploaded with transfer
acceleration.");
}
catch(AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
}
catch(SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
}
```

## Uso do Transfer Acceleration no AWS SDK para .NET

O exemplo a seguir mostra como usar o AWS SDK para .NET para habilitar o Transfer Acceleration em um bucket. Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

### Example

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-
developer-guide/blob/master/LICENSE-SAMPLECODE.)

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

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

```
s3Client = new AmazonS3Client(bucketRegion);
EnableAccelerationAsync().Wait();
}

static async Task EnableAccelerationAsync()
{
    try
    {
        var putRequest = new PutBucketAccelerateConfigurationRequest
        {
            BucketName = bucketName,
            AccelerateConfiguration = new AccelerateConfiguration
            {
                Status = BucketAccelerateStatus.Enabled
            }
        };
        await s3Client.PutBucketAccelerateConfigurationAsync(putRequest);

        var getRequest = new GetBucketAccelerateConfigurationRequest
        {
            BucketName = bucketName
        };
        var response = await
s3Client.GetBucketAccelerateConfigurationAsync(getRequest);

        Console.WriteLine("Acceleration state = '{0}' ", response.Status);
    }
    catch (AmazonS3Exception amazonS3Exception)
    {
        Console.WriteLine(
            "Error occurred. Message:'{0}' when setting transfer acceleration",
            amazonS3Exception.Message);
    }
}
}
```

Ao fazer o upload de um objeto a um bucket com Transfer Acceleration habilitado, especifique usando o endpoint de aceleração no momento da criação de um cliente conforme segue:

```
var client = new AmazonS3Client(new AmazonS3Config
{
    RegionEndpoint = TestRegionEndpoint,
    UseAccelerateEndpoint = true
})
```

## Uso do Transfer Acceleration no AWS SDK para JavaScript

Para ver um exemplo de ativação do Transfer Acceleration usando o AWS SDK para JavaScript, consulte [Chamada da operação putBucketAccelerateConfiguration](#) na Referência de API do AWS SDK para JavaScript.

## Usar o Transfer Acceleration na AWS SDK for Python (Boto)

Para ver um exemplo de ativação do Transfer Acceleration usando o SDK for Python, consulte [put\\_bucket\\_accelerate\\_configuration](#) no AWS SDK for Python (Boto 3) API Reference.

## Uso de outros AWS SDKs

Para obter informações sobre como usar outros AWS SDKs, consulte [Código de exemplo e bibliotecas](#).

# Buckets de Pagamento pelo solicitante

## Tópicos

- [Configurar Pagamento pelo solicitante usando o console do Amazon S3 \(p. 83\)](#)
- [Configurar Pagamento pelo solicitante com a API REST \(p. 84\)](#)
- [Detalhes da cobrança \(p. 86\)](#)

Geralmente, proprietários de bucket pagam por todos os custos de armazenamento e transferência de dados do Amazon S3 associados ao bucket. No entanto, um proprietário de bucket, pode configurar o bucket como um bucket de Pagamento pelo solicitante. Com buckets de Pagamento pelo solicitante, é o solicitante, em vez de o proprietário do bucket, quem paga pelo custo da solicitação e de download de dados do bucket. O proprietário do bucket sempre paga pelo custo de armazenamento de dados.

Normalmente, você configura buckets como Pagamento pelo solicitante quando quer compartilhar dados, mas não quer incorrer em cobranças associadas a outros que acessam os dados. Você pode, por exemplo, usar buckets de Pagamento pelo solicitante ao disponibilizar grandes conjuntos de dados, tais como diretórios de CEP, dados de referência, informações geoespaciais ou dados de crawling da web.

### Important

Se você habilitar Pagamento pelo solicitante em um bucket, o acesso anônimo a esse bucket não será permitido.

Você deve autenticar todas as solicitações que envolvem buckets de Pagamento pelo solicitante. A autenticação de solicitação permite que o Amazon S3 identifique e cobre o solicitante pelo uso do bucket de Pagamento pelo solicitante.

Quando o solicitante pressupõe uma função do AWS Identity and Access Management (IAM) antes de fazer a solicitação, a conta à qual a função pertence é cobrada pela solicitação. Para obter mais informações sobre funções do IAM, consulte [Funções do IAM](#) no Guia do usuário do IAM.

Após configurar um bucket para ser um bucket de Pagamento pelo solicitante, os solicitantes devem incluir `x-amz-request-payer` em suas solicitações no cabeçalho, para solicitações POST, GET e HEAD, ou como um parâmetro em uma solicitação REST para mostrar que entendem que serão cobrados pela solicitação e pelo download dos dados.

Os buckets de Pagamento pelo solicitante não oferecem suporte aos itens a seguir.

- Solicitações anônimas
- BitTorrent
- Solicitações de SOAP
- Você não pode usar um bucket de Pagamento pelo solicitante como o bucket de destino para registro em log de usuário final ou vice-versa. Contudo, você pode habilitar o registro em log de usuário final em um bucket de Pagamento pelo solicitante onde o bucket de destino não é um bucket de Pagamento pelo solicitante.

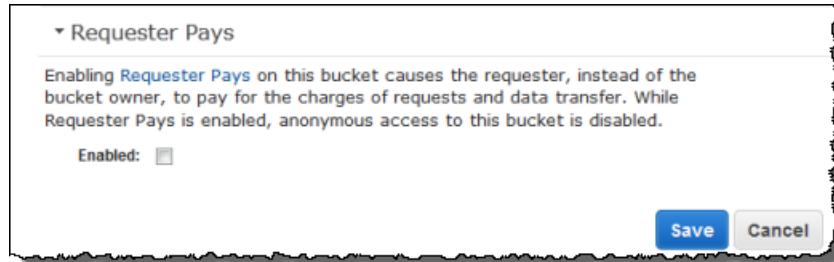
## Configurar Pagamento pelo solicitante usando o console do Amazon S3

Você pode configurar um bucket para Pagamento pelo solicitante usando o console do Amazon S3.

Para configurar um bucket para Pagamento pelo solicitante

1. Faça login no Console de gerenciamento da AWS e abra o console da Amazon S3 em <https://console.aws.amazon.com/s3/>.

2. Na lista Buckets, clique no ícone de detalhes à esquerda do nome do bucket e em Properties (Propriedades) para exibir as propriedades do bucket.
3. No painel Properties (Propriedades), clique em Requester Pays (Pagamento pelo solicitante).
4. Marque a caixa de seleção Enabled (Habilitado).



## Configurar Pagamento pelo solicitante com a API REST

### Tópicos

- [Definição da configuração do bucket requestPayment \(p. 84\)](#)
- [Recuperação da configuração de requestPayment \(p. 85\)](#)
- [Download de objetos em buckets de Pagamento pelo solicitante \(p. 85\)](#)

## Definição da configuração do bucket requestPayment

Somente o proprietário do bucket pode definir o valor de configuração `RequestPaymentConfiguration.payer` de um bucket como `BucketOwner`, o padrão, ou `Requester`. A definição do recurso `requestPayment` é opcional. Por padrão, o bucket não é um bucket de Pagamento pelo solicitante.

Para reverter um bucket de Pagamento pelo solicitante para um bucket regular, use o valor `BucketOwner`. Normalmente, você usaria `BucketOwner` ao carregar dados para o bucket do Amazon S3 e definiria o valor como `Requester` antes da publicação de objetos no bucket.

### Para definir requestPayment

- Use uma solicitação `PUT` para definir o valor `Payer` como `Requester` em um bucket especificado.

```
PUT ?requestPayment HTTP/1.1
Host: [BucketName].s3.amazonaws.com
Content-Length: 173
Date: Wed, 01 Mar 2009 12:00:00 GMT
Authorization: AWS [Signature]

<RequestPaymentConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<Payer>Requester</Payer>
</RequestPaymentConfiguration>
```

Se a solicitação for bem-sucedida, o Amazon S3 retornará uma resposta similar a esta.

```
HTTP/1.1 200 OK
x-amz-id-2: [id]
x-amz-request-id: [request_id]
```

```
Date: Wed, 01 Mar 2009 12:00:00 GMT
Content-Length: 0
Connection: close
Server: AmazonS3
x-amz-request-charged:requester
```

Você pode definir Pagamento pelo solicitante somente no nível do bucket; você não pode definir Pagamento pelo solicitante para objetos específicos no bucket.

Você pode configurar um bucket para ser **BucketOwner** ou **Requester** a qualquer momento. Contudo, tenha em mente que pode haver um pequeno atraso, de alguns minutos, antes que o valor da nova configuração entre em vigor.

#### Note

Proprietários de bucket que abrem mão de pre-signed URLs devem pensar duas vezes antes de configurar um bucket para ser de Pagamento pelo solicitante, especialmente se o URL tiver um ciclo de vida bem longo. O proprietário do bucket é cobrado cada vez que o solicitante usa um pre-signed URL que usa as credenciais do proprietário do bucket.

## Recuperação da configuração de requestPayment

Você pode determinar o **Payer** valor que é definido em um bucket solicitando o recurso **requestPayment**.

Para retornar o recurso **requestPayment**

- Use uma solicitação GET para obter o recurso **requestPayment**, conforme exibido na seguinte solicitação.

```
GET ?requestPayment HTTP/1.1
Host: [BucketName].s3.amazonaws.com
Date: Wed, 01 Mar 2009 12:00:00 GMT
Authorization: AWS [Signature]
```

Se a solicitação for bem-sucedida, o Amazon S3 retornará uma resposta similar a esta.

```
HTTP/1.1 200 OK
x-amz-id-2: [id]
x-amz-request-id: [request_id]
Date: Wed, 01 Mar 2009 12:00:00 GMT
Content-Type: [type]
Content-Length: [length]
Connection: close
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<RequestPaymentConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<Payer>Requester</Payer>
</RequestPaymentConfiguration>
```

Esta resposta mostra que o valor **payer** é definido como **Requester**.

## Download de objetos em buckets de Pagamento pelo solicitante

Como os solicitantes serão cobrados pelo download de dados dos buckets de Pagamento pelo solicitante, as solicitações deverão conter um parâmetro especial, **x-amz-request-payer**, que confirma que o solicitante sabe que será cobrado pelo download. Para acessar objetos em buckets de Pagamento pelo solicitante, as solicitações devem incluir um dos seguintes.

- Para solicitações GET, HEAD e POST, inclua `x-amz-request-payer : requester` no cabeçalho
- Para URLs assinados, inclua `x-amz-request-payer=requester` na solicitação

Se a solicitação for bem-sucedida e o solicitante for cobrado, a resposta incluirá o cabeçalho `x-amz-request-charged:requester`. Se `x-amz-request-payer` não estiver na solicitação, o Amazon S3 retornará um erro 403 e cobrará o proprietário do bucket pela solicitação.

**Note**

Proprietários de bucket não precisam adicionar `x-amz-request-payer` às suas solicitações. Certifique-se de que você tenha incluído `x-amz-request-payer` e seu valor no cálculo da assinatura. Para obter mais informações, consulte [Criar o elemento CanonicalizedAmzHeaders \(p. 662\)](#).

Para fazer download de objetos em um bucket de Pagamento pelo solicitante

- Use uma solicitação GET para fazer download de um objeto em um bucket de Pagamento pelo solicitante, conforme exibido na seguinte solicitação.

```
GET / [destinationObject] HTTP/1.1
Host: [BucketName].s3.amazonaws.com
x-amz-request-payer : requester
Date: Wed, 01 Mar 2009 12:00:00 GMT
Authorization: AWS [Signature]
```

Se a solicitação GET for bem-sucedida e o solicitante for cobrado, a resposta incluirá `x-amz-request-charged:requester`.

O Amazon S3 poderá retornar um erro `Access Denied` para solicitações que tentarem obter objetos de um bucket de Pagamento pelo solicitante. Para obter mais informações, consulte [Respostas de erro](#).

## Detalhes da cobrança

A cobrança por solicitações de Pagamento pelo solicitante bem-sucedidas é direta: o solicitante paga pela transferência de dados e pela solicitação; o proprietário do bucket paga pelo armazenamento de dados físico. Contudo, o proprietário do bucket é cobrado pela solicitação nas seguintes condições:

- O solicitante não inclui o parâmetro `x-amz-request-payer` no cabeçalho (GET, HEAD ou POST) ou como um parâmetro (REST) na solicitação (código HTTP 403).
- Falha na autenticação da solicitação (código HTTP 403).
- A solicitação é anônima (código HTTP 403).
- A solicitação é uma solicitação SOAP.

## Controle de acesso e buckets

Cada bucket tem uma política de controle de acesso associada. Essa política governa a criação, a exclusão e a enumeração de objetos no bucket. Para obter mais informações, consulte [Gerenciamento de permissões de acesso aos recursos do Amazon S3 \(p. 282\)](#).

## Relatórios de uso e faturamento dos buckets do S3

Ao usar o Amazon Simple Storage Service (Amazon S3), você não tem que pagar nenhuma taxa inicial nem assumir nenhum compromisso em relação à quantidade de conteúdo que armazenará. Quanto aos

outros serviços da Amazon Web Services (AWS), você paga conforme usar – e apenas por aquilo que usar.

A AWS fornece os seguintes relatórios do Amazon S3:

- Relatórios de faturamento – vários relatórios que fornecem visualizações de alto nível de todas as atividades dos serviços da AWS que você está usando, incluindo o Amazon S3. A AWS sempre cobra do proprietário do bucket do S3 as taxas do Amazon S3, a menos que o bucket tenha sido criado como um bucket de Pagamento pelo solicitante. Para obter mais informações sobre Pagamento pelo solicitante, consulte [Buckets de Pagamento pelo solicitante \(p. 83\)](#). Para obter mais informações sobre relatórios de faturamento, consulte [Relatórios de faturamento da AWS para Amazon S3 \(p. 87\)](#).
- Relatório de uso – um resumo da atividade de um serviço específico, agregado por hora, dia ou mês. Você pode escolher qual tipo e operação de uso incluir. Também é possível escolher a forma como os dados são agrupados. Para obter mais informações, consulte [Relatório de uso da AWS para o Amazon S3 \(p. 89\)](#).

Os tópicos seguintes fornecem informações sobre os relatórios de uso e faturamento do Amazon S3.

#### Tópicos

- [Relatórios de faturamento da AWS para Amazon S3 \(p. 87\)](#)
- [Relatório de uso da AWS para o Amazon S3 \(p. 89\)](#)
- [Entender seus relatórios de uso e faturamento da AWS para Amazon S3 \(p. 91\)](#)
- [Usar tags de alocação de custos para buckets do S3 \(p. 99\)](#)

## Relatórios de faturamento da AWS para Amazon S3

Sua fatura mensal da AWS separa suas informações de uso e o custo de um serviço e uma função da AWS. Há vários relatórios de faturamento da AWS disponíveis, o relatório mensal, o relatório de alocação de custos e os relatórios detalhados de faturamento. Para obter informações sobre como ver seus relatórios de faturamento, consulte [Exibir sua fatura](#) no Guia do usuário do AWS Billing and Cost Management.

Você também pode fazer download de um relatório de uso que forneça mais detalhes sobre o uso do armazenamento do Amazon S3 e os relatórios de faturamento. Para obter mais informações, consulte [Relatório de uso da AWS para o Amazon S3 \(p. 89\)](#).

A tabela a seguir lista as cobranças associadas ao uso do Amazon S3.

#### Cobranças de uso do Amazon S3

Cobrança	Comentários
Armazenamento	Você paga para armazenar objetos em seu bucket do S3. A taxa pela qual você é cobrado depende do tamanho dos seus objetos, quanto tempo você os armazenou durante o mês e a classe do armazenamento—STANDARD, INTELLIGENT_TIERING, STANDARD_IA (IA para acesso pouco frequente), ONEZONE_IA, GLACIER ou Reduced Redundancy Storage (RRS). Para obter mais informações sobre classes de armazenamento, consulte <a href="#">Classes de armazenamento (p. 107)</a> .

Cobrança	Comentários
Monitoramento e automação	Você paga uma taxa mensal de monitoramento e automação por objeto armazenado na classe de armazenamento INTELLIGENT_TIERING para monitorar padrões de acesso e mover objetos entre níveis de acesso em INTELLIGENT_TIERING.
Solicitações	Você paga por solicitações, por exemplo, solicitações GET, feitas em seus buckets e objetos do S3. Isso inclui solicitações de ciclo de vida. As taxas para solicitações dependem de qual tipo de solicitação você está fazendo. Para obter informações sobre a definição de preço das solicitações, consulte <a href="#">Definição de preço do Amazon S3</a> .
Recuperações	Você paga para recuperar objetos que são armazenados no armazenamento do STANDARD_IA, ONEZONE_IA e GLACIER.
Exclusões adiantadas	Se você excluir um objeto no armazenamento INTELLIGENT_TIERING, STANDARD_IA, ONEZONE_IA ou GLACIER antes que o compromisso de armazenamento mínimo tenha passado, pagará por uma exclusão antecipada desse objeto.
Gerenciamento de armazenamento	Você paga pelos recursos de gerenciamento de armazenamento (inventário, análise e marcação com tags de objetos do Amazon S3) que são ativados nos buckets da sua conta.
Largura de banda	<p>Você paga por toda a largura de entrada e saída do Amazon S3, exceto pelo seguinte:</p> <ul style="list-style-type: none"> <li>• Dados transferidos da Internet</li> <li>• Dados transferidos para uma instância do Amazon Elastic Compute Cloud (Amazon EC2), quando a instância está na mesma região da AWS que o bucket do S3</li> <li>• Dados transferidos para o Amazon CloudFront (CloudFront)</li> </ul> <p>Você também paga uma taxa por todos os dados transferidos por meio da Amazon S3 Transfer Acceleration.</p>

Para obter informações detalhadas sobre as cobranças de uso do Amazon S3 para armazenamento, transferência de dados e serviços, consulte [Definição de preço do Amazon S3](#) e as [Perguntas frequentes do Amazon S3](#).

Para obter informações sobre como entender os códigos e as abreviações usadas nos relatórios de uso e faturamento do Amazon S3, consulte [Entender seus relatórios de uso e faturamento da AWS para Amazon S3 \(p. 91\)](#).

## Mais informações

- Relatório de uso da AWS para o Amazon S3 (p. 89)
- Usar tags de alocação de custos para buckets do S3 (p. 99)
- Gerenciamento de custos e faturamento da AWS
- Definição de preço do Amazon S3
- Perguntas frequentes do Amazon S3
- Definição de preço do Glacier

## Relatório de uso da AWS para o Amazon S3

Para obter mais detalhes sobre o uso do armazenamento do Amazon S3, baixe os relatórios de uso da AWS gerados dinamicamente. Você pode escolher qual tipo de uso, operação e período de tempo incluir. Também é possível escolher a forma como os dados são agregados.

Ao fazer download de um relatório de uso, você pode optar por agregar os dados de uso por hora, dia ou mês. O relatório de uso do Amazon S3 lista as operações por tipo de uso e região da AWS, por exemplo, a quantidade de dados transferidos da região da Ásia-Pacífico (Sydney).

O relatório de uso do Amazon S3 inclui as seguintes informações:

- Serviço – Amazon Simple Storage Service
- Operação – a operação realizada em seu bucket ou objeto. Para obter uma explicação detalhada das operações do Amazon S3, consulte [Controle de operações em seus relatórios de uso \(p. 98\)](#).
- UsageType – Um dos seguintes valores:
  - Um código que identifica o tipo de armazenamento
  - Um código que identifica o tipo de solicitação
  - Um código que identifica o tipo de recuperação
  - Um código que identifica o tipo de transferência de dados
  - Um código que identifica exclusões antecipadas do armazenamento INTELLIGENT\_TIERING, STANDARD\_IA, ONEZONE\_IA ou GLACIER
- StorageObjectCount – a contagem de objetos armazenados em um determinado bucket

Para obter uma explicação detalhada dos tipos de uso do Amazon S3, consulte [Entender seus relatórios de uso e faturamento da AWS para Amazon S3 \(p. 91\)](#).

- Recurso – o nome do bucket associado ao uso relacionado.
- StartTime – a hora inicial do dia ao qual o uso se aplica, no Tempo Universal Coordenado (UTC).
- EndTime – a hora de término do dia ao qual o uso se aplica, no Tempo Universal Coordenado (UTC).
- UsageValue – um dos valores de volume a seguir:
  - O número de solicitações durante o período especificado
  - A quantidade de dados transferidos, em bytes
  - A quantidade de dados armazenados, em bytes por hora, que é o número de bytes armazenados em uma determinada hora
  - A quantidade de dados associados a restaurações do armazenamento do GLACIER, STANDARD\_IA ou ONEZONE\_IA, em bytes

### Tip

Para obter informações detalhadas sobre cada solicitação recebida pelo Amazon S3 para seus objetos, ative os logs de acesso do servidor para seus buckets. Para obter mais informações, consulte [Registro em log de acesso ao servidor Amazon S3 \(p. 625\)](#).

Você pode fazer download de um relatório de uso como um arquivo XML ou de valores separados por vírgula (CSV). A seguir há um relatório de uso de exemplo em formato CSV aberto em um aplicativo de planilha.

Service	Operation	UsageType	Resource	StartTime	EndTime	UsageValue
AmazonS3	HeadBucket	USW2-C3DataTransfer-Out-Bytes	admin-created3	6/1/2017 0:00	7/1/2017 0:00	15309
AmazonS3	PutObject	USW2-C3DataTransfer-In-Bytes	admin-created3	6/1/2017 0:00	7/1/2017 0:00	19062
AmazonS3	HeadBucket	USW2-Requests-Tier2	admin-created3	6/1/2017 0:00	7/1/2017 0:00	68
AmazonS3	PutObjectForRep1	USW1-Requests-SIA-Tier1	ca-example-bucket	6/1/2017 0:00	7/1/2017 0:00	178294
AmazonS3	PutObjectForRep1	USW1-USW2-AWS-In-Bytes	ca-example-bucket	6/1/2017 0:00	7/1/2017 0:00	387929083
AmazonS3	GetObjectForRep1	USW2-Requests-NoCharge	admin-created3	6/1/2017 0:00	7/1/2017 0:00	108
AmazonS3	GetObjectForRep1	USW2-USW1-AWS-Out-Bytes	my-test-bucket-bash	6/1/2017 0:00	7/1/2017 0:00	387910021

Para obter informações sobre como entender o relatório de uso, consulte [Entender seus relatórios de uso e faturamento da AWS para Amazon S3 \(p. 91\)](#).

## Download do relatório de uso da AWS

Você pode fazer download de um relatório de uso como um arquivo .xml ou .csv.

### Para fazer download do relatório de uso

1. Faça login no Console de gerenciamento da AWS e abra o console da Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na barra de títulos, escolha seu nome de usuário do AWS Identity and Access Management (IAM) e escolha My Billing Dashboard (Meu painel de faturamento).
3. No painel de navegação, escolha Reports (Relatórios).
4. Na seção Other Reports (Outros relatórios), escolha AWS Usage Report (Relatório de uso da AWS).
5. Para Services: (Serviços), escolha Amazon Simple Storage Service.
6. Para Download Usage Report (Fazer download do relatório de uso), escolha as seguintes configurações:
  - Usage Types (Tipos de uso) – Para obter uma explicação detalhada dos tipos de uso do Amazon S3, consulte [Entender seus relatórios de uso e faturamento da AWS para Amazon S3 \(p. 91\)](#).
  - Operation (Operação) – para obter uma explicação detalhada sobre as operações do Amazon S3, consulte [Controle de operações em seus relatórios de uso \(p. 98\)](#).
  - Time Period (Período) – o período de tempo de cobertura do relatório.
  - Report Granularity (Granularidade do relatório) – se você deseja que o relatório inclua subtotais por hora, dia ou mês.
7. Para escolher o formato do relatório, escolha Download (Fazer download) para esse formato e siga os prompts para ver ou salvar o relatório.

## Mais informações

- [Entender seus relatórios de uso e faturamento da AWS para Amazon S3 \(p. 91\)](#)
- [Relatórios de faturamento da AWS para Amazon S3 \(p. 87\)](#)

## Entender seus relatórios de uso e faturamento da AWS para Amazon S3

Os relatórios de uso e faturamento do Amazon S3 usam códigos e abreviações. Por exemplo, para o tipo de uso, que é definido na tabela seguinte, *região* é substituída por uma das seguintes abreviações:

- APN1: Ásia-Pacífico (Tóquio)
- APN2: Ásia-Pacífico (Seul)
- APS1: Ásia-Pacífico (Cingapura)
- APS2: Ásia-Pacífico (Sydney)
- APS3: Ásia Pacífico (Mumbai)
- CAN1: Canadá (Central)
- EUC1: UE (Frankfurt)
- EU: UE (Irlanda)
- EUW2: UE (Londres)
- EUW3: UE (Paris)
- SAE1: América do Sul (São Paulo)
- UGW1: AWS GovCloud (US-West)
- USE1 (or no prefix): Leste dos EUA (Norte da Virgínia)
- USE2: Leste dos EUA (Ohio)
- USW1: Oeste dos EUA (Norte da Califórnia)
- USW2: Oeste dos EUA (Oregon)

Para obter informações sobre a definição de preço por região da AWS, consulte [Definição de preço do Amazon S3](#).

A primeira coluna da tabela a seguir indica os tipos de uso que aparecem em seus relatórios de uso e faturamento.

### Tipos de uso

Tipo de uso	Unidades	Granularity	Descrição
<i>region1-region2</i> -AWS-In-Bytes	Bytes	Por hora	A quantidade de dados transferidos para a AWS Region1 da AWS Region2
<i>region1-region2</i> -AWS-Out-Bytes	Bytes	Por hora	A quantidade de dados transferidos da AWS Region1 para a AWS Region2
<i>região</i> -DataTransfer-In-Bytes	Bytes	Por hora	A quantidade de dados transferidos para o Amazon S3 pela Internet
<i>região</i> -DataTransfer-Out-Bytes	Bytes	Por hora	A quantidade de dados transferidos do Amazon S3 para Internet <sup>1</sup>
<i>region</i> -C3DataTransfer-In-Bytes	Bytes	Por hora	A quantidade de dados transferidos para o Amazon

Tipo de uso	Unidades	Granularity	Descrição
			S3 do Amazon EC2 dentro da mesma região da AWS
<i>region</i> -C3DataTransfer-Out-Bytes	Bytes	Por hora	A quantidade de dados transferidos do Amazon S3 para o Amazon EC2 dentro da mesma região da AWS
<i>region</i> -S3G-DataTransfer-In-Bytes	Bytes	Por hora	A quantidade de dados transferidos para o Amazon S3 para restaurar objetos do armazenamento do GLACIER
<i>region</i> -S3G-DataTransfer-Out-Bytes	Bytes	Por hora	A quantidade de dados transferidos do Amazon S3 para fazer a transição de objetos para o armazenamento do GLACIER
<i>region</i> -DataTransfer-Regional-Bytes	Bytes	Por hora	A quantidade de dados transferidos do Amazon S3 para os recursos da AWS dentro da mesma região da AWS
StorageObjectCount	Contagem	Diariamente	O número de objetos armazenados em um determinado bucket
<i>region</i> -CloudFront-In-Bytes	Bytes	Por hora	A quantidade de dados transferidos para uma região da AWS de uma distribuição do CloudFront
<i>region</i> -CloudFront-Out-Bytes	Bytes	Por hora	A quantidade de dados transferidos de uma região da AWS para uma distribuição do CloudFront
<i>region</i> -EarlyDelete-ByteHrs	Bytes por hora <sup>2</sup>	Por hora	Uso pro-rata de armazenamento de objetos excluídos do armazenamento GLACIER antes do término do compromisso mínimo de 90 dias <sup>3</sup>
<i>region</i> -EarlyDelete-SIA	Bytes por hora	Por hora	Uso pro-rata de armazenamento de objetos excluídos do STANDARD_IA antes do término do compromisso mínimo de 30 dias <sup>4</sup>

Tipo de uso	Unidades	Granularity	Descrição
<i>region</i> -EarlyDelete-ZIA	Bytes por hora	Por hora	Uso pro-rata de armazenamento de objetos excluídos do ONEZONE_IA antes do término do compromisso mínimo de 30 dias <sup>4</sup>
<i>region</i> -EarlyDelete-SIA-SmObjects	Bytes por hora	Por hora	Uso pro-rata de armazenamento de objetos pequenos (menores que 128 KB) que foram excluídos do STANDARD_IA antes que o compromisso mínimo de 30 dias terminasse <sup>4</sup>
<i>region</i> -EarlyDelete-ZIA-SmObjects	Bytes por hora	Por hora	Uso pro-rata de armazenamento de objetos pequenos (menores que 128 KB) excluídos do ONEZONE_IA antes do término do compromisso mínimo de 30 dias <sup>4</sup>
<i>region</i> -Inventory-ObjectsListed	Objetos	Por hora	O número de objetos listados em um grupo de objetos (eles são agrupados por bucket ou prefixo) com uma lista de inventários
<i>region</i> -Requests-GLACIER-Tier1	Contagem	Por hora	O número de solicitações PUT, COPY, POST, InitiateMultipartUpload, UploadPart ou CompleteMultipartUpload em objetos GLACIER
<i>region</i> -Requests-GLACIER-Tier2	Contagem	Por hora	O número de solicitações GET e todas as outras não listadas em objetos GLACIER
<i>region</i> -Requests-SIA-Tier1	Contagem	Por hora	O número de solicitações PUT, COPY, POST ou LIST em objetos do STANDARD_IA
<i>region</i> -Requests-ZIA-Tier1	Contagem	Por hora	Número de solicitações PUT, COPY, POST ou LIST em objetos do ONEZONE_IA

Tipo de uso	Unidades	Granularity	Descrição
<i>region</i> -Requests-SIA-Tier2	Contagem	Por hora	O número de solicitações GET e todas as outras solicitações non-SIA-Tier1 em objetos do STANDARD_IA
<i>region</i> -Requests-ZIA-Tier2	Contagem	Por hora	Número de solicitações GET e todas as outras solicitações non-ZIA-Tier1 em objetos ONEZONE_IA
<i>region</i> -Requests-Tier1	Contagem	Por hora	O número de solicitações PUT, COPY, POST ou LIST para STANDARD, RRS e marcas
<i>region</i> -Requests-Tier2	Contagem	Por hora	O número de solicitações GET e todas as outras solicitações non-Tier1
<i>region</i> -Requests-Tier3	Contagem	Por hora	O número de solicitações de arquivamento e de solicitações de restauração padrão do GLACIER
<i>region</i> -Requests-Tier4	Contagem	Por hora	O número de transições de ciclo de vida para o armazenamento INTELLIGENT_TIERING, STANDARD_IA ou ONEZONE_IA
<i>region</i> -Requests-Tier5	Contagem	Por hora	O número de solicitações de restauração em massa do GLACIER
<i>region</i> -Requests-Tier6	Contagem	Por hora	O número de solicitações de restauração expressas do GLACIER
<i>region</i> -Bulk-Retrieval-Bytes	Bytes	Por hora	O número de bytes de dados recuperados com solicitações em massa do GLACIER
<i>region</i> -Requests-INT-Tier1	Contagem	Por hora	O número de solicitações PUT, COPY, POST ou LIST em objetos INTELLIGENT_TIERING
<i>region</i> -Requests-INT-Tier2	Contagem	Por hora	O número de solicitações GET e todas as outras solicitações non-Tier1 para objetos INTELLIGENT_TIERING

Tipo de uso	Unidades	Granularity	Descrição
<i>region</i> -Select-Returned-INT-Bytes	Bytes	Por hora	Número de bytes de dados retornados com solicitações de seleção do armazenamento INTELLIGENT_TIERING
<i>region</i> -Select-Scanned-INT-Bytes	Bytes	Por hora	Número de bytes de dados examinados com solicitações de seleção do armazenamento INTELLIGENT_TIERING
<i>region</i> -EarlyDelete-INT	Bytes por hora	Por hora	Uso pro-rata de armazenamento de objetos excluídos do armazenamento INTELLIGENT_TIERING antes do término do compromisso mínimo de 30 dias
<i>region</i> -Monitoring-Automation-INT	Objetos	Por hora	O número de objetos exclusivos monitorados e autonivelados na classe de armazenamento INTELLIGENT_TIERING
<i>region</i> -Expedited-Retrieval-Bytes	Bytes	Por hora	O número de bytes de dados recuperados com solicitações expressas do GLACIER
<i>region</i> -Standard-Retrieval-Bytes	Bytes	Por hora	O número de bytes de dados recuperados com solicitações padrão do GLACIER
<i>region</i> -Retrieval-SIA	Bytes	Por hora	O número de bytes de dados recuperados do armazenamento do STANDARD_IA
<i>region</i> -Retrieval-ZIA	Bytes	Por hora	Número de bytes de dados recuperados do armazenamento ONEZONE_IA
<i>region</i> -StorageAnalytics-ObjCount	Objetos	Por hora	O número de objetos únicos em cada grupo de objetos (onde os eles são agrupados por bucket ou prefixo) controlado pela análise do armazenamento

Tipo de uso	Unidades	Granularity	Descrição
<i>region</i> -Select-Scanned-Bytes	Bytes	Por hora	Número de bytes de dados verificados com solicitações de seleção do armazenamento STANDARD
<i>region</i> -Select-Scanned-SIA-Bytes	Bytes	Por hora	Número de bytes de dados verificados com solicitações de seleção do armazenamento STANDARD_IA
<i>region</i> -Select-Scanned-ZIA-Bytes	Bytes	Por hora	Número de bytes de dados verificados com solicitações de seleção do armazenamento ONEZONE_IA
<i>region</i> -Select-Returned-Bytes	Bytes	Por hora	Número de bytes de dados retornados com solicitações de seleção do armazenamento STANDARD
<i>region</i> -Select-Returned-SIA-Bytes	Bytes	Por hora	Número de bytes de dados retornados com solicitações de seleção do armazenamento STANDARD_IA
<i>region</i> -Select-Returned-ZIA-Bytes	Bytes	Por hora	Número de bytes de dados retornados com solicitações de seleção do armazenamento ONEZONE_IA
<i>region</i> -TagStorage-TagHrs	Tag-Hours	Diariamente	O total de tags em todos os objetos no bucket informados por hora
<i>region</i> -TimedStorage-ByteHrs	Bytes por hora	Diariamente	O número de bytes por hora em que os dados ficaram no armazenamento do STANDARD
<i>region</i> -TimedStorage-GLACIERByteHrs	Bytes por hora	Diariamente	O número de bytes por hora em que os dados ficaram no armazenamento do GLACIER
<i>region</i> -TimedStorage-GlacierStaging	Bytes por hora	Diariamente	O número de bytes por hora em que os dados ficaram no armazenamento de preparação do GLACIER

Tipo de uso	Unidades	Granularity	Descrição
<i>region</i> -TimedStorage-INT-Freq-ByteHrs	Bytes por hora	Diariamente	O número de bytes por hora em que os dados ficaram armazenados no nível de acesso frequente do armazenamento INTELLIGENT_TIERING
<i>region</i> -TimedStorage-INT-InFreq-ByteHrs	Bytes por hora	Diariamente	O número de bytes por hora em que os dados ficaram armazenados no nível de acesso pouco frequente do armazenamento INTELLIGENT_TIERING
<i>region</i> -TimedStorage-RRS-ByteHrs	Bytes por hora	Diariamente	O número de bytes por hora em que os dados ficaram no armazenamento do Reduced Redundancy Storage (RRS)
<i>region</i> -TimedStorage-SIA-ByteHrs	Bytes por hora	Diariamente	O número de bytes por hora em que os dados ficaram no armazenamento do STANDARD_IA
<i>region</i> -TimedStorage-ZIA-ByteHrs	Bytes por hora	Diariamente	Número de bytes por hora em que os dados ficaram no armazenamento ONEZONE_IA
<i>region</i> -TimedStorage-SIA-SmObjects	Bytes por hora	Diariamente	O número de bytes por hora em que pequenos objetos (menores que 128 KB) ficaram no armazenamento do STANDARD_IA
<i>region</i> -TimedStorage-ZIA-SmObjects	Bytes por hora	Diariamente	Número de bytes por hora em que pequenos objetos (menores que 128 KB) ficaram no armazenamento ONEZONE_IA

Observações:

1. Caso você encerre uma transferência antes da conclusão, o volume de dados transferidos pode exceder o volume de dados recebidos pelo aplicativo. Essa discrepância pode ocorrer porque uma solicitação de encerramento de transferência não pode ser executada instantaneamente, e parte do volume de dados pode estar em trânsito com execução pendente da solicitação de encerramento. Esses dados em trânsito são faturados como dados de "saída" transferidos.
2. Para obter mais informações sobre a unidade de bytes por hora, consulte [Converter bytes por hora de uso para GB por mês cobrado \(p. 98\)](#).
3. Para objetos que são arquivados na classe de armazenamento GLACIER, quando eles são excluídos antes de 90 dias, existe uma cobrança pro-rata por gigabyte pelos dias restantes.

4. Para objetos que estão no armazenamento INTELLIGENT\_TIERING, STANDARD\_IA ou ONEZONE\_IA, quando são excluídos, substituídos ou movidos para uma classe de armazenamento diferente antes de 30 dias, existe uma cobrança pro-rata por gigabyte pelos dias restantes.
5. Para pequenos objetos (menores que 128 KB) que estão no armazenamento STANDARD\_IA ou ONEZONE\_IA, quando são excluídos, substituídos ou movidos para uma classe de armazenamento diferente antes de 30 dias, existe uma cobrança pro-rata por gigabyte pelos dias restantes.
6. Não há tamanho de objeto mínimo faturável para objetos na classe de armazenamento INTELLIGENT\_TIERING, mas objetos menores que 128 KB não estão qualificados para autonivelamento e sempre são cobrados segundo a taxa do nível de acesso frequente INTELLIGENT\_TIERING.

## Controle de operações em seus relatórios de uso

As operações descrevem a ação realizada em seu objeto ou bucket da AWS pelo tipo de uso especificado. As operações são indicadas por códigos autoexplicativos, como `PutObject` ou `ListBucket`. Para ver quais ações em seu bucket geraram um tipo de uso específico, use estes códigos. Ao criar um relatório de uso, você pode optar por incluir All Operations (Todas as operações) ou uma operação específica, por exemplo, `GetObject`, para relatar.

## Converter bytes por hora de uso para GB por mês cobrado

O volume de armazenamento pelo qual cobramos você a cada mês é baseado na quantidade média de armazenamento usada durante o mês. Você é cobrado por todos os dados e metadados do objeto armazenados em buckets que você criou em sua conta da AWS. Para obter mais informações sobre metadados, consulte [Chave de objeto e metadados \(p. 102\)](#).

Medimos o uso do armazenamento em "TimedStorage-ByteHrs", que são totalizados no final do mês para gerar seus custos mensais. O relatório de uso informa seu uso do armazenamento em bytes por hora e os relatórios de faturamento informam o uso do armazenamento em GB por mês. Para correlacionar seu relatório de uso com seus relatórios de faturamento, você precisa converter bytes por hora em GB por mês.

Por exemplo, se você armazena 100 GB (107.374.182.400 bytes) de dados de armazenamento STANDARD do Amazon S3 em seu bucket pelos primeiros 15 dias de março, e 100 TB (109.951.162.777.600 bytes) de dados de armazenamento STANDARD do Amazon S3 pelos últimos 16 dias de março, você terá utilizado 42.259.901.212.262.400 bytes por hora.

Primeiro, calcule o uso total de bytes por hora:

```
[107,374,182,400 bytes x 15 days x (24 hours/day)]
+ [109,951,162,777,600 bytes x 16 days x (24 hours/day)]
= 42,259,901,212,262,400 byte-hours
```

Em seguida, converta bytes por hora para GB por mês:

```
42,259,901,212,262,400 byte-hours/1,073,741,824 bytes per GB/24 hours per day
/31 days in March
=52,900 GB-Months
```

## Mais informações

- Relatório de uso da AWS para o Amazon S3 (p. 89)
- Relatórios de faturamento da AWS para Amazon S3 (p. 87)
- Definição de preço do Amazon S3

- [Perguntas frequentes do Amazon S3](#)
- [Definição de preço do Glacier](#)
- [Perguntas frequentes do Glacier](#)

## Usar tags de alocação de custos para buckets do S3

Para monitorar o custo de armazenamento ou outros critérios de projetos individuais ou grupos de projetos, rotule seus buckets do Amazon S3 usando tags de alocação de custos. Uma tag de alocação de custos é um par nome-valor que você associa a um bucket do S3. Depois que você ativa as tags de alocação de custos, a AWS as utiliza para organizar os custos de recursos em seu relatório de alocação de custo. As tags de alocação de custos só podem ser usadas para identificar buckets. Para obter informações sobre tags usadas para identificar objetos, consulte [Marcação de objetos \(p. 114\)](#).

O relatório de alocação de custos indica o uso da AWS da sua conta por categoria de produto e usuário do AWS Identity and Access Management (IAM). O relatório contém os mesmos itens de linha do relatório de faturamento detalhado (consulte [Entender seus relatórios de uso e faturamento da AWS para Amazon S3 \(p. 91\)](#)) e colunas adicionais para suas chaves de tag.

A AWS fornece dois tipos de tags de alocação de custos: uma tag gerada pela AWS e tags definidas pelo usuário. A AWS define, cria e aplica a tag `createdBy` gerada pela AWS para você depois de um evento `CreateBucket` do Amazon S3. Você define, cria e aplica tags definidas pelo usuário ao seu bucket do S3.

Você deve ativar ambos os tipos de tags separadamente no console de Gerenciamento de custos e faturamento antes que elas possam aparecer em seus relatórios de faturamento. Para obter mais informações sobre tags geradas pela AWS, consulte [Tags de alocação de custos geradas pela AWS](#). Para obter mais informações sobre como ativar tags, consulte [Usar tags de alocação de custos](#) no Guia do usuário do AWS Billing and Cost Management.

Uma tag de alocação de custos definida pelo usuário tem os seguintes componentes:

- A chave de tags. A chave de tags é o nome da tag. Por exemplo, no projeto de tags/Trinity, o projeto é a chave. A chave de tags é uma string que diferencia maiúsculas e minúsculas que pode conter de 1 a 128 caracteres Unicode.
- O valor da tag. O valor da tag é uma string obrigatória. Por exemplo, no projeto de tags/Trinity, Trinity é o valor. O valor da tag é uma string que diferencia maiúsculas e minúsculas que pode conter de 0 a 256 caracteres Unicode.

Para obter detalhes sobre os caracteres permitidos em tags definidas pelo usuário e outras restrições, consulte [Restrições de tags definidas pelo usuário](#) no Guia do usuário do AWS Billing and Cost Management.

Cada bucket do S3 tem um conjunto de tags. Um conjunto de tags contém todas as tags que são atribuídas àquele bucket. Um conjunto de tags pode conter até 10 tags ou estar vazio. As chaves podem ser únicas em um conjunto de tags, mas os valores nele não precisam ser únicos. Por exemplo, você pode ter o mesmo valor nos conjuntos de tags chamados `project/Trinity` e `cost-center/Trinity`.

Em um bucket, se você adicionar uma tag que tenha a mesma chave de uma tag existente, o novo valor substituirá o antigo.

A AWS não aplica nenhum significado semântico às suas tags. Interpretamos as tags estritamente como sequências de caracteres.

Para adicionar, listar, editar ou excluir tags, você pode usar o console do Amazon S3, a AWS Command Line Interface (AWS CLI) ou a API do Amazon S3.

Para obter mais informações sobre como criar tags, consulte o tópico apropriado:

- Para criar tags no console, consulte [Como visualizo as propriedades de um bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.
- Para criar tags usando a API do Amazon S3, consulte [Atribuição de PUT Bucket](#) no Amazon Simple Storage Service API Reference.
- Para criar tags usando a AWS CLI, consulte [put-bucket-tagging](#) no AWS CLI Command Reference.

Para obter mais informações sobre tags definidas pelo usuário, consulte [Tags de alocação de custos definidas pelo usuário](#) no Guia do usuário do AWS Billing and Cost Management.

## Mais informações

- [Usar tags de alocação de custos](#) no Guia do usuário do AWS Billing and Cost Management
- [Entender seus relatórios de uso e faturamento da AWS para Amazon S3 \(p. 91\)](#)
- [Relatórios de faturamento da AWS para Amazon S3 \(p. 87\)](#)

# Trabalho com objetos do Amazon S3

Amazon S3 é uma chave simples, um depósito de valor projetado para armazenar quantos objetos você desejar. Você armazena esses objetos em um ou mais buckets. Um objeto consiste no seguinte:

- Chave – o nome que você designa a um objeto. Você usa a chave de objeto para recuperar o objeto.

Para obter mais informações, consulte [Chave de objeto e metadados \(p. 102\)](#)

- ID de versão – em um bucket, um ID de chave e de versão identifica unicamente um objeto.

O ID de versão é uma string que o Amazon S3 gera quando você adiciona um objeto a um bucket. Para obter mais informações, consulte [Versionamento de objeto \(p. 111\)](#).

- Valor – o conteúdo que você está armazenando.

Um valor de objeto pode ser qualquer sequência de bytes. Objetos podem variar em tamanho de zero a 5 TB. Para obter mais informações, consulte [Upload de objetos \(p. 175\)](#).

- Metadados – um par de nome-valor com o qual você pode armazenar informações sobre o objeto.

É possível atribuir metadados, chamados de metadados definidos pelo usuário, aos seus objetos no Amazon S3. O Amazon S3 também atribui metadados do sistema a esses objetos, que ele usa para gerenciar objetos. Para obter mais informações, consulte [Chave de objeto e metadados \(p. 102\)](#).

- Sub-recursos – o Amazon S3 usa o mecanismo de sub-recurso para armazenar informações adicionais específicas do objeto.

Como os sub-recursos são subordinados aos objetos, eles estão sempre associados com qualquer outra entidade, tal como um objeto ou um bucket. Para obter mais informações, consulte [Sub-recursos de objeto \(p. 111\)](#).

- Informações de controle de acesso – você pode controlar o acesso aos objetos que você armazena em Amazon S3.

Amazon S3 suporta o controle de acesso baseado em recursos, tal como uma Access Control List (ACL, lista de controle de acesso) e políticas de bucket, e controle de acesso de dados baseados no usuário. Para obter mais informações, consulte [Gerenciamento de permissões de acesso aos recursos do Amazon S3 \(p. 282\)](#).

Para obter mais informações sobre objetos, consulte as seguintes seções. Seus recursos do Amazon S3 (por exemplo, buckets e objetos) são privados por padrão. É necessário conceder permissão expressa para que outras pessoas acessem esses recursos. Por exemplo, você pode querer compartilhar um vídeo ou uma foto armazenados em seu bucket Amazon S3 em seu site. Isso funcionará somente se você tornar o objeto público ou usar um pre-signed URL em seu site. Para obter mais informações sobre compartilhamento de objetos, consulte [Compartilhe um objeto \(p. 172\)](#).

## Tópicos

- [Chave de objeto e metadados \(p. 102\)](#)
- [Classes de armazenamento \(p. 107\)](#)
- [Sub-recursos de objeto \(p. 111\)](#)
- [Versionamento de objeto \(p. 111\)](#)
- [Marcação de objetos \(p. 114\)](#)
- [Gerenciamento do ciclo de vida de objetos \(p. 122\)](#)

- Cross-Origin Resource Sharing (CORS, Compartilhamento de recursos de origem cruzada) (p. 156)
- Operações em objetos (p. 165)

## Chave de objeto e metadados

Cada objeto do Amazon S3 tem dados, uma chave e metadados. A chave de objeto (ou nome da chave) identifica, unicamente, o objeto em um bucket. Metadados de objeto são um conjunto de pares nome-valor. Você pode definir metadados de objeto no momento em que fizer seu upload. Após fazer upload do objeto, você não pode modificar seus metadados. A única forma de modificar metadados de objeto é fazer uma cópia do objeto e definir os metadados.

### Tópicos

- [Chaves de objeto \(p. 102\)](#)
- [Metadados do objeto \(p. 104\)](#)

## Chaves de objeto

Quando você cria um objeto, especifica o nome da chave que, exclusivamente, identifica o objeto no bucket. Por exemplo, no console do Amazon S3 (consulte [Console de Gerenciamento da AWS](#)), quando você destaca um bucket, aparece uma lista de objetos em seu bucket. Esses nomes são as chaves de objeto. O nome para uma chave é uma sequência de caracteres Unicode cuja codificação UTF-8 é de, no máximo, 1024 bytes de comprimento.

O modelo de dados do Amazon S3 é uma estrutura plana: você cria um bucket e o bucket armazena objetos. Não há hierarquia de subbuckets ou de subpastas; contudo, você pode pressupor a hierarquia lógica, usando prefixos e delimitadores de nome de chave como o console do Amazon S3 faz. O console do Amazon S3 suporta o conceito de pastas. Vamos supor que seu bucket (`admin-created`) tenha quatro objetos com as seguintes chaves de objeto:

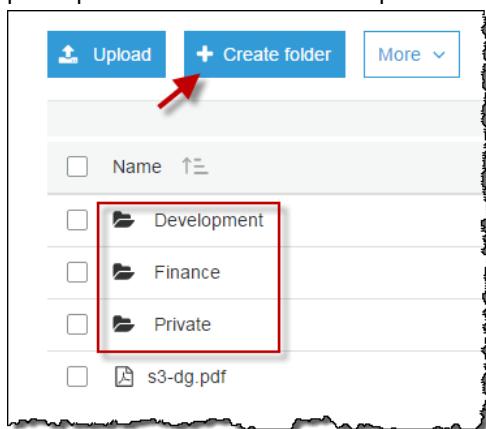
`Development/Projects1.xls`

`Finance/statement1.pdf`

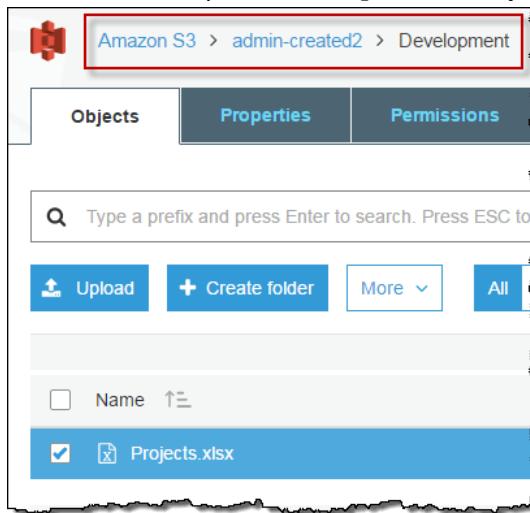
`Private/taxdocument.pdf`

`s3-dg.pdf`

O console usa prefixos de nome de chave (`Development/`, `Finance/` e `Private/`) e o delimitador (“`/`”) para apresentar uma estrutura de pasta como mostrada:



A chave s3-dg.pdf não tem um prefixo, de modo que seu objeto aparece diretamente no nível da raiz do bucket. Ao abrir a pasta Development/, o objeto Projects.xlsx é exibido.



#### Note

O Amazon S3 oferece suporte a buckets e objetos. Além disso, não há nenhuma hierarquia no Amazon S3. No entanto, os prefixes e os delimitadores em um nome de chave de objeto permitem que o console do Amazon S3 e os SDKs da AWS infiram a hierarquia e apresentem o conceito de pastas.

## Diretrizes de nomeação de chave de objeto

Você pode usar qualquer caractere UTF-8 em um nome de chave de objeto. No entanto, o uso de determinados caracteres em nomes de chave pode causar problemas com alguns aplicativos e protocolos. As seguintes diretrizes ajudam você a maximizar a conformidade com DNS, caracteres seguros da web, parsers de XML e outras APIs.

### Caracteres seguros

Os seguintes conjuntos de caracteres são, geralmente, confiáveis para uso em nomes de chave:

#### Caracteres alfanuméricos

- 0-9
- a-z
- A-Z

#### Caracteres especiais

- !
- -
- \_
- .
- \*
- '
- (
- )

Os seguintes são exemplos de nomes de chave válidos:

- 4my-organization
- my.great\_photos-2014/jan/myvacation.jpg
- videos/2014/birthday/video1.wmv

## Caracteres que podem exigir tratamento especial

Os caracteres a seguir em um nome de chave podem exigir tratamento adicional do código e, provavelmente, precisarão ser criptografados por URL ou referenciados como HEX. Alguns desses caracteres não são imprimíveis, e seu navegador pode não reconhecê-los, o que também exigirá tratamento especial:

- Sinal tipográfico ("&")
- Dólar ("\$")
- Caracteres ASCII variam de 00–1F em hexadecimal (0–31 decimal) e 7F (127 decimal)
- Símbolo 'Arroba' ("@")
- Igual a ("=")
- Ponto-e-vírgula (";")
- Dois pontos (":")
- Mais ("+")
- Espaço – Sequências significativas de espaços podem ser perdidas em alguns usos (especialmente múltiplos espaços)
- Vírgula (",")
- Ponto de interrogação ("?")

## Caracteres a serem evitados

Evite os caracteres a seguir em um nome de chave devido ao tratamento especial significativo necessário para consistência em todos os aplicativos.

- Barra invertida ("\")
- Chave esquerda ("{"})
- Caracteres ASCII não imprimíveis (128–255 caracteres decimais)
- Circunflexo (^")
- Chave direita ("}")
- Caractere de porcentagem ("%")
- Crase (`")
- Colchete direito ("]")
- Pontos de interrogação
- Sinal de maior (>")
- Colchete esquerdo ("[")
- Til (~")
- Sinal de menor (<")
- Caractere de libra (#")
- Barra vertical ("|")

## Metadados do objeto

Há dois tipos de metadados: metadados de sistema e metadados definidos pelo usuário.

## Metadados definidos por sistema

Para cada objeto armazenado em um bucket, o Amazon S3 mantém um conjunto de metadados do sistema. O Amazon S3 processa esses metadados do sistema conforme necessário. Por exemplo, o Amazon S3 mantém a data de criação e o tamanho dos metadados e usa estas informações como parte do gerenciamento do objeto.

Existem duas categorias de metadados de sistema:

1. Metadados tais como a data de criação de objeto são controlados pelo sistema onde apenas o Amazon S3 pode modificar o valor.
2. Outros metadados de sistema, como a classe de armazenamento configurada para o objeto e se o objeto tem criptografia habilitada no lado do servidor, são exemplos cujos valores são controlados por você. Se o bucket está configurado como um site, você pode querer redirecionar uma solicitação de página para outra página ou para um URL externo. Nesse caso, uma página da Web é um objeto em seu bucket. O Amazon S3 armazena o valor de redirecionamento de página como metadados do sistema cujo valor é controlado por você.

Ao criar objetos, você pode configurar os valores desses itens de metadados de sistema ou atualizar os valores quando necessário. Para obter mais informações sobre classes de armazenamento, consulte [Classes de armazenamento \(p. 107\)](#). Para obter mais informações sobre criptografia no lado do servidor, consulte [Proteção de dados usando criptografia \(p. 409\)](#).

A tabela a seguir fornece uma lista dos metadados definidos por sistema e se você pode atualizá-los.

Nome	Descrição	O usuário pode modificar o valor?
Data	Data e hora atual.	Não
Content-Length	Tamanho de objeto em bytes.	Não
Última modificação	Data de criação do objeto ou data da última modificação, o que aconteceu por último.	Não
Conteúdo-MD5	O resumo MD5 de 128 bits com codificação base64 do objeto.	Não
x-amz-server-side-encryption	Indica se a criptografia do lado do servidor está habilitada para o objeto e se essa criptografia é do AWS Key Management Service (SSE-KMS) ou da criptografia gerenciada pela AWS (SSE-S3). Para obter mais informações, consulte <a href="#">Proteção de dados usando criptografia no lado do servidor (p. 410)</a> .	Sim
x-amz-version-id	Versão do objeto. Quando você permite o versionamento em um bucket, o Amazon S3 atribui um número de versão aos objetos adicionados ao bucket. Para obter mais informações, consulte <a href="#">Usar versionamento (p. 448)</a> .	Não
x-amz-delete-marker	Em um bucket com o versionamento habilitado, o marcador booleano indica se o objeto é um marcador de exclusão.	Não
x-amz-storage-class	Classe de armazenamento usada para armazenamento do objeto. Para obter mais informações, consulte <a href="#">Classes de armazenamento (p. 107)</a> .	Sim

Nome	Descrição	O usuário pode modificar o valor?
x-amz-website-redirect-location	Redireciona solicitações do objeto associado para outro objeto no mesmo bucket ou um URL externo. Para obter mais informações, consulte <a href="#">(Opcional) Configuração de um redirecionamento de uma página da web (p. 502)</a> .	Sim
x-amz-server-side-encryption-aws-kms-key-id	Se a x-amz-server-side-encryption estiver presente e tiver o valor de aws:kms, isso indicará o ID da chave mestra de criptografia do AWS Key Management Service (AWS KMS) que foi usada para o objeto.	Sim
x-amz-server-side-encryption-customer-algorithm	Indica se a criptografia do lado do servidor com as chaves fornecidas pelo cliente (SSE-C) está habilitada. Para obter mais informações, consulte <a href="#">Proteção de dados usando criptografia de servidor com chaves de criptografia fornecidas pelo cliente (SSE-C) (p. 425)</a> .	Sim

## Metadados definidos pelo usuário

Ao fazer upload de um objeto, você também pode atribuir metadados ao objeto. Você fornece essas informações opcionais como um par de nome-valor (valor-chave) quando envia uma solicitação PUT ou POST para criar o objeto. Ao fazer upload de objetos usando a API REST, os nomes de metadados opcionais definidos pelo usuário devem começar com "x-amz-meta-", para diferenciá-los de outros cabeçalhos HTTP. Quando você recupera o objeto usando a API REST, o prefixo é retornado. Ao fazer upload de objetos usando a API SOAP, o prefixo não é obrigatório. Quando você recupera o objeto usando SOAP API, o prefixo é removido, independentemente da API que você usou para fazer upload do objeto.

### Note

O suporte de SOAP via HTTP está obsoleto, mas continua disponível via HTTPS. Novos recursos do Amazon S3 não serão compatíveis com o SOAP. Recomendamos que você use o REST API ou os SDKs da AWS.

Quando os metadados são recuperados por API REST, o Amazon S3 combina os cabeçalhos que têm o mesmo nome (ignorando maiúsculas) em uma lista delimitada por vírgula. Se alguns metadados contêm caracteres não imprimíveis, eles não são retornados. Em vez disso, o cabeçalho x-amz-missing-meta é retornado com o valor do número de entradas de metadados não imprimíveis.

Metadados definidos pelo usuário são um conjunto de pares de chave e valor. O Amazon S3 armazena chaves de metadados definidos pelo usuário em letras minúsculas. Cada par de chave/valor deve estar em conformidade com US-ASCII ao usar REST e com UTF-8 ao usar SOAP ou uploads baseados em navegadores por meio de POST.

### Note

O cabeçalho da solicitação PUT é limitado a 8 KB. No cabeçalho da solicitação PUT, os metadados definidos pelo usuário são limitados a 2 KB. O tamanho de metadados definidos pelo usuário é medido pela soma do número de bytes na codificação UTF-8 de cada chave e valor.

Para obter mais informações sobre como adicionar metadados ao objeto após fazer o upload dele, consulte [Como posso adicionar metadados a um objeto do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

## Classes de armazenamento

Cada objeto em Amazon S3 tem uma classe de armazenamento associada a ela. Por exemplo, se você lista os objetos em um bucket do S3, o console mostra a classe de armazenamento de todos os objetos na lista.

	Name	Last modified	Size	Storage class
<input type="checkbox"/>	notice.pdf	Jul 13, 2016 7:19:13 PM GMT-0700	175.5 KB	Standard
<input type="checkbox"/>	screen-shot.png	Apr 2, 2018 6:47:22 PM GMT-0700	109.8 KB	Standard-IA
<input type="checkbox"/>	screen-shot3.png	Apr 2, 2018 7:08:32 PM GMT-0700	109.8 KB	Standard-IA

O Amazon S3 oferece uma variedade de classes de armazenamento de objetos que você armazena. Escolha uma classe de acordo com seu cenário de caso de uso e dos requisitos de acesso de desempenho. Todas essas classes de armazenamento oferecem alta durabilidade.

### Tópicos

- [Classes de armazenamento de objetos acessados com frequência \(p. 107\)](#)
- [Classe de armazenamento que otimiza automaticamente objetos muito e pouco acessados \(p. 108\)](#)
- [Classes de armazenamento de objetos acessados com pouca frequência \(p. 108\)](#)
- [Comparar as classes de armazenamento do Amazon S3 \(p. 110\)](#)
- [Configurar a classe de armazenamento de um objeto \(p. 110\)](#)

## Classes de armazenamento de objetos acessados com frequência

Para casos de uso nos quais o desempenho é importante (exigem tempo de acesso de milissegundos) e dados acessados com frequência, o Amazon S3 fornece as seguintes classes de armazenamento:

- **STANDARD**— a classe de armazenamento padrão. Se você não especificar a classe de armazenamento ao fazer upload de um objeto, o Amazon S3 atribuirá a classe STANDARD.
- **REDUCED\_REDUNDANCY** — a classe de armazenamento Reduced Redundancy Storage (RRS) foi criada para dados reproduzíveis não críticos que podem ser armazenados em níveis de redundância menores do que a classe STANDARD.

### Important

Não recomendamos o uso dessa classe de armazenamento. A classe de armazenamento STANDARD é mais econômica.

Para durabilidade, os objetos RRS têm uma perda anual média prevista de 0,01%. Se um objeto RRS for perdido, quando forem feitas solicitações a ele, o Amazon S3 retornará um erro 405.

## Classe de armazenamento que otimiza automaticamente objetos muito e pouco acessados

A classe de armazenamento INTELLIGENT\_TIERING foi projetada para otimizar os custos de armazenamento movendo automaticamente os dados para o nível de acesso ao armazenamento mais econômico, sem impacto no desempenho ou sobrecarga operacional. INTELLIGENT\_TIERING oferece uma economia de custo automática movendo dados em um nível de objeto granular entre dois níveis de acesso, um nível de acesso frequente e um nível de acesso pouco frequente econômico, quando os padrões de acesso mudam. A classe de armazenamento INTELLIGENT\_TIERING é ideal caso você queira otimizar custos de armazenamento automaticamente para dados duradouros quando os padrões de acesso são desconhecidos ou imprevisíveis.

A classe INTELLIGENT\_TIERING armazena os objetos em duas camadas de acesso: uma otimizada para acesso frequente e outra camada de baixo custo otimizada para dados pouco acessados. Por uma pequena taxa mensal de automação e monitoramento por objeto, o Amazon S3 monitora os padrões de acesso dos objetos na classe de armazenamento INTELLIGENT\_TIERING e move os objetos que não foram acessados por 30 dias consecutivos para o nível de acesso infrequente. Não há taxas de recuperação durante o uso da classe de armazenamento INTELLIGENT\_TIERING. Se um objeto no nível de acesso infrequente for acessado, ele será automaticamente movido de volta para o nível de acesso frequente. Nenhuma taxa adicional de camada se aplica quando objetos são movidos entre as camadas de acesso na classe de armazenamento INTELLIGENT\_TIERING.

### Note

A classe de armazenamento INTELLIGENT\_TIERING é indicada para objetos maiores que 128 KB que você pretende armazenar por pelo menos 30 dias. Se o tamanho de um objeto for menor que 128 KB, ele não estará qualificado para o nivelamento automático. Os objetos menores podem ser armazenados, mas são sempre cobrados com base nas taxas do nível de acesso frequente na classe de armazenamento INTELLIGENT\_TIERING. Se excluir um objeto antes do período mínimo de 30 dias, você será cobrado por 30 dias. Para obter informações sobre a definição de preço, consulte [Definição de preço do Amazon S3](#).

## Classes de armazenamento de objetos acessados com pouca frequência

As classes de armazenamento STANDARD\_IA e ONEZONE\_IA foram desenvolvidas para dados duradouros e acessados com pouca frequência. (IA significa, em inglês, acesso pouco frequente.) Os objetos STANDARD\_IA e ONEZONE\_IA estão disponíveis para acesso de milissegundos (semelhante à classe de armazenamento STANDARD). O Amazon S3 cobra um valor de recuperação para esses valores, portanto, eles são mais adequados para dados raramente acessados. Para obter informações sobre a definição de preço, consulte [Definição de preço do Amazon S3](#).

Por exemplo, é possível escolher as classes de armazenamento STANDARD\_IA e ONEZONE\_IA:

- Para armazenar backups.
- Para dados mais antigos acessados com pouca frequência, mas que ainda exigem acesso de milissegundos. Por exemplo, ao fazer upload de dados, é possível escolher a classe de armazenamento STANDARD e usar a configuração de ciclo de vida para solicitar que o Amazon S3 transfira os objetos para a classe STANDARD\_IA ou ONEZONE\_IA. Para obter mais informações sobre gerenciamento de ciclo de vida, consulte [Gerenciamento do ciclo de vida de objetos \(p. 122\)](#).

#### Note

As classes de armazenamento STANDARD\_IA e ONEZONE\_IA são adequadas para objetos maiores que 128 KB que você deseja armazenar por pelo menos 30 dias. Se um objeto for menor que 128 KB, o Amazon S3 cobrará por 128 KB. Se excluir um objeto antes do período mínimo de 30 dias, você será cobrado por 30 dias. Para obter informações sobre a definição de preço, consulte [Definição de preço do Amazon S3](#).

As diferenças entre essas classes de armazenamento são:

- STANDARD\_IA — o Amazon S3 armazena dados de objeto de maneira redundante em várias zonas de disponibilidade separadas geograficamente (de maneira semelhante à classe de armazenamento STANDARD). Os objetos STANDARD\_IA são resistentes à perda de uma zona de disponibilidade. Essa classe de armazenamento oferece maior disponibilidade, durabilidade e resiliência que a classe ONEZONE\_IA.
- ONEZONE\_IA — o Amazon S3 armazena dados de objeto em apenas uma zona de disponibilidade, e isso a torna menos cara que a classe STANDARD\_IA. No entanto, os dados não são resilientes à perda física da zona de disponibilidade resultante de desastres, como terremotos e inundações. A classe de armazenamento ONEZONE\_IA é tão durável quanto a classe STANDARD\_IA, mas é menos disponível e resistente. Para uma comparação de durabilidade e disponibilidade das classes de armazenamento, consulte a tabela Durabilidade e disponibilidade, no fim desta seção. Para obter informações sobre preços, consulte [Definição de preço do Amazon S3](#).

Recomendamos o seguinte:

- STANDARD\_IA: use para seu cópia principal ou única de dados, que não pode ser recriada.
- ONEZONE\_IA: use se você puder recriar os dados em caso de falha da zona de disponibilidade, e para réplicas de objeto ao configurar a replicação entre regiões (CRR).

## Classe de armazenamento GLACIER

A GLACIER classe de armazenamento é adequada para dados arquivados em que o acesso aos dados é pouco frequente. Oferece a mesma durabilidade e resiliência que a classe de armazenamento STANDARD.

Defina a classe de armazenamento de um objeto como GLACIER da mesma maneira que faz para outras classes de armazenamento conforme descrito na seção [Configurar a classe de armazenamento de um objeto \(p. 110\)](#). Porém, os objetos de arquivo GLACIER não estão disponíveis para acesso em tempo real. Você deve primeiro recuperar os objetos GLACIER antes de acessá-los (os objetos STANDARD, RRS, STANDARD\_IA e ONEZONE\_IA estão disponíveis para acesso a qualquer momento). Para obter mais informações, consulte [Restaurar objetos arquivados \(p. 259\)](#).

#### Important

Ao selecionar a classe de armazenamento do GLACIER, o Amazon S3 usa o serviço Glacier de baixo custo para armazenar objetos. Embora os objetos sejam armazenados no Glacier, eles continuam sendo objetos do Amazon S3 que você gerencia no Amazon S3, e não é possível acessá-los diretamente por meio do Glacier.

Para saber mais sobre o serviço Glacier, consulte o [Guia do desenvolvedor do Amazon S3 Glacier](#).

## Comparar as classes de armazenamento do Amazon S3

A tabela a seguir compara as classes de armazenamento.

Storage Class	Designed for	Durability (designed for)	Availability (designed for)	Availability Zones	Min storage duration	Min billable object size	Other Considerations
STANDARD	Frequently accessed data	99.999999999%	99.99%	>= 3	None	None	None
STANDARD_IA	Long-lived, infrequently accessed data	99.999999999%	99.9%	>= 3	30 days	128 KB	Per GB retrieval fees apply.
INTELLIGENT_TIERING	Long-lived data with changing or unknown access patterns	99.999999999%	99.9%	>= 3	30 days	None	Monitoring and automation fees per object apply. No retrieval fees.
ONEZONE_IA	Long-lived, infrequently accessed, non-critical data	99.999999999%	99.5%	1	30 days	128 KB	Per GB retrieval fees apply. Not resilient to the loss of the Availability Zone.
GLACIER	Long-term data archiving with retrieval times ranging from minutes to hours	99.999999999%	99.99% (after you restore objects)	>= 3	90 days	None	Per GB retrieval fees apply. You must first restore archived objects before you can access them. For more information, see <a href="#">Restoring Archived Objects</a> .
RRS (Not recommended)	Frequently accessed, non-critical data	99.99%	99.99%	>= 3	None	None	None

Todas as classes de armazenamento, exceto a ONEZONE\_IA, são desenvolvidas para serem resilientes à perda simultânea de dados completos em uma única zona de disponibilidade e à perda parcial em outra zona de disponibilidade.

Considere o preço, além dos requisitos de desempenho do cenário do seu aplicativo. Para obter informações sobre os preços das classes de armazenamento, consulte [Definição de preço do Amazon S3](#).

## Configurar a classe de armazenamento de um objeto

As APIs do Amazon S3 são compatíveis com a configuração (ou atualização) da classe de armazenamento de objetos, da seguinte forma:

- Ao criar um objeto, é possível especificar a classe de armazenamento dele. Por exemplo, ao criar objetos usando as APIs [PUT Object](#), [POST Object](#) e [Initiate Multipart Upload](#), adicione o cabeçalho de solicitação `x-amz-storage-class` para especificar uma classe de armazenamento. Se você não adicionar esse cabeçalho, o Amazon S3 usará a classe de armazenamento padrão, STANDARD.
- Também é possível alterar a classe de armazenamento de um objeto já armazenado no Amazon S3 fazendo uma cópia desse objeto usando a API [PUT Object - Copy](#). Copie o objeto no mesmo bucket usando o mesmo nome de chave e especifique os cabeçalhos de solicitação da seguinte forma:
  - Defina o cabeçalho `x-amz-metadata-directive` como COPY.
  - Defina `x-amz-storage-class` como a classe de armazenamento desejada.

Em um bucket com versionamento habilitado, não é possível alterar a classe de armazenamento de uma versão específica de um objeto. Quando você a copia, o Amazon S3 fornece um novo ID de versão.

- É possível direcionar o Amazon S3 para alterar a classe de armazenamento de objetos adicionando a configuração do ciclo de vida em um bucket. Para obter mais informações, consulte [Gerenciamento do ciclo de vida de objetos \(p. 122\)](#).

- Ao definir uma configuração Cross-Region Replication (CRR – Replicação entre regiões), você pode definir a classe de armazenamento para objetos replicados. Para obter mais informações, consulte [Visão geral da configuração da replicação \(p. 548\)](#).

Para criar e atualizar classes de armazenamento de objetos, é possível usar o console do Amazon S3, AWS SDKs ou a AWS Command Line Interface (AWS CLI). Todos eles usam APIs do Amazon S3 para enviar solicitações para o Amazon S3.

## Sub-recursos de objeto

O Amazon S3 define um conjunto de sub-recursos associados a buckets e objetos. Sub-recursos são subordinados aos objetos; isto é, não existem por si mesmos; sempre estão associados a alguma outra entidade tal como um objeto ou um bucket.

A tabela a seguir lista os sub-recursos associados a objetos do Amazon S3.

Sub-recurso	Descrição
acl	Contém uma lista de concessões que identifica os concessionários e permissões concedidas. Quando você cria um objeto, o acl identifica o proprietário do objeto como tendo total controle sobre o objeto. Você pode recuperar a ACL de um objeto ou substituí-la por uma lista atualizada de concessões. Qualquer atualização para um ACL requer que você substitua o ACL existente. Para obter mais informações sobre ACLs, consulte <a href="#">Gerenciar o acesso com ACLs (p. 390)</a> .
torrent	O Amazon S3 oferece suporte ao protocolo BitTorrent. O Amazon S3 usa o sub-recurso torrent para retornar o arquivo de torrent associado ao objeto específico. Para recuperar um arquivo de torrent, especifique o sub-recurso torrent em sua solicitação GET. O Amazon S3 cria um arquivo de torrent e o retorna. Você só pode recuperar o sub-recurso torrent; não pode criar, atualizar ou excluir o sub-recurso torrent. Para obter mais informações, consulte <a href="#">Usar o BitTorrent com o Amazon S3 (p. 614)</a> .

## Versionamento de objeto

Use o versionamento para manter várias versões de um objeto em um bucket. Por exemplo, é possível armazenar `my-image.jpg` (versão 111111) e `my-image.jpg` (versão 222222) em um único bucket. O versionamento protege contra as consequências de substituições e exclusões não intencionais. Também é possível usar o versionamento para arquivar objetos a fim de ter acesso às versões anteriores.

### Note

A API SOAP não suporta versionamento. O suporte de SOAP via HTTP está obsoleto, mas continua disponível via HTTPS. Os novos recursos do Amazon S3 não são compatíveis com SOAP.

Para personalizar sua abordagem de retenção de dados e controlar os custos de armazenamento, use o versionamento de objetos com [Gerenciamento do ciclo de vida de objetos \(p. 122\)](#). Para obter informações sobre a utilização de políticas de ciclo de vida usando o Console de gerenciamento da AWS, consulte [Como crio uma política de ciclo de vida para um bucket S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

Se você tem uma política de ciclo de vida de expiração do objeto em seu bucket sem versão e quer manter o mesmo comportamento de exclusão permanente quando ativar o controle de versão, precisará adicionar uma política de expiração de versão desatualizada. A política de expiração do ciclo de vida gerenciará as exclusões de versões desatualizadas de objeto no bucket habilitado para versão. (Um bucket habilitado para versão mantém uma versão atual e zero ou mais versões desatualizadas de objeto.)

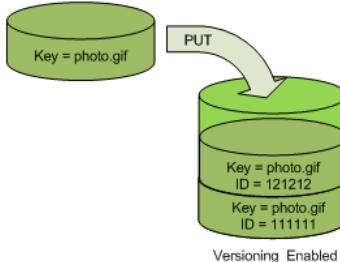
Você deve, explicitamente, habilitar o versionamento em seu bucket. Por padrão, o versionamento é desabilitado. Independentemente de você ter habilitado o versionamento, cada objeto em seu bucket terá um ID de versão. Se você não tiver habilitado o versionamento, o Amazon S3 definirá o valor do ID da versão como nulo. Se você habilitou o versionamento, o Amazon S3 atribui um valor de ID de versão único para o objeto. Ao habilitar o versionamento em um bucket, os objetos já armazenados nele permanecerão inalterados. Os IDs de versão (nulos), o conteúdo e as permissões continuarão os mesmos.

A habilitação e a suspensão do versionamento são feitas no nível do bucket. Quando você habilita o versionamento para um bucket, todos os objetos adicionados a ele terão um ID exclusivo de versão. Os IDs exclusivos de versão são gerados aleatoriamente, com codificação Unicode e UTF-8, e prontos para URL, com strings opacas de, no máximo, 1.024 bytes de extensão. Um exemplo de ID de versão é `3/L4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY+MTRCx3vjVBH40Nr8X8gdRQBpUMLUo`. Somente o Amazon S3 gera IDs de versão. Eles não podem ser editados.

#### Note

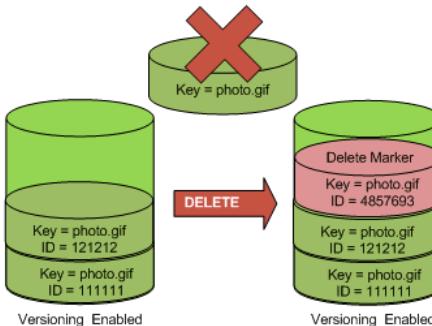
Para manter a simplicidade, usaremos IDs muito mais curtos em todos os exemplos.

Quando você `PUT` um objeto em um bucket com versionamento ativado, a versão desatualizada não é substituída. A figura a seguir mostra que, quando uma nova versão de `photo.gif` é `PUT` em um bucket que já contém um objeto com o mesmo nome, o objeto exclusivo (ID = 111111) permanece no bucket, o Amazon S3 gera um ID de nova versão (121212) e adiciona a versão mais recente ao bucket.

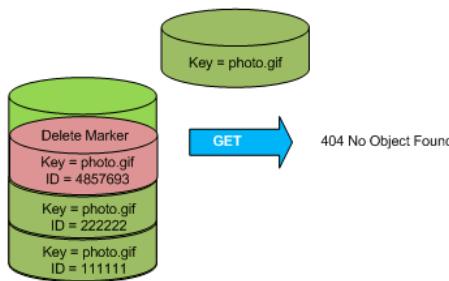


Essa funcionalidade impede que você acidentalmente substitua ou exclua objetos e permite que você recupere uma versão anterior de um objeto.

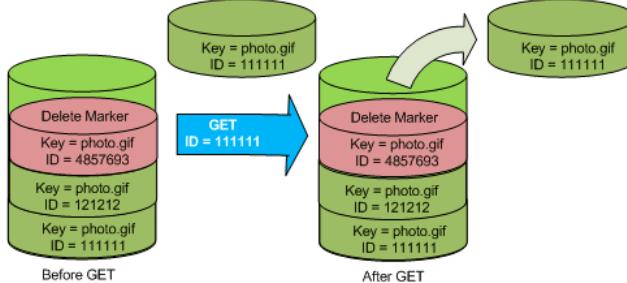
Quando você `DELETE` um objeto, todas as versões permanecem no bucket e o Amazon S3 insere um marcador de exclusão, conforme exibido na figura a seguir.



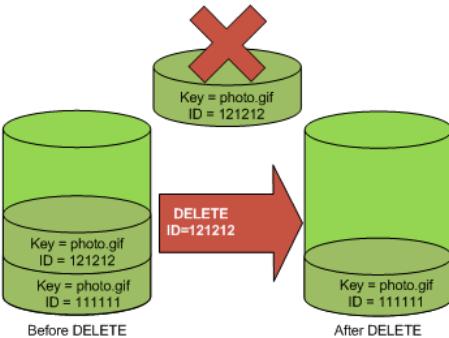
O marcador de exclusão torna-se a versão atual de objeto. Por padrão, o `GET` requisita a recuperação da versão armazenada mais recente. A execução de uma solicitação `GET Object` simples quando a versão atual é um marcador de exclusão retorna um erro `404 Not Found`, conforme exibido na figura a seguir.



Você pode, contudo, fazer uma solicitação GET de uma versão desatualizada de um objeto por especificar seu ID de versão. Na figura a seguir, nós GET uma versão de objeto específica, 111111. O Amazon S3 retorna essa versão de objeto, embora não seja a versão atual.



Você pode excluir permanentemente um objeto, especificando a versão que você deseja excluir. Somente o proprietário de um bucket do Amazon S3 pode excluir uma versão permanentemente. A figura a seguir mostra como DELETE `versionId`permanentemente, um objeto de um bucket e o Amazon S3 não insere um marcador de exclusão.



Você pode obter segurança adicional configurando um bucket para permitir a exclusão de MFA (autenticação multifator). Quando você faz isso, o proprietário do bucket precisa incluir dois formulários de autenticação em qualquer solicitação para excluir uma versão ou modificar o estado de versionamento do bucket. Para obter mais informações, consulte [Exclusão MFA \(p. 449\)](#).

#### Important

Se você perceber um aumento significativo do número de respostas HTTP 503 recebidas com lentidão do Amazon S3 de solicitações PUT ou DELETE de objetos a um bucket que tenha versionamento habilitado, talvez tenha um ou mais objetos no bucket para os quais há milhões de versões. Para obter mais informações, consulte [Solução de problemas do Amazon S3 \(p. 621\)](#).

Para obter mais informações, consulte [Usar versionamento \(p. 448\)](#).

## Marcação de objetos

Use a marcação de objetos para classificar o armazenamento. Cada tag é um par de chave-valor. Considere os seguintes exemplos de marcação:

- Suponha que um objeto contenha dados protegidos de informações de saúde (PHI). Você pode marcar o objeto usando o seguinte par de chave-valor, como mostrado a seguir:

```
PHI=True
```

ou

```
Classification=PHI
```

- Suponha que você armazene arquivos de projeto em seu bucket do S3. Você pode marcar esses objetos com uma chave chamada Project e um valor, como mostrado a seguir:

```
Project=Blue
```

- Você pode adicionar várias tags a um objeto, como mostrado a seguir:

```
Project=x  
Classification=confidential
```

Você pode adicionar tags a objetos novos ao fazer upload deles ou pode adicioná-las aos objetos existentes. Observe o seguinte:

- Você pode associar até 10 tags a um objeto. As tags associadas a um objeto devem ter chaves de tag exclusivas.
- Um chave de tag pode ter até 128 caracteres Unicode e os valores de tag podem ter até 256 caracteres Unicode.
- As chaves e os valores diferenciam letras maiúsculas de minúsculas.

Os prefixos de nome de chave de objeto também permitem classificar o armazenamento, mas a classificação com base em prefixo tem uma dimensão. Considere os seguintes nomes de chave de objeto:

```
photos/photo1.jpg  
project/projectx/document.pdf  
project/projecty/document2.pdf
```

Esses nomes de chave têm os prefixos photos/, project/projectx/ e project/projecty/. Esses prefixos habilitam a classificação de uma dimensão. Isto é, tudo que tiver um prefixo pertencerá a uma categoria. Por exemplo, o prefixo projeto/projetox identifica todos os documentos relacionados ao projeto x.

Com a marcação, você agora tem outra dimensão. Se você quiser que photo1 esteja na categoria projeto x, poderá marcar o objeto conforme necessário. Além de classificação de dados, a marcação oferece outros benefícios. Por exemplo,

- As tags de objeto permitem ter controle de acesso de permissões. Por exemplo, você pode conceder a um usuário do IAM permissões para ler somente objetos com tags específicas.
- As tags de objeto permitem o gerenciamento de ciclo de vida do objeto em que você pode especificar o filtro com base em tag, além do prefixo de nome da chave, em uma regra de ciclo de vida.

- Ao usar a análise do Amazon S3, você pode configurar filtros para agrupar objetos para análise por tags de objeto, prefixo de nome da chave ou ambos, prefixo e tags.
- Você também pode personalizar métricas do Amazon CloudWatch para exibir informações por filtros de tag específicos. As seguintes seções fornecem detalhes.

#### Important

Quando é aceitável usar tags para identificar objetos que contêm dados confidenciais (como informações de identificação pessoal (PII) ou informações de saúde protegidas (PHI)), as tags não devem conter informações confidenciais.

## Operações de API relacionadas à marcação de objetos

O Amazon S3 oferece suporte às seguintes operações de API que são especificamente para marcação de objetos:

### Operações de API de objeto

- [Atribuição de tags de objeto PUT](#) – Substitui tags em um objeto. Especifique tags no corpo de solicitação. Há dois cenários distintos de gerenciamento de tags de objeto usando essa API.
  - O objeto não tem tags – Usando essa API, você pode adicionar um conjunto de tags a um objeto (o objeto não tem nenhuma tag anterior).
  - O objeto tem um conjunto de tags existentes – Para modificar o conjunto de tags existente, você deve primeiro recuperar o conjunto de tags existente, modificá-lo no lado do cliente e usar essa API para substituir o conjunto de tags. Se você enviar essa solicitação com o conjunto de tags vazio, o S3 excluirá o conjunto de tags existente no objeto.
- [Atribuição de tags de objeto GET](#) – Retorna o conjunto de tags associado a um objeto. O Amazon S3 retorna tags de objeto no corpo da resposta.
- [Atribuição de tags de objeto DELETE](#) – Exclui o conjunto de tags associado a um objeto.

### Outras operações de API que oferecem suporte à marcação

- [Objeto PUT e Iniciar multipart upload](#)– Você pode especificar tags ao criar objetos. Especifique tags usando o cabeçalho de solicitação `x-amz-tagging`.
- [Objeto GET](#) – Em vez de retornar o conjunto de tags, o Amazon S3 retorna a contagem de tags de objeto no cabeçalho `x-amz-tag-count` (somente se o solicitante tiver permissões para ler tags) porque o tamanho de resposta do cabeçalho está limitado a 8 KB. Caso queira ver as tags, faça outra solicitação para a operação de API [GET atribuição de tags de objeto](#).
- [Objeto POST](#) – Você pode especificar tags na solicitação POST.

Contanto que as tags na solicitação não ultrapassem o limite de tamanho de cabeçalho de solicitações HTTP de 8 KB, você pode usar a `PUT Object` API para criar objetos com tags. Se as tags especificadas ultrapassarem o limite de tamanho do cabeçalho, você poderá usar esse método POST para incluir as tags no corpo.

- **Objeto PUT - Copiar** – Você pode especificar `x-amz-tagging-directive` na solicitação para instruir o Amazon S3 a copiar (comportamento padrão) as tags ou substituir as tags por um novo conjunto de tags fornecido na solicitação.

Observe o seguinte:

- A marcação segue o modelo de consistência eventual. Isto é, logo após adicionar tags a um objeto, se você tentar recuperar as tags, poderá obter tags antigas, se houver, nos objetos. Contudo, uma chamada subsequente provavelmente fornecerá as tags atualizadas.

## Marcação de objetos e informações adicionais

Esta seção explica como a marcação de objetos está relacionada a outras configurações.

### Marcação de objetos e gerenciamento do ciclo de vida

Na configuração de ciclo de vida de bucket, você pode especificar um filtro para selecionar um subconjunto de objetos ao qual a regra se aplica. Você pode especificar um filtro com base em prefixos de nome de chave, em tags de objeto ou em ambos.

Suponha que você armazene fotos (brutas e no formato concluído) no bucket do Amazon S3. Você pode marcar esses objetos como mostrado a seguir:

```
phototype=raw
or
phototype=finished
```

Você pode considerar o arquivamento das fotos brutas no Glacier pouco tempo depois de serem criadas. Você pode configurar uma regra de ciclo de vida com um filtro que identifica o subconjunto de objetos com o prefixo de nome de chave (`photos/`) que têm uma tag específica (`phototype=raw`).

Para obter mais informações, consulte [Gerenciamento do ciclo de vida de objetos \(p. 122\)](#).

### Marcação de objetos e replicação entre regiões (CRR)

Se você tiver configurado a replicação entre regiões (CRR) no bucket, o Amazon S3 replicará as tags, contanto que você conceda permissão ao S3 para ler as tags. Para obter mais informações, consulte [Visão geral da configuração da CRR \(p. 547\)](#).

### Marcação de objetos e políticas de controle de acesso

Você também pode usar políticas de permissões (políticas de bucket e de usuário) para gerenciar permissões relacionadas à atribuição de tags de objetos. Para ver ações de política, consulte os seguintes tópicos:

- [Permissões para operações de objeto \(p. 330\)](#)
- [Permissões relacionadas a operações de bucket \(p. 331\)](#)

As tags de objeto permitem ter controle de acesso para gerenciar permissões. Você pode conceder permissões condicionais com base em tags de objeto. O Amazon S3 oferece suporte às seguintes chaves de condição que você pode usar para conceder permissões condicionais com base em tags de objeto:

- **s3:ExistingObjectTag/<tag-key>**– Use essa chave de condição para verificar se uma tag de objeto existente tem a chave e o valor de tag específicos.

#### Note

Para conceder permissões para as operações `PUT Object` e `DELETE Object`, não é permitido usar essa chave de condição. Isto é, você não pode criar uma política para conceder ou negar permissões de usuário para excluir ou substituir um objeto existente com base nas tags existentes.

- **s3:RequestObjectTagKeys** – Use essa chave de condição para restringir as chaves de tag que deseja permitir em objetos. Isso é útil para adicionar tags a objetos usando as solicitações `PutObjectTagging` e `PutObject` e de `POST` objeto.
- **s3:RequestObjectTag/<tag-key>** – Use essa chave de condição para restringir as chaves e os valores de tag que deseja permitir em objetos. Isso é útil para adicionar tags a objetos usando as solicitações `PutObjectTagging` e `PutObject` e de bucket `POST`.

Para obter uma lista completa de chaves de condição específicas de serviço do Amazon S3, consulte [Chaves de condição disponíveis \(p. 336\)](#). As seguintes políticas de permissões ilustram como a marcação de objetos permite gerenciar permissões de acesso.

**Example 1:** Permitir que um usuário leia somente os objetos que têm uma tag específica

A política de permissões a seguir concede ao usuário permissão para ler objetos, mas a condição limita a permissão de leitura somente a objetos que possuem a chave e o valor de tag específicos a seguir:

```
security : public
```

Observe que a política usa a chave de condição do Amazon S3, `s3:ExistingObjectTag/<tag-key>`, para especificar a chave e o valor.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::examplebucket/*"
            ],
            "Condition": {
                "StringEquals": {
                    "s3:ExistingObjectTag/security": "public"
                }
            }
        }
    ]
}
```

Example 2: Permitir que um usuário adicione tags de objeto com restrições nas chaves de tag permitidas

A política de permissões a seguir concede ao usuário permissões para executar a ação s3:PutObjectTagging, que permite que o usuário adicione tags a um objeto existente. A condição limita as chaves de tag que o usuário pode usar. A condição usa a chave de condição s3:RequestObjectTagKeys para especificar o conjunto de chaves de tag.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:PutObjectTagging"  
            ],  
            "Resource": [  
                "arn:aws:s3:::examplebucket/*"  
            ],  
            "Condition": {  
                "ForAllValues:StringLike": {  
                    "s3:RequestObjectTagKeys": [  
                        "Owner",  
                        "CreationDate"  
                    ]  
                }  
            }  
        }  
    ]  
}
```

A política garante que o conjunto de tags, se especificado na solicitação, tenha as chaves especificadas. Um usuário pode enviar um conjunto de tags vazio em PutObjectTagging, o que é permitido por essa política (um conjunto de tags vazio na solicitação remove as tags existentes no objeto). Se você quiser impedir que um usuário remova o conjunto de tags, adicione outra condição para garantir que o usuário forneça pelo menos um valor. O ForAnyValue na condição garante que pelo menos um dos valores especificados deva estar presente na solicitação.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:PutObjectTagging"  
            ],  
            "Resource": [  
                "arn:aws:s3:::examplebucket/*"  
            ],  
            "Condition": {  
                "ForAllValues:StringLike": {  
                    "s3:RequestObjectTagKeys": [  
                        "Owner",  
                        "CreationDate"  
                    ]  
                },  
                "ForAnyValue:StringLike": {  
                    "s3:RequestObjectTagKeys": [  
                        "Owner",  
                        "CreationDate"  
                    ]  
                }  
            }  
        }  
    ]  
}
```

```
        }
    ]
}
```

Para obter mais informações, consulte [Criar uma condição que testa vários valores de chave \(operações de conjunto\)](#) no Guia do usuário do IAM.

**Example 3:** Permitir que um usuário adicione tags de objeto que incluam uma chave e um valor de tag específicos

A política de usuário a seguir concede ao usuário permissões para executar a ação `s3:PutObjectTagging`, que permite que o usuário adicione tags a um objeto existente. A condição requer que o usuário inclua uma tag específica (`Project`) com o valor definido como `x`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObjectTagging"
      ],
      "Resource": [
        "arn:aws:s3:::examplebucket/*"
      ],
      "Condition": {
        "StringEquals": {
          "s3:RequestObjectTag/Project": "x"
        }
      }
    ]
}
```

Tópicos relacionados

[Gerenciamento de tags de objeto \(p. 119\)](#)

## Gerenciamento de tags de objeto

Esta seção explica como você pode adicionar tags de objeto programaticamente usando AWS SDK para Java ou o console do Amazon S3.

Tópicos

- [Gerenciamento de tags de objeto usando o console \(p. 119\)](#)
- [Gerenciar tags usando o AWS SDK for Java \(p. 120\)](#)
- [Gerenciamento de tags usando AWS SDK para .NET \(p. 121\)](#)

## Gerenciamento de tags de objeto usando o console

Você pode usar o console do Amazon S3 para adicionar tags a objetos novos ao fazer upload deles ou pode adicioná-las aos objetos existentes. Para obter instruções sobre como adicionar tags a objetos usando o console do Amazon S3, consulte [Adicionar tags de objeto](#) no Guia do usuário do console do Amazon Simple Storage Service.

## Gerenciar tags usando o AWS SDK for Java

O exemplo a seguir mostra como usar o AWS SDK for Java para definir tags para um objeto novo e recuperar ou substituir tags para um objeto existente. Para obter mais informações sobre marcação de objetos, consulte [Marcação de objetos \(p. 114\)](#). Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-
developer-guide/blob/master/LICENSE-SAMPLECODE.)

import java.io.File;
import java.util.ArrayList;
import java.util.List;

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

public class ManagingObjectTags {

    public static void main(String[] args) {
        String clientRegion = "**** Client region ****";
        String bucketName = "**** Bucket name ****";
        String keyName = "**** Object key ****";
        String filePath = "**** File path ****";

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

            // Create an object, add two new tags, and upload the object to Amazon S3.
            PutObjectRequest putRequest = new PutObjectRequest(bucketName, keyName, new
File(filePath));
            List<Tag> tags = new ArrayList<Tag>();
            tags.add(new Tag("Tag 1", "This is tag 1"));
            tags.add(new Tag("Tag 2", "This is tag 2"));
            putRequest.setObjectTagging(new ObjectTagging(tags));
            PutObjectResult putResult = s3Client.putObject(putRequest);

            // Retrieve the object's tags.
            GetObjectTaggingRequest getTaggingRequest = new
GetObjectTaggingRequest(bucketName, keyName);
            GetObjectTaggingResult getTagsResult =
s3Client.getObjectTagging(getTaggingRequest);

            // Replace the object's tags with two new tags.
            List<Tag> newTags = new ArrayList<Tag>();
            newTags.add(new Tag("Tag 3", "This is tag 3"));
            newTags.add(new Tag("Tag 4", "This is tag 4"));
            s3Client.setObjectTagging(new SetObjectTaggingRequest(bucketName, keyName, new
ObjectTagging(newTags)));
        }
        catch(AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        }
        catch(SdkClientException e) {
```

```
// Amazon S3 couldn't be contacted for a response, or the client
// couldn't parse the response from Amazon S3.
e.printStackTrace();
}
}
```

## Gerenciamento de tags usando AWS SDK para .NET

O exemplo a seguir mostra como usar o AWS SDK para .NET para definir as tags para um objeto novo e recuperar ou substituir as tags para um objeto existente. Para obter mais informações sobre marcação de objetos, consulte [Marcação de objetos \(p. 114\)](#).

Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-
developer-guide/blob/master/LICENSE-SAMPLECODE.)

using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    public class ObjectTagsTest
    {
        private const string bucketName = "**** bucket name ****";
        private const string keyName = "**** key name for the new object ****";
        private const string filePath = @"**** file path ****";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            PutObjectWithTagsTestAsync().Wait();
        }

        static async Task PutObjectWithTagsTestAsync()
        {
            try
            {
                // 1. Put an object with tags.
                var putRequest = new PutObjectRequest
                {
                    BucketName = bucketName,
                    Key = keyName,
                    FilePath = filePath,
                    TagSet = new List<Tag>{
                        new Tag { Key = "Keyx1", Value = "Value1" },
                        new Tag { Key = "Keyx2", Value = "Value2" }
                    }
                };

                PutObjectResponse response = await client.PutObjectAsync(putRequest);
                // 2. Retrieve the object's tags.
                GetObjectTaggingRequest getTagsRequest = new GetObjectTaggingRequest
                {
```

```
        BucketName = bucketName,
        Key = keyName
    };

    GetObjectTaggingResponse objectTags = await
client.GetObjectTaggingAsync(getTagsRequest);
    for (int i = 0; i < objectTags.Tagging.Count; i++)
        Console.WriteLine("Key: {0}, Value: {1}", objectTags.Tagging[i].Key,
objectTags.Tagging[0].Value);

    // 3. Replace the tagset.

    Tagging newTagSet = new Tagging();
    newTagSet.TagSet = new List<Tag>{
        new Tag { Key = "Key3", Value = "Value3" },
        new Tag { Key = "Key4", Value = "Value4" }
    };

    PutObjectTaggingRequest putObjTagsRequest = new PutObjectTaggingRequest()
{
    BucketName = bucketName,
    Key = keyName,
    Tagging = newTagSet
};

    PutObjectTaggingResponse response2 = await
client.PutObjectTaggingAsync(putObjTagsRequest);

    // 4. Retrieve the object's tags.
    GetObjectTaggingRequest getTagsRequest2 = new GetObjectTaggingRequest();
    getTagsRequest2.BucketName = bucketName;
    getTagsRequest2.Key = keyName;
    GetObjectTaggingResponse objectTags2 = await
client.GetObjectTaggingAsync(getTagsRequest2);
    for (int i = 0; i < objectTags2.Tagging.Count; i++)
        Console.WriteLine("Key: {0}, Value: {1}", objectTags2.Tagging[i].Key,
objectTags2.Tagging[0].Value);

}
catch (AmazonS3Exception e)
{
    Console.WriteLine(
        "Error encountered ***. Message:'{0}' when writing an object"
        , e.Message);
}
catch (Exception e)
{
    Console.WriteLine(
        "Encountered an error. Message:'{0}' when writing an object"
        , e.Message);
}
}
```

## Gerenciamento do ciclo de vida de objetos

Para gerenciar seus objetos de maneira que sejam armazenados de maneira econômica durante todo o ciclo de vida, configure o ciclo de vida deles. Configuração de ciclo de vida é um conjunto de regras que define as ações aplicadas pelo Amazon S3 a um grupo de objetos. Existem dois tipos de ações:

- Ações de transição—Definem quando os objetos fazem a transição para outra [classe de armazenamento](#). Por exemplo, você pode optar por fazer a transição de objetos para a classe de armazenamento STANDARD\_IA 30 dias após a criação ou arquivá-los na classe GLACIER um ano após a criação.

Há custos associados às solicitações de transição do ciclo de vida. Para obter informações sobre preços, consulte [Definição de preço do Amazon S3](#).

- Ações de expiração—definem quando os objetos expiram. O Amazon S3 exclui os objetos expirados em seu nome.

Os custos de expiração do ciclo de vida dependem de quando você escolhe tornar objetos expirados. Para obter mais informações, consulte [Configurar a expiração de objeto \(p. 129\)](#).

Para obter mais informações sobre regras de ciclo de vida, consulte [Elementos de configuração do ciclo de vida \(p. 130\)](#).

## Quando devo usar a configuração de ciclo de vida?

Defina regras de configuração de ciclo de vida para objetos com ciclo de vida bem definido. Por exemplo:

- Se você fizer upload periódico de logs em um bucket, é possível que seu aplicativo precise deles por uma semana ou um mês. Depois disso, você pode excluí-los.
- Alguns documentos são acessados frequentemente por um período limitado. Depois disso, eles serão acessados com pouca frequência. Em algum ponto, você pode não precisar de acesso em tempo real a esses objetos, mas sua organização ou as regulamentações podem exigir que você os arquive por um período específico. Depois disso, é possível excluí-los.
- É possível fazer upload de alguns tipos de dados no Amazon S3 para fins de arquivamento. Por exemplo, é possível arquivar mídias digitais, registros financeiros e de saúde, dados não processados de sequência genômica, backups de banco de dados de longo prazo e dados que devem ser retidos para conformidade regulamentar.

Com regras de configuração de ciclo de vida, é possível solicitar que o Amazon S3 faça a transição de objetos para classes de armazenamento menos caras, arquive-os ou exclua-os.

## Como configuro um ciclo de vida?

Uma configuração de ciclo de vida, um arquivo XML, consiste em um conjunto de regras com ações predefinidas que você deseja que o Amazon S3 execute em objetos durante sua vida útil.

O Amazon S3 fornece um conjunto de operações de API para gerenciamento da configuração de ciclo de vida em um bucket. O Amazon S3 armazena a configuração como um sub-recurso de ciclo de vida anexado ao seu bucket. Para obter detalhes, consulte:

[Ciclo de vida de PUT Bucket](#)

[Ciclo de vida de GET Bucket](#)

[DELETE Bucket lifecycle](#)

Você também pode configurar o ciclo de vida usando o console do Amazon S3 ou de maneira programática, usando as bibliotecas wrapper de SDK da AWS. Se necessário, você também pode chamar a API REST diretamente. Para obter mais informações, consulte [Definir a configuração do ciclo de vida em um bucket \(p. 147\)](#).

Para obter mais informações, consulte os tópicos a seguir:

- [Outras considerações sobre a configuração de ciclo de vida \(p. 124\)](#)
- [Elementos de configuração do ciclo de vida \(p. 130\)](#)
- [Exemplos de configuração de ciclo de vida \(p. 137\)](#)
- [Definir a configuração do ciclo de vida em um bucket \(p. 147\)](#)

## Outras considerações sobre a configuração de ciclo de vida

Ao configurar o ciclo de vida de objetos, é necessário entender as diretrizes a seguir de transição de objetos, configuração de datas de expiração e outras configurações de objeto.

### Tópicos

- [Fazer a transição de objetos \(p. 124\)](#)
- [Configurar a expiração de objeto \(p. 129\)](#)
- [Ciclo de vida e outras configurações de bucket \(p. 129\)](#)

## Fazer a transição de objetos

É possível adicionar regras a uma configuração de ciclo de vida para solicitar que o Amazon S3 faça a transição de objetos para outra [classe de armazenamento](#) do Amazon S3. Por exemplo:

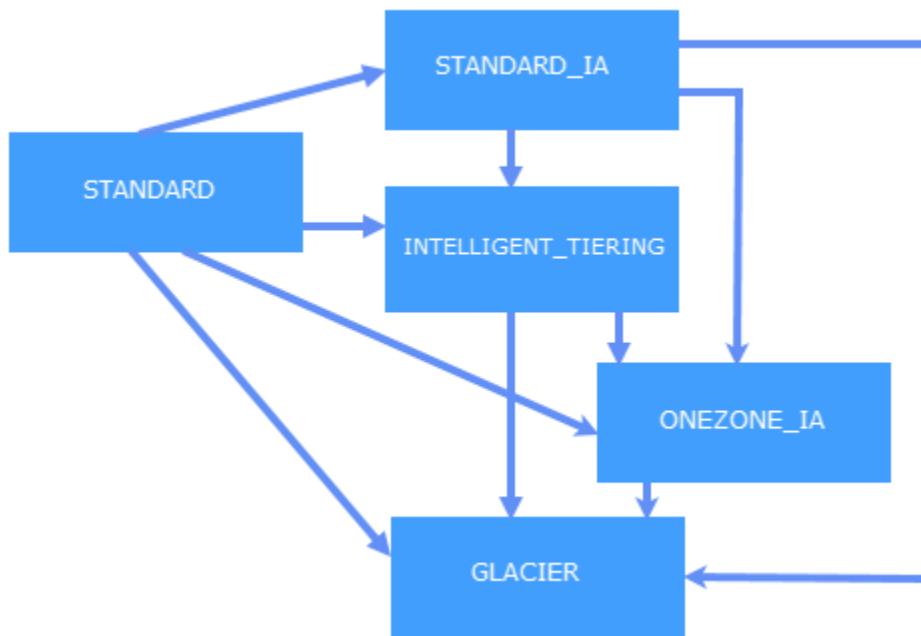
- Quando você sabe que esses objetos são acessos com pouca frequência, é possível fazer a transição deles para a classe de armazenamento STANDARD\_IA.
- É possível arquivar objetos que não precisam de acesso em tempo real à classe de armazenamento GLACIER.

As seções a seguir descrevem transições com suporte, limitações relacionadas e a transição para a classe de armazenamento GLACIER.

### Transições com suporte e limitações relacionadas

Em uma configuração de ciclo de vida, você pode definir regras para fazer a transição de objetos de uma classe de armazenamento para outra a fim de economizar custos de armazenamento. Quando desconhece os padrões de acesso dos objetos, ou os padrões de acesso mudam com o passar do tempo, você faz a transição para os objetos para a classe de armazenamento INTELLIGENT\_TIERING para economia automática. Para obter informações sobre classes de armazenamento, consulte [Classes de armazenamento \(p. 107\)](#).

O Amazon S3 da suporte a um modelo de cachoeira para fazer a transição entre classes de armazenamento, conforme mostrado no diagrama a seguir.



#### Note

O diagrama não menciona a classe de armazenamento de **REDUCED\_REDUNDANCY**, pois não recomendamos o uso dela.

O Amazon S3 dá suporte às seguintes transições do ciclo de vida entre as classes de armazenamento usando uma configuração do ciclo de vida:

- Você pode fazer a transição da classe de armazenamento **STANDARD** para qualquer outra classe.
- Você pode fazer a transição de qualquer classe de armazenamento para a classe de armazenamento **GLACIER**.
- Você pode fazer a transição da classe de armazenamento **STANDARD\_IA** para as classes de armazenamento **INTELLIGENT\_TIERING** ou **ONEZONE\_IA**.
- Você pode fazer a transição da classe de armazenamento **INTELLIGENT\_TIERING** para a classe de armazenamento **ONEZONE\_IA**.

Não há suporte para as seguintes transições do ciclo de vida:

- Você não pode fazer a transição de qualquer classe de armazenamento para a classe de armazenamento **STANDARD**.
- Você não pode fazer a transição de qualquer classe de armazenamento para a classe de armazenamento **REDUCED\_REDUNDANCY**.
- Você não pode fazer a transição da classe de armazenamento **INTELLIGENT\_TIERING** para a classe de armazenamento **STANDARD\_IA**.
- Você não pode fazer a transição da classe de armazenamento **ONEZONE\_IA** para as classes de armazenamento **STANDARD\_IA** ou **INTELLIGENT\_TIERING**.
- Você não pode fazer a transição da classe de armazenamento **GLACIER** para qualquer outra classe.

As transições da classe de armazenamento do ciclo de vida têm as seguintes restrições:

- Da classe de armazenamento STANDARD ou STANDARD\_IA para INTELLIGENT\_TIERING. As limitações a seguir aplicam-se:
  - Para objetos maiores, há custo-benefício para fazer a transição para INTELLIGENT\_TIERING. O Amazon S3 não faz a transição de objetos que sejam menores que 128 KB para a classe de armazenamento INTELLIGENT\_TIERING porque não é econômico.
- Das classes de armazenamento STANDARD para STANDARD\_IA ou ONEZONE\_IA. As limitações a seguir aplicam-se:
  - Para objetos maiores, há custo-benefício para fazer a transição para STANDARD\_IA ou ONEZONE\_IA. O Amazon S3 não faz a transição de objetos que sejam menores que 128 KB para as classes de armazenamento STANDARD\_IA ou ONEZONE\_IA porque não é econômico.
- Os objetos devem ser armazenados por pelo menos 30 dias na classe de armazenamento atual antes da transição para STANDARD\_IA ou ONEZONE\_IA. Por exemplo, não é possível criar uma regra de ciclo de vida para fazer a transição de objetos para a classe de armazenamento STANDARD\_IA um dia após a criação deles.

O Amazon S3 não faz a transição de objetos nos primeiros 30 dias porque os objetos mais novos são geralmente acessados com mais frequência ou excluídos mais cedo do que é apropriado para o armazenamento STANDARD\_IA ou ONEZONE\_IA.

- Se você estiver fazendo a transição de objetos desatualizados (em buckets com versões), poderá mover apenas objetos desatualizados há pelo menos 30 dias para o armazenamento STANDARD\_IA ou ONEZONE\_IA.
- Da classe de armazenamento STANDARD\_IA para ONEZONE\_IA. As limitações a seguir aplicam-se:
  - Os objetos devem ser armazenados por pelo menos 30 dias na classe de armazenamento STANDARD\_IA antes da transição para a classe ONEZONE\_IA.

Você pode combinar essas ações de ciclo de vida para gerenciar o ciclo de vida completo de um objeto. Por exemplo, suponha que os objetos criados tenham um ciclo de vida bem definido. No início, os objetos são acessados com frequência em um período de 30 dias. Depois disso, eles são acessados com pouca frequência por 90 dias. Depois desse período, eles não são mais necessário. Portanto, é possível optar por arquivá-los ou excluí-los.

Nesse cenário, é possível criar uma regra de ciclo de vida na qual você especifica a ação inicial de transição para o armazenamento INTELLIGENT\_TIERING, STANDARD\_IA ou ONEZONE\_IA, outra ação de transição para o armazenamento GLACIER para arquivamento e uma ação de expiração. Ao mover objetos de uma classe de armazenamento para outra, você economiza no custo de armazenamento. Para obter mais informações sobre considerações de custo, consulte [Definição de preço do Amazon S3](#).

#### Important

Não é possível especificar uma única regra de ciclo de vida para transições para INTELLIGENT\_TIERING (ou STANDARD\_IA ou ONEZONE\_IA) e GLACIER quando a transição para GLACIER ocorre em menos de 30 dias após a transição para INTELLIGENT\_TIERING, STANDARD\_IA ou ONEZONE\_IA. Isso se deve ao fato de que há uma cobrança mínima de

armazenamento de 30 dias associada às classes de armazenamento INTELLIGENT\_TIERING, STANDARD\_IA e ONEZONE\_IA.

O mesmo mínimo de 30 dias se aplica ao especificar uma transição do armazenamento STANDARD\_IA para ONEZONE\_IA ou INTELLIGENT\_TIERING. É possível especificar duas regras para realizar isso, mas é necessário pagar as cobranças mínimas de armazenamento. Para obter mais informações sobre considerações de custo, consulte [Definição de preço do Amazon S3](#).

## Transição para a classe de armazenamento GLACIER (arquivamento de objeto)

Usando a configuração de ciclo de vida, você pode fazer a transição de objetos para a classe de armazenamento GLACIER—isto é, arquivar dados no Glacier, uma solução de armazenamento de baixo custo.

### Important

Ao selecionar a classe de armazenamento do GLACIER, o Amazon S3 usa o serviço Glacier de baixo custo para armazenar objetos. Embora os objetos sejam armazenados no Glacier, eles continuam sendo objetos do Amazon S3 que você gerencia no Amazon S3, e não é possível acessá-los diretamente por meio do Glacier.

Para que você arquive objetos, reveja as seguintes seções para considerações relevantes.

### Considerações gerais

Veja a seguir as considerações gerais que você deve fazer antes de arquivar objetos:

- Os objetos criptografados permanecem criptografados durante todo o processo de transição da classe de armazenamento.
- Os objetos na classe de armazenamento GLACIER não estão disponíveis em tempo real.

Os objetos arquivados são objetos do Amazon S3, mas para acessar um objeto arquivado, primeiro você deve restaurar uma cópia temporária dele. A cópia restaurada do objeto fica disponível somente pelo tempo que você especifica na solicitação de restauração. Depois disso, o Amazon S3 exclui a cópia temporária, e o objeto permanece arquivado no Glacier.

Você pode restaurar um objeto usando o console do Amazon S3 ou, programaticamente, usando bibliotecas wrapper dos SDKs da AWS ou a API REST do Amazon S3 em seu código. Para obter mais informações, consulte [Restaurar objetos arquivados \(p. 259\)](#).

- A transição de objetos para a classe de armazenamento GLACIER é unidirecional.

Você não pode usar uma regra de configuração de ciclo de vida para converter a classe de armazenamento de um objeto de GLACIER em qualquer outra classe de armazenamento. Se você quiser mudar a classe de armazenamento de um objeto arquivado em outra classe de armazenamento, deverá usar a operação de restauração para fazer primeiro uma cópia temporária do objeto. Em seguida, use a operação de cópia para substituir o objeto especificando STANDARD, INTELLIGENT\_TIERING, STANDARD\_IA, ONEZONE\_IA ou REDUCED\_REDUNDANCY como a classe de armazenamento.

- Os objetos de classe de armazenamento GLACIER estão visíveis e disponíveis somente pelo Amazon S3, e não pelo Glacier.

O Amazon S3 armazena os objetos arquivados no Glacier. Contudo, esses são objetos do Amazon S3, e você pode acessá-los apenas usando o console do Amazon S3 ou a API do Amazon S3. Você não pode acessar os objetos arquivados pelo console do Glacier ou pela API do Glacier.

### Considerações sobre custos

Se você planeja arquivar dados acessados com pouca frequência por um período de meses ou anos, a classe de armazenamento GLACIER geralmente reduzirá seus custos com armazenamento. Porém, considere o seguinte para garantir que a classe de armazenamento GLACIER seja adequada para você:

- Cobranças extras de armazenamento – Quando você faz a transição de objetos para a classe de armazenamento GLACIER, uma quantidade fixa de armazenamento é adicionada a cada objeto para acomodar metadados para gerenciar o objeto.
- Para cada objeto arquivado no Glacier, o Amazon S3 usa 8 KB de armazenamento para o nome do objeto e outros metadados. O Amazon S3 armazena esses metadados para que você possa obter uma lista em tempo real dos seus objetos arquivados usando a API do Amazon S3. Para obter mais informações, consulte [GET bucket \(listar objetos\)](#). Você é cobrado pelas taxas padrão do Amazon S3 nesse armazenamento adicional.
- Para cada objeto arquivado, o Glacier adiciona 32 KB de armazenamento para o índice e metadados relacionados. Esses dados extras são necessários para identificar e recuperar seu objeto. Você é cobrado pelas taxas do Glacier nesse armazenamento adicional.

Se você estiver arquivando objetos pequenos, considere esses encargos de armazenamento. Considere também a possibilidade de agregar vários objetos pequenos em um número menor de objetos grandes para reduzir os gastos adicionais.

- Número de dias que você planeja manter objetos arquivados —O Glacier é uma solução de arquivamento a longo prazo. A exclusão de dados arquivados no Glacier será gratuita se os objetos que você excluir estiverem arquivados por três meses ou mais. Se você excluir ou substituir um objeto dentro de três meses de seu arquivamento, o Amazon S3 cobrará uma taxa proporcional pela exclusão antecipada.
- Cobranças por solicitação de arquivamento no Glacier — Cada objeto que você migra para a classe de armazenamento GLACIER constitui uma solicitação de arquivamento. Há um custo para cada solicitação desse tipo. Se você pretende fazer a transição de um grande número de objetos, considere os custos de solicitação.
- Cobranças pela restauração de dados do Glacier — O Glacier foi desenvolvido para arquivamento de longo prazo de dados que você acessará raramente. Para obter informações sobre cobranças de restauração de dados, consulte [Quanto custa para recuperar dados do Glacier?](#) nas Perguntas frequentes do Amazon S3. Para obter informações sobre como restaurar dados do Glacier, consulte [Restaurar objetos arquivados \(p. 259\)](#).

Quando você arquiva objetos no Glacier usando o gerenciamento de ciclo de vida do objeto, o Amazon S3 faz a transição desses objetos assincronamente. Pode haver um atraso entre a data de transição na regra de configuração de ciclo de vida e a data de transição física. Você é cobrado pelos preços do Glacier com base na data de transição especificada na regra.

A página de detalhes de produto do Amazon S3 fornece informações sobre definição de preço e exemplos de cálculo para arquivamento de objetos no Amazon S3. Para obter mais informações, consulte os tópicos a seguir:

- [Como é calculada a cobrança de armazenamento de objetos do Amazon S3 arquivados no Glacier?](#)
- [Como sou cobrado pela exclusão de objetos do Glacier com menos de 3 meses de arquivamento?](#)
- [Quanto custa para recuperar dados do Glacier?](#)
- [Definição de preço do Amazon S3 para custos de armazenamento para as classes de armazenamento Standard e GLACIER.](#)

### Restaurar objetos arquivados

Os objetos arquivados não estão acessíveis em tempo real. Primeiro inicie uma solicitação de restauração e, em seguida, aguarde até que uma cópia temporária do objeto esteja disponível pelo tempo que você especificar na solicitação. Depois de receber uma cópia temporária do objeto restaurado, a classe de armazenamento do objeto continuará sendo GLACIER (uma solicitação GET ou HEAD retornará GLACIER como a classe de armazenamento).

#### Note

Ao restaurar um arquivo, você pagará pelo arquivamento (taxa do GLACIER) e pela cópia restaurada temporariamente (taxa de armazenamento REDUCED\_REDUNDANCY). Para obter mais informações sobre definição de preços, consulte [Definição de preços do Amazon S3](#).

É possível restaurar uma cópia de objeto de maneira programática ou usando o console do Amazon S3. O Amazon S3 processa apenas uma solicitação de restauração por vez, por objeto. Para obter mais informações, consulte [Restaurar objetos arquivados \(p. 259\)](#).

### Configurar a expiração de objeto

Quando um objeto atinge o fim de seu ciclo de vida, o Amazon S3 coloca o objeto em uma fila para remoção e o remove assincronamente. Pode haver um atraso entre a data de expiração e a data em que o Amazon S3 remove um objeto. Você não será cobrado pelo tempo de armazenamento associado a um objeto que expirou.

Para descobrir quando um objeto está programado para expirar, use as operações de API [HEAD Object](#) ou [GET Object](#). Essas operações de API retornam os cabeçalhos de resposta que fornecem essas informações.

Se criar uma regra de expiração de ciclo de vida que resulte na expiração de objetos armazenados na classe INTELLIGENT\_TIERING, STANDARD\_IA (ou ONEZONE\_IA) por pelo menos 30 dias, você será cobrado por 30 dias. Se criar uma regra de expiração de ciclo de vida que resulte na expiração de objetos armazenados na classe GLACIER por pelo menos 90 dias, você será cobrado por 90 dias. Para obter mais informações, consulte a [Definição de preço do Amazon S3](#).

### Ciclo de vida e outras configurações de bucket

Além de configurações de ciclo de vida, você pode associar outras configurações a seu bucket. Esta seção explica como a configuração de ciclo de vida está relacionada a outras configurações de bucket.

#### Ciclo de vida e versionamento

Você pode adicionar configurações de ciclo de vida a buckets com e sem versionamento. Para obter mais informações, consulte [Versionamento de objeto \(p. 111\)](#).

Um bucket com versionamento habilitado mantém uma versão atual do objeto e versões desatualizadas do objeto (se disponíveis). Você pode definir regras separadas de ciclo de vida para versões atuais e não atuais do objeto.

Para obter mais informações, consulte [Elementos de configuração do ciclo de vida \(p. 130\)](#). Para obter informações sobre versionamento, consulte [Versionamento de objeto \(p. 111\)](#).

## Configuração do ciclo de vida em buckets com MFA habilitado

Não há suporte para a configuração de ciclo de vida em buckets com MFA habilitada.

## Ciclo de vida e registro

Se você tiver o registro habilitado em seu bucket, o Amazon S3 informará os resultados da ação de expiração do seguinte modo:

- Se a ação de expiração de ciclo de vida fizer o Amazon S3 remover permanentemente o objeto, o Amazon S3 informará isso como uma operação `S3.EXPIRE.OBJECT` nos registros de log.
- Para um bucket com versionamento habilitado, se a ação de expiração de ciclo de vida levar a uma exclusão lógica da versão atual, na qual o Amazon S3 adiciona um marcador de exclusão, o Amazon S3 informará a exclusão lógica como uma operação `S3.CREATE.DELETEMARKER` no registro de log. Para obter mais informações, consulte [Versionamento de objeto \(p. 111\)](#).
- Ao fazer a transição de um objeto para a classe de armazenamento GLACIER, o Amazon S3 informa isso como uma operação `S3.TRANSITION.OBJECT` no registro de log para indicar que iniciou a operação. Quando é feita a transição do objeto para a classe de armazenamento STANDARD\_IA (ou ONEZONE\_IA), isso é informado como uma operação `S3.TRANSITION_SIA.OBJECT` (ou `S3.TRANSITION_ZIA.OBJECT`).

## Mais informações

- [Elementos de configuração do ciclo de vida \(p. 130\)](#)
- [Transição para a classe de armazenamento GLACIER \(arquivamento de objeto\) \(p. 127\)](#)
- [Definir a configuração do ciclo de vida em um bucket \(p. 147\)](#)

# Elementos de configuração do ciclo de vida

## Tópicos

- [Elemento ID \(p. 131\)](#)
- [Elemento Status \(p. 131\)](#)
- [Elemento Filter \(p. 131\)](#)
- [Elementos para descrever ações de ciclo de vida \(p. 133\)](#)

Especifique uma configuração de ciclo de vida como XML, consistindo em uma ou mais regras de ciclo de vida.

```
<LifecycleConfiguration>
  <Rule>
    ...
  </Rule>
  <Rule>
    ...
  </Rule>
```

```
</LifecycleConfiguration>
```

Cada regra consiste no seguinte:

- Metadados de regra que incluem um ID de regra e o status que indica se a regra está ativada ou desativada. Se uma regra estiver desativada, o Amazon S3 não executará as ações especificadas nela.
- Filtro que identifica os objetos aos quais a regra se aplica. Você pode especificar um filtro usando um prefixo de chaves de objeto, uma ou mais tags de objeto ou ambos.
- Uma ou mais ações de transição ou expiração com uma data ou um período no ciclo de vida do objeto quando você deseja que o Amazon S3 realize a ação especificada.

As seções a seguir descrevem os elementos XML em uma configuração de ciclo de vida. Para obter configurações de ciclo de vida de exemplo, consulte [Exemplos de configuração de ciclo de vida \(p. 137\)](#).

## Elemento ID

Uma configuração de ciclo de vida pode ter até 1.000 regras. O elemento <ID> identifica uma regra com exclusividade. O tamanho do ID está limitado a 255 caracteres.

## Elemento Status

O valor de elemento <Status> pode ser Ativado ou Desativado. Se uma regra estiver desativada, o Amazon S3 não executará as ações definidas nela.

## Elemento Filter

Uma regra de ciclo de vida pode ser aplicada a todos os objetos ou a um subconjunto de objetos em um bucket com base no elemento <Filter> que você especifica na regra de ciclo de vida.

É possível filtrar objetos por prefixo de chaves, por tag de objeto ou por uma combinação dos dois. Nesse último caso, o Amazon S3 usa um E lógico para combinar os filtros. Considere os seguintes exemplos:

- Especificação de um filtro usando prefixos de chaves – Este exemplo mostra uma regra de ciclo de vida que se aplica a um subconjunto de objetos com base no prefixo de nome de chave (logs/). Por exemplo, a regra de ciclo de vida se aplica aos objetos logs/mylog.txt, logs/temp1.txt e logs/test.txt. A regra não se aplica ao objeto example.jpg.

```
<LifecycleConfiguration>
  <Rule>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    transition/expiration actions.
    ...
  </Rule>
  ...
</LifecycleConfiguration>
```

Caso queira aplicar uma ação de ciclo de vida a um subconjunto de objetos com base em prefixos de nome de chave diferentes, especifique regras separadas. Em cada regra, especifique um filtro com base em prefixo. Por exemplo, para descrever uma ação de ciclo de vida para objetos com prefixos de chaves projectA/ e projectB/, especifique duas regras da seguinte forma:

```
<LifecycleConfiguration>
  <Rule>
    <Filter>
      <Prefix>projectA/</Prefix>
```

```
</Filter>
transition/expiration actions.
...
</Rule>

<Rule>
<Filter>
<Prefix>projectB/<Prefix>
</Filter>
transition/expiration actions.
...
</Rule>
</LifecycleConfiguration>
```

Para obter mais informações sobre chaves de objeto, consulte [Chaves de objeto \(p. 102\)](#).

- Especificação de um filtro com base em tags de objeto – No seguinte exemplo, a regra de ciclo de vida especifica um filtro com base em uma tag (**chave**) e um valor (**valor**). A regra aplica-se somente a um subconjunto de objetos com a tag específica.

```
<LifecycleConfiguration>
<Rule>
<Filter>
<Tag>
<Key>key</Key>
<Value>value</Value>
</Tag>
</Filter>
transition/expiration actions.
...
</Rule>
</LifecycleConfiguration>
```

Você pode especificar um filtro com base em várias tags. Você deve envolver as tags no elemento <AND> mostrado no exemplo a seguir. A regra instrui o Amazon S3 a executar ações de ciclo de vida em objetos com duas tags (com a chave e o valor específicos da tag).

```
<LifecycleConfiguration>
<Rule>
<Filter>
<And>
<Tag>
<Key>key1</Key>
<Value>value1</Value>
</Tag>
<Tag>
<Key>key2</Key>
<Value>value2</Value>
</Tag>
...
</And>
</Filter>
transition/expiration actions.
</Rule>
</Lifecycle>
```

A regra de ciclo de vida se aplica a objetos que têm as duas tags especificadas. O Amazon S3 executa o operador lógico AND. Observe o seguinte:

- Cada tag deve corresponder exatamente à chave e ao valor.
- A regra se aplica a um subconjunto de objetos com uma ou mais tags especificadas na regra. Se um objeto tem outras tags especificadas, isso não importa.

### Note

Quando você especifica várias tags em um filtro, cada chave de tag deve ser exclusiva.

- Especificação de um filtro com base no prefixo e em uma ou mais tags – Em uma regra de ciclo de vida, você pode especificar um filtro com base no prefixo de chaves e em uma ou mais tags. Além disso, você deve encapsular tudo isso no elemento `<And>` como mostrado a seguir:

```
<LifecycleConfiguration>
  <Rule>
    <Filter>
      <And>
        <Prefix>key-prefix</Prefix>
        <Tag>
          <Key>key1</Key>
          <Value>value1</Value>
        </Tag>
        <Tag>
          <Key>key2</Key>
          <Value>value2</Value>
        </Tag>
        ...
      </And>
    </Filter>
    <Status>Enabled</Status>
    transition/expiration actions.
  </Rule>
</LifecycleConfiguration>
```

O Amazon S3 combina esses filtros usando um E lógico. Isto é, a regra aplica-se ao subconjunto de objetos com o prefixo de chaves específico e as tags específicas. Um filtro pode ter somente um prefixo e zero ou mais tags.

- Você pode especificar um filtro vazio e, nesse caso, a regra se aplica a todos os objetos no bucket.

```
<LifecycleConfiguration>
  <Rule>
    <Filter>
    </Filter>
    <Status>Enabled</Status>
    transition/expiration actions.
  </Rule>
</LifecycleConfiguration>
```

## Elementos para descrever ações de ciclo de vida

Você pode instruir o Amazon S3 a executar ações específicas no ciclo de vida de um objeto, especificando uma ou mais das seguintes ações predefinidas em uma regra de ciclo de vida. O efeito dessas ações depende do estado do versionamento de seu bucket.

- Elemento da ação Transição – Você especifica a ação `Transition` para fazer a transição de objetos de uma classe de armazenamento para outra. Para obter mais informações sobre transição de objetos, consulte [Transições com suporte e limitações relacionadas \(p. 124\)](#). Quando uma data ou um período especificado no ciclo de vida do objeto é atingido, o Amazon S3 executa a transição.

Para um bucket com versões (versionamento ativado ou suspenso no bucket), a ação `Transition` aplica-se à versão do objeto atual. Para gerenciar versões não atuais, o Amazon S3 define a ação `NoncurrentVersionTransition` (descrita abaixo).

- Elemento de ação de expiração – A ação `Expiration` expira os objetos identificados na regra e se aplica a objetos qualificados em qualquer uma das classes de armazenamento do Amazon S3. Para obter mais informações sobre classes de armazenamento, consulte [Classes de armazenamento \(p. 107\)](#). O Amazon S3 torna todos os objetos expirados indisponíveis. A remoção permanente dos objetos depende do estado de versionamento do bucket.

**Important**

As políticas de ciclo de vida de expiração de objeto não removem multipart uploads incompletos. Para remover os multipart uploads incompletos, você deve usar a ação de configuração de ciclo de vida `AbortIncompleteMultipartUpload` que é descrita posteriormente nesta seção.

- Bucket sem versão – A ação `Expiration` resulta na remoção permanente do objeto pelo Amazon S3.
- Bucket com versão – Para um bucket com versão (ou seja, versionamento ativado ou suspenso), há várias considerações que orientam como o Amazon S3 trata a ação `expiration`. Para obter mais informações, consulte [Usar versionamento \(p. 448\)](#). Independentemente do estado do versionamento, o seguinte é aplicado:
  - A ação `Expiration` se aplica somente à versão atual (não afeta versões não atuais do objeto).
  - O Amazon S3 não realizará ações se houver uma ou mais versões de objeto e se o marcador de exclusão estiver na versão atual.
  - Se a versão atual do objeto for a única versão do objeto e também houver um marcador de exclusão (também chamado de marcador de exclusão de objeto expirado, onde todas as versões de objeto são excluídas e você tem somente um marcador de exclusão restante), o Amazon S3 removerá o marcador de exclusão de objeto expirado. Você também pode usar a ação de expiração para instruir o Amazon S3 a remover os marcadores de exclusão de objeto expirado. Para ver um exemplo, consulte [Exemplo 7: remoção de marcadores de exclusão de objetos expirados \(p. 144\)](#).

Ao configurar o Amazon S3 para gerenciar a expiração, considere também:

- Bucket com versionamento ativado

Se a versão atual do objeto não for um marcador de exclusão, o Amazon S3 adicionará um com um ID exclusivo de versão. Isso torna a versão atual desatualizada, e o marcador de exclusão se torna a versão atual.

- Bucket com versionamento suspenso

Em um bucket com versionamento suspenso, a ação de expiração faz com que o Amazon S3 crie um marcador de exclusão com ID de versão nulo. Esse marcador de exclusão substitui qualquer versão de objeto por um ID de versão nulo na hierarquia de versões, que exclui o objeto.

Além disso, o Amazon S3 fornece as seguintes ações que você pode usar para gerenciar versões de objeto não atuais em um bucket com versão (isto é, buckets com versionamento ativado e suspenso).

- Elemento de ação `NoncurrentVersionTransition` – Use essa ação para especificar quanto tempo (desde quando os objetos passam a ser não atuais) você deseja que os objetos permaneçam na classe de armazenamento atual antes que o Amazon S3 fizesse a transição deles para a classe de armazenamento especificada. Para obter mais informações sobre transição de objetos, consulte [Transições com suporte e limitações relacionadas \(p. 124\)](#).
- Elemento de ação `NoncurrentVersionExpiration` – Use essa ação para especificar quanto tempo (desde quando os objetos passam a ser não atuais) você deseja manter as versões não atuais do objeto antes que o Amazon S3 as remova permanentemente. O objeto excluído não pode ser recuperado.

Essa remoção retardada de objetos não atuais pode ser útil quando você precisa corrigir exclusões ou substituições acidentais. Por exemplo, você pode configurar uma regra de expiração para excluir versões não atuais cinco dias após ficarem nesse estado. Por exemplo, imagine que, em 1/1/2014 10:30 AM UTC, você crie um objeto de [versão 1 da API 2006-03-10](#) de versão 111111). Em 1/2/2014 11:30

AM UTC, você exclui acidentalmente `photo.gif` (ID de versão 111111), o que cria um marcador de exclusão com um novo ID de versão (como ID de versão 4857693). Agora você tem cinco dias para recuperar a versão original de `photo.gif` (ID de versão 111111) até que a exclusão seja permanente. Em 1/8/2014 00:00 UTC, a regra de ciclo de vida para expiração executa e exclui permanentemente `photo.gif` (ID de versão 111111), cinco dias depois que ele passa a ser uma versão não atual.

#### Important

As políticas de ciclo de vida de expiração do objeto não removem multipart uploads incompletos. Para remover os multipart uploads incompletos, você deve usar a ação de configuração de ciclo de vida `AbortIncompleteMultipartUpload` que é descrita posteriormente nesta seção.

Além das ações de transição e expiração, você pode usar a ação de configuração de ciclo de vida a seguir para instruir o Amazon S3 a abortar multipart uploads incompletos.

- Elemento de ação `AbortIncompleteMultipartUpload` – Use esse elemento para definir o tempo máximo (em dias) que você deseja permitir que os multipart uploads permaneçam em andamento. Se os multipart uploads aplicáveis (determinados pelo nome de chave `prefix` especificado na regra de ciclo de vida) não forem concluídos no período predefinido, o Amazon S3 abortará os multipart uploads incompletos. Para obter mais informações, consulte [Anular multipart uploads incompletos usando uma política de ciclo de vida de bucket \(p. 183\)](#).

#### Note

Você não pode especificar essa ação de ciclo de vida em uma regra que especifica um filtro com base em tags de objeto.

- Elemento de ação `ExpiredObjectDeleteMarker` – Em um bucket com versionamento ativado, um marcador de exclusão sem versões não atuais é chamado de marcador de exclusão de objeto expirado. Você pode usar essa ação de ciclo de vida para instruir o S3 a remover os marcadores de exclusão de objeto expirado. Para ver um exemplo, consulte [Exemplo 7: remoção de marcadores de exclusão de objetos expirados \(p. 144\)](#).

#### Note

Você não pode especificar essa ação de ciclo de vida em uma regra que especifica um filtro com base em tags de objeto.

## Como o Amazon S3 calcula quanto tempo um objeto ficou desatualizado

Em um bucket com versionamento ativado, você pode ter várias versões de um objeto, sempre ter uma versão atual e nenhuma ou mais versões desatualizadas. Sempre que você faz upload de um objeto, a versão atual é retida como a versão não atual e a versão recém-adicionada, a sucessora, se torna a versão atual. Para determinar o número de dias em que um objeto fica desatualizado, o Amazon S3 verifica a data de criação de seu sucessor. O Amazon S3 usa o número de dias desde que seu sucessor foi criado como o número de dias em que um objeto fica desatualizado.

### Restauração de versões anteriores de um objeto ao usar configurações de ciclo de vida

Como explicado em detalhes no tópico [Restauração de versões anteriores \(p. 466\)](#), você pode usar qualquer um dos dois métodos seguintes para recuperar versões anteriores de um objeto:

1. Copiando uma versão não atual do objeto no mesmo bucket. O objeto copiado torna-se a versão atual desse objeto e todas as versões são preservadas.
2. Excluindo permanentemente a versão atual do objeto. Ao excluir a versão atual do objeto, você acaba transformando a versão não atual na versão atual do objeto.

Ao usar regras de configuração de ciclo de vida com buckets com versionamento ativado, recomendamos como melhores práticas o uso do primeiro método.

Devido à semântica de consistência eventual do Amazon S3, uma versão atual excluída permanentemente talvez não desapareça até que as alterações sejam propagadas (o Amazon S3 pode não ter conhecimento da exclusão). Entretanto, a regra de ciclo de vida configurada para expirar objetos não atuais pode remover, permanentemente, objetos não atuais, incluindo aquele que você deseja restaurar. Assim, copiar a versão antiga, como recomendado no primeiro método, é uma alternativa mais confiável.

## Regras de ciclo de vida: com base na idade de um objeto

É possível especificar um período, em número de dias desde a criação (ou modificação) dos objetos, no qual o Amazon S3 pode realizar a ação.

Quando você especificar o número de dias nas ações `Transition` e `Expiration` em uma configuração de ciclo de vida, observe o seguinte:

- Trata-se do número de dias desde a criação de objeto quando a ação ocorrerá.
- O Amazon S3 calcula o tempo, adicionando o número de dias especificado na regra ao momento de criação do objeto e arredondando o tempo resultante para a meia-noite UTC do próximo dia. Por exemplo, se um objeto foi criado em 1/15/2014 10:30 AM UTC e você especificar 3 dias em uma regra de transição, a data de transição do objeto será calculada como 1/19/2014 00:00 UTC.

### Note

O Amazon S3 mantém apenas a data da última modificação para cada objeto. Por exemplo, o console do Amazon S3 mostra a data `Last Modified` (Última modificação) no painel `Properties` (Propriedades) do objeto. Quando você cria inicialmente um novo objeto, essa data reflete a data em que o objeto é criado. Se você substituir o objeto, a data será alterada conforme necessário. Assim, o termo data de criação é sinônimo do termo data da última modificação.

Ao especificar o número de dias nas ações `NoncurrentVersionTransition` e `NoncurrentVersionExpiration` em uma configuração de ciclo de vida, observe o seguinte:

- O Amazon S3 executará a ação nos objetos especificados a partir do número de dias em que a versão do objeto passar a ser desatualizada (ou seja, quando o objeto for substituído ou excluído).
- O Amazon S3 calcula o tempo, adicionando o número de dias especificado na regra ao momento em que a nova versão sucessora do objeto é criada e arredondando o tempo resultante para a meia-noite UTC do próximo dia. Por exemplo, no seu bucket, suponha que a versão atual de um objeto foi criada em 01/01/2014, 10:30 AM UTC. Se a nova versão do objeto que substitui a atual tiver sido criada em 15/01/2014, 10:30 AM UTC, e você especificar uma regra de transição de três dias, a data de transição do objeto será calculada como 19/01/2014, 00:00 UTC.

## Regras de ciclo de vida: com base em uma data especificada

Ao especificar uma ação em uma regra de ciclo de vida, é possível especificar uma data em que deseja que o Amazon S3 realize a ação. Quando a data especificada chegar, o S3 aplicará a ação a todos os objetos qualificados (com base nos critérios de filtro).

Se você especificar uma ação de ciclo de vida com uma data no passado, todos os objetos qualificados estarão imediatamente qualificados para essa ação de ciclo de vida.

### Important

A ação com base em data não é uma ação única. O S3 continuará aplicando a ação com base em data mesmo após a data ter passado, contanto que o status da regra seja Ativado.

Por exemplo, imagine que você especifique uma ação de expiração com base em data para excluir todos os objetos (supondo que nenhum filtro seja especificado na regra). Na data especificada, o S3 expira todos os objetos no bucket. O S3 também continuará expirando todos os objetos novos que você criar no bucket. Para parar a ação de ciclo de vida, você deve remover a ação da configuração de ciclo de vida, desabilitar a regra ou excluir a regra da configuração de ciclo de vida.

O valor de data deve estar em conformidade com o formato ISO 8601. A hora é sempre meia-noite (UTC).

Note

Você não pode criar regras de ciclo de vida com base em data usando o console do Amazon S3, mas pode visualizar, desabilitar ou excluir essas regras.

## Exemplos de configuração de ciclo de vida

Esta seção apresenta exemplos de configuração de ciclo de vida. Cada exemplo mostra como você pode especificar o XML em cada um dos cenários de exemplo.

Tópicos

- [Exemplo 1: especificação de um filtro \(p. 137\)](#)
- [Exemplo 2: desativação de uma regra de ciclo de vida \(p. 139\)](#)
- [Exemplo 3: rebaixamento de uma classe de armazenamento pela duração do ciclo de vida de um objeto \(p. 140\)](#)
- [Exemplo 4: especificação de várias regras \(p. 140\)](#)
- [Exemplo 5: sobreposição de filtros, ações de ciclo de vida conflitantes e o que o Amazon S3 faz \(p. 141\)](#)
- [Exemplo 6: especificação de uma regra de ciclo de vida para um bucket com versionamento habilitado \(p. 144\)](#)
- [Exemplo 7: remoção de marcadores de exclusão de objetos expirados \(p. 144\)](#)
- [Exemplo 8: configuração de ciclo de vida para anular multipart uploads \(p. 146\)](#)

### Exemplo 1: especificação de um filtro

Cada regra de ciclo de vida inclui um filtro que pode ser usado para identificar um subconjunto de objetos em seu bucket ao qual a regra de ciclo de vida se aplica. As configurações de ciclo de vida a seguir mostram exemplos de como você pode especificar um filtro.

- Nesta regra de configuração de ciclo de vida, o filtro especifica um prefixo de chave (`tax/`). Portanto, a regra aplica-se a objetos com o prefixo de nome de chave `tax/`, como `tax/doc1.txt` e `tax/doc2.txt`

A regra especifica duas ações que direcionam o Amazon S3 a fazer o seguinte:

- Faça a transição de objetos para a classe de armazenamento GLACIER 365 dias (um ano) após a criação.
- Exclua objetos (a ação `Expiration`) 3650 dias (10 anos) após a criação.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Transition and Expiration Rule</ID>
    <Filter>
      <Prefix>tax/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
```

```
<Days>365</Days>
<StorageClass>GLACIER</StorageClass>
</Transition>
<Expiration>
<Days>3650</Days>
</Expiration>
</Rule>
</LifecycleConfiguration>
```

Em vez de especificar a idade do objeto em termos de dias após a criação, você pode especificar uma data para cada ação. No entanto, você não pode usar Date e Days na mesma regra.

- Se você quiser que a regra de ciclo de vida se aplique a todos os objetos no bucket, especifique um prefixo vazio. Na configuração a seguir, a regra especifica uma ação Transition levando o Amazon S3 a fazer a transição de objetos para a classe de armazenamento GLACIER 0 dias após a criação e, nesse caso, os objetos são qualificados para arquivamento no Glacier à meia-noite UTC após a criação.

```
<LifecycleConfiguration>
<Rule>
<ID>Archive all object same-day upon creation</ID>
<Filter>
<Prefix></Prefix>
</Filter>
<Status>Enabled</Status>
<Transition>
<Days>0</Days>
<StorageClass>GLACIER</StorageClass>
</Transition>
</Rule>
</LifecycleConfiguration>
```

- Você pode especificar zero ou um prefixo de nome de chave ou zero ou mais tags de objeto em um filtro. O exemplo de código a seguir aplica a regra de ciclo de vida a um subconjunto de objetos com prefixo de chave tax/ e a objetos que têm duas tags com uma chave e valor específicos. Observe que, ao especificar mais de um filtro, você deve incluir AND conforme mostrado (o Amazon S3 aplica um AND lógico para combinar as condições de filtro especificadas).

```
...
<Filter>
<And>
<Prefix>tax/</Prefix>
<Tag>
<Key>key1</Key>
<Value>value1</Value>
</Tag>
<Tag>
<Key>key2</Key>
<Value>value2</Value>
</Tag>
</And>
</Filter>
...
```

- Você pode filtrar objetos com base apenas nas tags. Por exemplo, a regra de ciclo de vida a seguir aplica-se a objetos que tenham duas tags especificadas (não especifica nenhum prefixo):

```
...
<Filter>
<And>
<Tag>
<Key>key1</Key>
<Value>value1</Value>
</Tag>
...
```

```
</Tag>
<Tag>
  <Key>key2</Key>
  <Value>value2</Value>
</Tag>
</And>
</Filter>
...

```

#### Important

Quando você tem várias regras em uma configuração de ciclo de vida, um objeto pode se tornar qualificado para várias ações de ciclo de vida. Nesses casos, as regras gerais que o Amazon S3 são:

- A exclusão permanente tem precedência sobre a transição.
- A transição tem precedência sobre a criação de marcadores de exclusão.
- Quando um objeto está qualificado para uma transição para as classes GLACIER e STANDARD\_IA (ou ONEZONE\_IA), o Amazon S3 escolhe a transição para GLACIER.

Para ver exemplos, consulte [Exemplo 5: sobreposição de filtros, ações de ciclo de vida conflitantes e o que o Amazon S3 faz](#) (p. 141)

## Exemplo 2: desativação de uma regra de ciclo de vida

Você pode desabilitar temporariamente uma regra de ciclo de vida. A configuração de ciclo de vida a seguir especifica duas regras:

- Na regra 1 o Amazon S3 faz a transição de objetos com o prefixo logs/ para a classe de armazenamento GLACIER logo após a criação.
- Na regra 2 o Amazon S3 faz a transição de objetos com o prefixo documents/ para a classe de armazenamento GLACIER logo após a criação.

Na política, a Regra 1 está habilitada e a Regra 2 está desativada. O Amazon S3 não executará nenhuma ação com base em regras desativadas.

```
<LifecycleConfiguration>
<Rule>
  <ID>Rule1</ID>
  <Filter>
    <Prefix>logs/</Prefix>
  </Filter>
  <Status>Enabled</Status>
  <Transition>
    <Days>0</Days>
    <StorageClass>GLACIER</StorageClass>
  </Transition>
</Rule>
<Rule>
  <ID>Rule2</ID>
  <Prefix>documents/</Prefix>
  <Status>Disabled</Status>
  <Transition>
    <Days>0</Days>
    <StorageClass>GLACIER</StorageClass>
  </Transition>
</Rule>
```

```
</LifecycleConfiguration>
```

## Exemplo 3: rebaixamento de uma classe de armazenamento pela duração do ciclo de vida de um objeto

Neste exemplo, você aproveita a configuração de ciclo de vida para rebaixar a classe de armazenamento de objetos pela sua vida útil. O rebaixamento pode ajudar a reduzir os custos de armazenamento. Para obter mais informações sobre definição de preço, consulte [Definição de preço do Amazon S3](#).

A configuração de ciclo de vida a seguir especifica uma regra que se aplica a objetos com o prefixo de nome de chave `logs/`. A regra especifica as seguintes ações:

- Duas ações de transição:
  - A transição de objetos para a classe de armazenamento STANDARD\_IA 30 dias após a criação.
  - A transição de objetos para a classe de armazenamento GLACIER 90 dias após a criação.
- Uma ação de expiração que leva o Amazon S3 a excluir esses objetos um ano após a criação.

```
<LifecycleConfiguration>
  <Rule>
    <ID>example-id</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>30</Days>
      <StorageClass>STANDARD_IA</StorageClass>
    </Transition>
    <Transition>
      <Days>90</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
    <Expiration>
      <Days>365</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

### Note

Você pode usar uma regra para descrever todas as ações de ciclo de vida se todas as ações se aplicarem ao mesmo conjunto de objetos (identificados pelo filtro). Caso contrário, você pode adicionar várias regras com cada uma especificando um filtro diferente.

## Exemplo 4: especificação de várias regras

Você pode especificar várias regras se quiser diferentes ações de ciclo de vida de diferentes objetos. A configuração de ciclo de vida a seguir tem duas regras:

- A regra 1 se aplica a objetos com prefixo de nome de chave `classA/`. Com ela, o Amazon S3 faz a transição de objetos para a classe de armazenamento GLACIER um ano após a criação e expira esses objetos 10 anos após a criação.
- A regra 2 se aplica a objetos com prefixo de nome de chave `classB/`. Com ela o Amazon S3 faz a transição de objetos para a classe de armazenamento STANDARD\_IA 90 dias após a criação e os exclui um ano após a criação.

```
<LifecycleConfiguration>
  <Rule>
    <ID>ClassADocRule</ID>
    <Filter>
      <Prefix>classA/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>365</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
    <Expiration>
      <Days>3650</Days>
    </Expiration>
  </Rule>
  <Rule>
    <ID>ClassBDocRule</ID>
    <Filter>
      <Prefix>classB/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>90</Days>
      <StorageClass>STANDARD_IA</StorageClass>
    </Transition>
    <Expiration>
      <Days>365</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

## Exemplo 5: sobreposição de filtros, ações de ciclo de vida conflitantes e o que o Amazon S3 faz

Você pode especificar uma configuração de ciclo de vida em que especifica prefixos ou ações sobrepostos. Os exemplos a seguir mostram como o Amazon S3 resolve possíveis conflitos.

### Example 1: prefixos sobrepostos (nenhum conflito)

O exemplo de configuração a seguir tem duas regras especificando prefixos sobrepostos da seguinte maneira:

- A primeira regra especifica um filtro vazio, indicando todos os objetos no bucket.
- A segunda regra especifica um prefixo de nome de chave logs/ indicando somente um subconjunto de objetos.

A regra 1 solicita que o Amazon S3 exclua todos os objetos um ano após a criação, e a regra 2 solicita que o Amazon S3 faça a transição de um subconjunto de objetos para a classe de armazenamento STANDARD\_IA 30 dias após a criação.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Rule 1</ID>
    <Filter>
    </Filter>
    <Status>Enabled</Status>
    <Expiration>
      <Days>365</Days>
    </Expiration>
  </Rule>
```

```
<Rule>
  <ID>Rule 2</ID>
  <Filter>
    <Prefix>logs/</Prefix>
  </Filter>
  <Status>Enabled</Status>
  <Transition>
    <StorageClass>STANDARD_IA<StorageClass>
    <Days>30</Days>
  </Transition>
</Rule>
</LifecycleConfiguration>
```

#### Example 2: ações de ciclo de vida conflitantes

Nessa configuração de exemplo, há duas regras que levam o Amazon S3 a executar ao mesmo tempo duas diferentes ações no mesmo conjunto de objetos na vida útil do objeto:

- Ambas as regras especificam o mesmo prefixo de nome de chave, de modo que ambas as regras se aplicam ao mesmo conjunto de objetos.
- Ambas as regras especificam os mesmos 365 dias após a criação de objeto quando as regras se aplicam.
- Uma regra leva o Amazon S3 a fazer a transição de objetos para a classe de armazenamento STANDARD\_IA e outra regra quer que o Amazon S3 expire os objetos ao mesmo tempo.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Rule 1</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Expiration>
      <Days>365</Days>
    </Expiration>
  </Rule>
  <Rule>
    <ID>Rule 2</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <StorageClass>STANDARD_IA<StorageClass>
      <Days>365</Days>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

Nesse caso, como você deseja que os objetos sejam expirados (removidos), não faz sentido mudar a classe de armazenamento, e o Amazon S3 simplesmente escolhe a ação de expiração desses objetos.

#### Example 3: prefixes sobrepostos que resultam em ações de ciclo de vida conflitantes

Neste exemplo, a configuração tem duas regras que especificam prefixes sobrepostos da seguinte maneira:

- A regra 1 especifica um prefixo vazio (indicando todos os objetos).
- A regra 2 especifica um prefixo de nome de chave (logs/) que identifica um subconjunto de todos os objetos.

Para o subconjunto de objetos com o prefixo de nome de chave logs/, as ações de ciclo de vida em ambas as regras se aplicam. Uma regra que leva o Amazon S3 a fazer a transição de objetos 10 dias após a criação e outra regra em que o Amazon S3 faz a transição de objetos 365 dias após a criação.

```
<LifecycleConfiguration>
<Rule>
  <ID>Rule 1</ID>
  <Filter>
    <Prefix></Prefix>
  </Filter>
  <Status>Enabled</Status>
  <Transition>
    <StorageClass>STANDARD_IA<StorageClass>
    <Days>10</Days>
  </Transition>
</Rule>
<Rule>
  <ID>Rule 2</ID>
  <Filter>
    <Prefix>logs/</Prefix>
  </Filter>
  <Status>Enabled</Status>
  <Transition>
    <StorageClass>STANDARD_IA<StorageClass>
    <Days>365</Days>
  </Transition>
</Rule>
</LifecycleConfiguration>
```

Nesse caso, o Amazon S3 escolhe fazer a transição dos objetos 10 dias após a criação.

#### Example 4: filtragem baseada em tags e ações de ciclo de vida conflitantes resultantes

Suponha que você tenha a seguinte política de ciclo de vida que tem duas regras, cada uma especificando um filtro de tag:

- A regra 1 especifica um filtro baseado em tag (tag1/value1). Essa regra leva o Amazon S3 a fazer a transição de objetos para a classe de armazenamento GLACIER 365 dias após a criação.
- A regra 2 especifica um filtro baseado em tag (tag2/value2). Essa regra leva o Amazon S3 a expirar objetos 14 dias após a criação.

A configuração de ciclo de vida é apresentada como segue:

```
<LifecycleConfiguration>
<Rule>
  <ID>Rule 1</ID>
  <Filter>
    <Tag>
      <Key>tag1</Key>
      <Value>value1</Value>
    </Tag>
  </Filter>
  <Status>Enabled</Status>
  <Transition>
    <StorageClass>GLACIER<StorageClass>
    <Days>365</Days>
  </Transition>
</Rule>
<Rule>
  <ID>Rule 2</ID>
  <Filter>
```

```
<Tag>
  <Key>tag2</Key>
  <Value>value1</Value>
</Tag>
</Filter>
<Status>Enabled</Status>
<Expiration>
  <Days>14</Days>
</Expiration>
</Rule>
</LifecycleConfiguration>
```

A política está correta, mas se houver um objeto com ambas as tags, o S3 precisará decidir o que fazer. Em outras palavras, as duas regras se aplicam a um objeto e, na realidade, você está levando o Amazon S3 a executar ações conflitantes. Nesse caso, o Amazon S3 expira o objeto 14 dias após a criação. O objeto é removido e, portanto, a ação de transição não acontece.

## Exemplo 6: especificação de uma regra de ciclo de vida para um bucket com versionamento habilitado

Suponha que você tenha um bucket com versionamento habilitado, o que significa que, para cada objeto, há uma versão atual e zero ou mais versões desatualizadas. Você deseja manter o equivalente a um ano de histórico e, em seguida, excluir as versões desatualizadas. Para obter mais informações sobre versionamento, consulte [Versionamento de objeto \(p. 111\)](#).

Além disso, você deseja economizar os custos de armazenamento movendo as versões desatualizadas para GLACIER 30 dias depois de se tornarem desatualizadas (supondo que são dados antigos que você não precisa acessar em tempo real). Além disso, você também espera que a frequência de acesso das versões atuais diminua 90 dias após a criação para poder mover esses objetos para a classe de armazenamento STANDARD\_IA.

```
<LifecycleConfiguration>
  <Rule>
    <ID>sample-rule</ID>
    <Filter>
      <Prefix></Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>90</Days>
      <StorageClass>STANDARD_IA</StorageClass>
    </Transition>
    <NoncurrentVersionTransition>
      <NoncurrentDays>30</NoncurrentDays>
      <StorageClass>GLACIER</StorageClass>
    </NoncurrentVersionTransition>
    <NoncurrentVersionExpiration>
      <NoncurrentDays>365</NoncurrentDays>
    </NoncurrentVersionExpiration>
  </Rule>
</LifecycleConfiguration>
```

## Exemplo 7: remoção de marcadores de exclusão de objetos expirados

Um bucket com versionamento habilitado mantém uma versão atual e nenhuma ou mais versões desatualizadas de cada objeto. Ao excluir um objeto, observe o seguinte:

- Se você não especificar um ID de versão na solicitação de exclusão, o Amazon S3 adicionará um marcador de exclusão em vez de excluir o objeto. A versão atual do objeto se torna desatualizada e o marcador de exclusão se torna a versão atual.
- Se você não especificar um ID de versão na solicitação de exclusão, o Amazon S3 excluirá permanentemente a versão do objeto (um marcador de exclusão não é criado).
- Um marcador de exclusão sem versões desatualizadas é chamado de marcador de exclusão do objeto expirado.

Este exemplo mostra um cenário que pode criar marcadores de exclusão de objetos expirados em seu bucket e como você pode usar a configuração de ciclo de vida para que o Amazon S3 remova os marcadores de exclusão de objetos expirados.

Suponha que você elabore uma política de ciclo de vida que especifique a ação `NoncurrentVersionExpiration` para remover as versões desatualizadas 30 dias após se tornarem desatualizadas como mostrado a seguir:

```
<LifecycleConfiguration>
  <Rule>
    ...
    <NoncurrentVersionExpiration>
      <NoncurrentDays>30</NoncurrentDays>
    </NoncurrentVersionExpiration>
  </Rule>
</LifecycleConfiguration>
```

A ação `NoncurrentVersionExpiration` se aplica somente à versão atual do objeto. Ela remove apenas versões desatualizadas.

Para versões atuais do objeto, você tem as opções abaixo para gerenciar o ciclo de vida. Tudo vai depender de as versões atuais seguirem ou não um ciclo de vida bem definido.

- As versões atuais do objeto seguem um ciclo de vida bem definido.

Neste caso, você pode usar a política de ciclo de vida com a ação `Expiration` para que o Amazon S3 remova as versões atuais, conforme exibido no seguinte exemplo:

```
<LifecycleConfiguration>
  <Rule>
    ...
    <Expiration>
      <Days>60</Days>
    </Expiration>
    <NoncurrentVersionExpiration>
      <NoncurrentDays>30</NoncurrentDays>
    </NoncurrentVersionExpiration>
  </Rule>
</LifecycleConfiguration>
```

O Amazon S3 remove as versões atuais 60 dias após serem criadas adicionando um marcador de exclusão para cada uma das versões atuais do objeto. Isso torna a versão atual do objeto desatualizada e o marcador de exclusão se torna a versão atual. Para obter mais informações, consulte [Usar versionamento \(p. 448\)](#).

A ação `NoncurrentVersionExpiration` na mesma configuração de ciclo de vida remove os objetos desatualizados 30 dias após se tornarem desatualizados. Assim, todas as versões do objeto são removidas e você tem marcadores de exclusão de objeto expirado, mas o Amazon S3 detecta e remove os marcadores de exclusão de objeto expirado.

- As versões atuais do objeto não têm um ciclo de vida bem definido.

Neste caso, você pode remover os objetos manualmente quando não forem mais necessários criando um marcador de exclusão com uma ou mais versões desatualizadas. Se a configuração de ciclo de vida com a ação `NoncurrentVersionExpiration` remover todas as versões desatualizadas, agora você terá marcadores de exclusão de objetos expirados.

Especificamente para este cenário, a configuração de ciclo de vida do Amazon S3 fornece uma ação `Expiration` em que você pode solicitar que o Amazon S3 remova os marcadores de exclusão de objetos expirados:

```
<LifecycleConfiguration>
  <Rule>
    <ID>Rule 1</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Expiration>
      <ExpiredObjectDeleteMarker>true</ExpiredObjectDeleteMarker>
    </Expiration>
    <NoncurrentVersionExpiration>
      <NoncurrentDays>30</NoncurrentDays>
    </NoncurrentVersionExpiration>
  </Rule>
</LifecycleConfiguration>
```

Ao definir o elemento `ExpiredObjectDeleteMarker` como verdadeiro na ação `Expiration`, você faz o Amazon S3 remover os marcadores de exclusão de objetos expirados.

**Note**

Ao especificar a ação `ExpiredObjectDeleteMarker` de ciclo de vida, a regra não pode especificar um filtro baseado em tag.

## Exemplo 8: configuração de ciclo de vida para anular multipart uploads

Você pode usar a API de multipart upload para fazer upload de objetos grandes em partes. Para obter mais informações sobre multipart uploads, consulte [Visão geral do multipart upload \(p. 181\)](#).

Usando a configuração de ciclo de vida, você pode levar o Amazon S3 a interromper multipart uploads incompletos (identificados pelo prefixo de nome de chave especificado na regra) se eles não forem concluídos dentro de um número especificado de dias após a inicialização. Quando o Amazon S3 interrompe um multipart upload, ele exclui todas as partes associadas ao multipart upload. Isso garante que você não tenha multipart uploads incompletos com partes que estão armazenadas no Amazon S3 e, portanto, você não tem que pagar nada por custo de armazenamento para essas partes.

**Note**

Ao especificar a ação `AbortIncompleteMultipartUpload` de ciclo de vida, a regra não pode especificar um filtro baseado em tag.

Veja a seguir um exemplo de configuração de ciclo de vida que especifica uma regra com a ação `AbortIncompleteMultipartUpload`. Essa ação solicita que o Amazon S3 interrompa multipart uploads incompletos sete dias após a inicialização.

```
<LifecycleConfiguration>
  <Rule>
```

```
<ID>sample-rule</ID>
<Filter>
    <Prefix>SomeKeyPrefix</Prefix>
</Filter>
<Status>rule-status</Status>
<AbortIncompleteMultipartUpload>
    <DaysAfterInitiation>7</DaysAfterInitiation>
</AbortIncompleteMultipartUpload>
</Rule>
</LifecycleConfiguration>
```

## Definir a configuração do ciclo de vida em um bucket

### Tópicos

- [Gerenciar o ciclo de vida de um objeto usando o console do Amazon S3 \(p. 147\)](#)
- [Definir as configurações do ciclo de vida usando a AWS CLI \(p. 148\)](#)
- [Gerenciar os ciclos de vida do objeto usando o AWS SDK for Java \(p. 150\)](#)
- [Gerenciar o ciclo de vida de um objeto usando o AWS SDK para .NET \(p. 152\)](#)
- [Gerenciar o ciclo de vida de um objeto usando o AWS SDK para Ruby \(p. 155\)](#)
- [Gerenciar o ciclo de vida de um objeto usando a API REST \(p. 155\)](#)

Esta seção explica como definir configurações do ciclo de vida em um bucket por programação usando AWS SDKs, ou usando o console do Amazon S3 ou a AWS CLI. Observe o seguinte:

- Quando você adiciona uma configuração do ciclo de vida a um bucket, costuma haver algum atraso antes que uma configuração nova ou atualizada do ciclo de vida seja totalmente propagada para todos os sistemas do Amazon S3. Considere um atraso de alguns minutos antes que a configuração do ciclo de vida entre totalmente em vigor. Esse atraso também pode ocorrer quando você exclui uma configuração do ciclo de vida.
- Quando você desabilita ou exclui uma regra do ciclo de vida, depois de um pequeno atraso o Amazon S3 para de agendar a exclusão ou transição de novos objetos. Todos os objetos que já haviam sido agendados serão removidos do cronograma e não serão excluídos ou migrados.
- Quando você adiciona uma configuração do ciclo de vida a um bucket, as regras de configuração se aplicam aos objetos existentes e aos objetos que serão adicionados no futuro. Por exemplo, se você adicionar uma regra de configuração do ciclo de vida hoje com uma ação de expiração que faz com que os objetos com um prefixo específico expirem 30 dias após sua criação, o Amazon S3 organizará para exclusão todos os objetos existentes com mais de 30 dias.
- Pode haver um atraso entre o momento em que as regras de configuração do ciclo de vida são satisfeitas e o momento em que a ação, ativada pela satisfação da regra, é tomada. No entanto, as alterações na cobrança acontecem assim que a regra de configuração do ciclo de vida é satisfeita, mesmo que a ação ainda não tenha sido tomada. Um exemplo é a não cobrança por armazenamento após o período de expiração do objeto, mesmo que o objeto não seja excluído imediatamente. Outro exemplo é a cobrança das taxas de armazenamento do Glacier assim que o tempo de transição do objeto termina, mesmo que o objeto não seja migrado para o Glacier imediatamente.

Para obter informações sobre a configuração do ciclo de vida, consulte [Gerenciamento do ciclo de vida de objetos \(p. 122\)](#).

## Gerenciar o ciclo de vida de um objeto usando o console do Amazon S3

Especifique regras do ciclo de vida em um bucket usando o console do Amazon S3.

Para obter instruções sobre como configurar regras do ciclo de vida usando o Console de gerenciamento da AWS, consulte [Como faço para criar uma política do ciclo de vida para um bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

## Definir as configurações do ciclo de vida usando a AWS CLI

Use os comandos da AWS CLI a seguir para gerenciar as configurações do ciclo de vida:

- `put-bucket-lifecycle-configuration`
- `get-bucket-lifecycle-configuration`
- `delete-bucket-lifecycle`

Para obter instruções de configuração da AWS CLI, consulte [Configurar a CLI da AWS \(p. 645\)](#).

Observe que a configuração do ciclo de vida do Amazon S3 é um arquivo XML. No entanto, ao usar a CLI, você não pode especificar o XML. Em vez disso, especifique o JSON. A seguir, exemplos de configurações do ciclo de vida em XML e o JSON equivalente que pode ser especificado no comando da CLI da AWS:

- Considere o seguinte exemplo de configuração do ciclo de vida:

```
<LifecycleConfiguration>
    <Rule>
        <ID>ExampleRule</ID>
        <Filter>
            <Prefix>documents/</Prefix>
        </Filter>
        <Status>Enabled</Status>
        <Transition>
            <Days>365</Days>
            <StorageClass>GLACIER</StorageClass>
        </Transition>
        <Expiration>
            <Days>3650</Days>
        </Expiration>
    </Rule>
</LifecycleConfiguration>
```

O JSON equivalente é mostrado:

```
{
    "Rules": [
        {
            "Filter": {
                "Prefix": "documents/"
            },
            "Status": "Enabled",
            "Transitions": [
                {
                    "Days": 365,
                    "StorageClass": "GLACIER"
                }
            ],
            "Expiration": {
                "Days": 3650
            },
            "ID": "ExampleRule"
        }
    ]
}
```

- Considere o seguinte exemplo de configuração do ciclo de vida:

```
<LifecycleConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">  <Rule>    <ID>id-1</ID>    <Expiration>      <Days>1</Days>    </Expiration>    <Filter>      <And>        <Prefix>myprefix</Prefix>        <Tag>          <Key>mytagkey1</Key>          <Value>mytagvalue1</Value>        </Tag>        <Tag>          <Key>mytagkey2</Key>          <Value>mytagvalue2</Value>        </Tag>      </And>    </Filter>    <Status>Enabled</Status>  </Rule></LifecycleConfiguration>
```

O JSON equivalente é mostrado:

```
{  "Rules": [    {      "ID": "id-1",      "Filter": {        "And": {          "Prefix": "myprefix",          "Tags": [            {              "Value": "mytagvalue1",              "Key": "mytagkey1"            },            {              "Value": "mytagvalue2",              "Key": "mytagkey2"            }          ]        }      },      "Status": "Enabled",      "Expiration": {        "Days": 1      }    }  ]}
```

Teste o comando `put-bucket-lifecycle-configuration` da seguinte forma:

1. Salve a configuração do ciclo de vida JSON em um arquivo (`lifecycle.json`).
2. Execute o comando da AWS CLI a seguir para definir a configuração do ciclo de vida no bucket:

```
$ aws s3api put-bucket-lifecycle-configuration \
--bucket bucketname \
```

```
--lifecycle-configuration file://lifecycle.json
```

3. Para verificar, recupere a configuração do ciclo de vida usando o comando `get-bucket-lifecycle-configuration` da AWS CLI, da seguinte forma:

```
$ aws s3api get-bucket-lifecycle-configuration \
--bucket bucketname
```

4. Para excluir a configuração do ciclo de vida use o comando `delete-bucket-lifecycle` da AWS CLI, da seguinte forma:

```
aws s3api delete-bucket-lifecycle \
--bucket bucketname
```

## Gerenciar os ciclos de vida do objeto usando o AWS SDK for Java

Use o AWS SDK for Java para gerenciar a configuração do ciclo de vida de um bucket. Para obter mais informações sobre o gerenciamento da configuração do ciclo de vida, consulte [Gerenciamento do ciclo de vida de objetos \(p. 122\)](#).

### Note

Quando você adiciona uma configuração do ciclo de vida em um bucket, o Amazon S3 substitui a configuração do ciclo de vida atual do bucket, se houver. Para atualizar uma configuração, recupere a configuração, faça as alterações desejadas e, em seguida, adicione a configuração revisada ao bucket.

### Example

O exemplo a seguir mostra como usar o AWS SDK for Java para adicionar, atualizar e excluir a configuração do ciclo de vida de um bucket. O exemplo faz o seguinte:

- Adiciona a configuração do ciclo de vida a um bucket.
- Recupera a configuração do ciclo de vida e a atualiza adicionando outra regra.
- Adiciona a configuração do ciclo de vida ao bucket, o Amazon S3 substitui a configuração do ciclo de vida existente.
- Recupera a configuração novamente e verifica se ela tem o número certo de regras pela impressão do número de regras.
- Exclui a configuração do ciclo de vida e verifica se ela foi excluída ao tentar recuperá-la novamente.

Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

import java.io.IOException;
import java.util.Arrays;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
```

```
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.BucketLifecycleConfiguration;
import com.amazonaws.services.s3.model.BucketLifecycleConfiguration.Transition;
import com.amazonaws.services.s3.model.StorageClass;
import com.amazonaws.services.s3.model.Tag;
import com.amazonaws.services.s3.model.lifecycle.LifecycleAndOperator;
import com.amazonaws.services.s3.model.lifecycle.LifecycleFilter;
import com.amazonaws.services.s3.model.lifecycle.LifecyclePrefixPredicate;
import com.amazonaws.services.s3.model.lifecycle.LifecycleTagPredicate;

public class LifecycleConfiguration {

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

        // Create a rule to archive objects with the "glacierobjects/" prefix to Glacier
        // immediately.
        BucketLifecycleConfiguration.Rule rule1 = new BucketLifecycleConfiguration.Rule()
            .withId("Archive immediately rule")
            .withFilter(new LifecycleFilter(new
                LifecyclePrefixPredicate("glacierobjects/")))
            .addTransition(new
                Transition().withDays(0).withStorageClass(StorageClass.Glacier))
            .withStatus(BucketLifecycleConfiguration.ENABLED);

        // Create a rule to transition objects to the Standard-Infrequent Access storage
        // class
        // after 30 days, then to Glacier after 365 days. Amazon S3 will delete the objects
        // after 3650 days.
        // The rule applies to all objects with the tag "archive" set to "true".
        BucketLifecycleConfiguration.Rule rule2 = new BucketLifecycleConfiguration.Rule()
            .withId("Archive and then delete rule")
            .withFilter(new LifecycleFilter(new LifecycleTagPredicate(new
                Tag("archive", "true"))))
            .addTransition(new
                Transition().withDays(30).withStorageClass(StorageClass.StandardInfrequentAccess))
            .addTransition(new
                Transition().withDays(365).withStorageClass(StorageClass.Glacier))
            .withExpirationInDays(3650)
            .withStatus(BucketLifecycleConfiguration.ENABLED);

        // Add the rules to a new BucketLifecycleConfiguration.
        BucketLifecycleConfiguration configuration = new BucketLifecycleConfiguration()
            .withRules(Arrays.asList(rule1, rule2));

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

            // Save the configuration.
            s3Client.setBucketLifecycleConfiguration(bucketName, configuration);

            // Retrieve the configuration.
            configuration = s3Client.getBucketLifecycleConfiguration(bucketName);

            // Add a new rule with both a prefix predicate and a tag predicate.
            configuration.getRules().add(new
                BucketLifecycleConfiguration.Rule().withId("NewRule")
                    .withFilter(new LifecycleFilter(new LifecycleAndOperator(
                        Arrays.asList(new LifecyclePrefixPredicate("YearlyDocuments/"),
                            new LifecycleTagPredicate(new Tag("expire_after",
                                "ten_years"))))))
                    .withExpirationInDays(3650)

```

```
        .withStatus(BucketLifecycleConfiguration.ENABLED));

    // Save the configuration.
    s3Client.setBucketLifecycleConfiguration(bucketName, configuration);

    // Retrieve the configuration.
    configuration = s3Client.getBucketLifecycleConfiguration(bucketName);

    // Verify that the configuration now has three rules.
    configuration = s3Client.getBucketLifecycleConfiguration(bucketName);
    System.out.println("Expected # of rules = 3; found: " +
configuration.getRules().size());

    // Delete the configuration.
    s3Client.deleteBucketLifecycleConfiguration(bucketName);

    // Verify that the configuration has been deleted by attempting to retrieve it.
    configuration = s3Client.getBucketLifecycleConfiguration(bucketName);
    String s = (configuration == null) ? "No configuration found." : "Configuration
found.";
    System.out.println(s);
}
catch(AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
}
catch(SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
}
```

## Gerenciar o ciclo de vida de um objeto usando o AWS SDK para .NET

Use o AWS SDK para .NET para gerenciar as configurações do ciclo de vida em um bucket. Para obter mais informações sobre o gerenciamento da configuração do ciclo de vida, consulte [Gerenciamento do ciclo de vida de objetos \(p. 122\)](#).

### Note

Quando você adiciona uma configuração do ciclo de vida, o Amazon S3 substitui a configuração do ciclo de vida existente no bucket especificado. Para atualizar uma configuração, recupere a configuração do ciclo de vida, faça as alterações e, em seguida, adicione a configuração revisada ao bucket.

### Example Exemplo de código .NET

O exemplo a seguir mostra como usar o AWS SDK para .NET para adicionar, atualizar e excluir uma configuração do ciclo de vida de um bucket. O exemplo de código faz o seguinte:

- Adiciona a configuração do ciclo de vida a um bucket.
- Recupera a configuração do ciclo de vida e a atualiza adicionando outra regra.
- Adiciona a configuração do ciclo de vida ao bucket, o Amazon S3 substitui a configuração do ciclo de vida existente.
- Recupera a configuração novamente e a verifica imprimindo o número de regras na configuração.
- Exclui a configuração do ciclo de vida e verifica a exclusão

Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-
// developer-guide/blob/master/LICENSE-SAMPLECODE.)

using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class LifecycleTest
    {
        private const string bucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 client;
        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            AddUpdateDeleteLifecycleConfigAsync().Wait();
        }

        private static async Task AddUpdateDeleteLifecycleConfigAsync()
        {
            try
            {
                var lifeCycleConfiguration = new LifecycleConfiguration()
                {
                    Rules = new List<LifecycleRule>
                    {
                        new LifecycleRule
                        {
                            Id = "Archive immediately rule",
                            Filter = new LifecycleFilter()
                            {
                                LifecycleFilterPredicate = new
LifecyclePrefixPredicate()
                            },
                            Prefix = "glacierobjects/"
                        },
                        Status = LifecycleRuleStatus.Enabled,
                        Transitions = new List<LifecycleTransition>
                        {
                            new LifecycleTransition
                            {
                                Days = 0,
                                StorageClass = S3StorageClass.Glacier
                            }
                        },
                    },
                    new LifecycleRule
                    {
                        Id = "Archive and then delete rule",
                        Filter = new LifecycleFilter()
                        {
                            LifecycleFilterPredicate = new
LifecyclePrefixPredicate()
                        },
                        Prefix = "projectdocs/"
                    }
                };
            
```

```
        }
    },
    Status = LifecycleRuleStatus.Enabled,
    Transitions = new List<LifecycleTransition>
    {
        new LifecycleTransition
        {
            Days = 30,
            StorageClass =
S3StorageClass.StandardInfrequentAccess
        },
        new LifecycleTransition
        {
            Days = 365,
            StorageClass = S3StorageClass.Glacier
        }
    },
    Expiration = new LifecycleRuleExpiration()
    {
        Days = 3650
    }
}
};

// Add the configuration to the bucket.
await AddExampleLifecycleConfigAsync(client, lifeCycleConfiguration);

// Retrieve an existing configuration.
lifeCycleConfiguration = await RetrieveLifecycleConfigAsync(client);

// Add a new rule.
lifeCycleConfiguration.Rules.Add(new LifecycleRule
{
    Id = "NewRule",
    Filter = new LifecycleFilter()
    {
        LifecycleFilterPredicate = new LifecyclePrefixPredicate()
        {
            Prefix = "YearlyDocuments/"
        }
    },
    Expiration = new LifecycleRuleExpiration()
    {
        Days = 3650
    }
});

// Add the configuration to the bucket.
await AddExampleLifecycleConfigAsync(client, lifeCycleConfiguration);

// Verify that there are now three rules.
lifeCycleConfiguration = await RetrieveLifecycleConfigAsync(client);
Console.WriteLine("Expected # of rules=3; found:{0}",
lifeCycleConfiguration.Rules.Count);

// Delete the configuration.
await RemoveLifecycleConfigAsync(client);

// Retrieve a nonexistent configuration.
lifeCycleConfiguration = await RetrieveLifecycleConfigAsync(client);

}
catch (AmazonS3Exception e)
{
```

```
        Console.WriteLine("Error encountered ***. Message:'{0}' when writing an
object", e.Message);
    }
    catch (Exception e)
    {
        Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
    }
}

static async Task AddExampleLifecycleConfigAsync(IAmazonS3 client,
LifecycleConfiguration configuration)
{
    PutLifecycleConfigurationRequest request = new PutLifecycleConfigurationRequest
    {
        BucketName = bucketName,
        Configuration = configuration
    };
    var response = await client.PutLifecycleConfigurationAsync(request);
}

static async Task<LifecycleConfiguration> RetrieveLifecycleConfigAsync(IAmazonS3
client)
{
    GetLifecycleConfigurationRequest request = new GetLifecycleConfigurationRequest
    {
        BucketName = bucketName
    };
    var response = await client.GetLifecycleConfigurationAsync(request);
    var configuration = response.Configuration;
    return configuration;
}

static async Task RemoveLifecycleConfigAsync(IAmazonS3 client)
{
    DeleteLifecycleConfigurationRequest request = new
DeleteLifecycleConfigurationRequest
    {
        BucketName = bucketName
    };
    await client.DeleteLifecycleConfigurationAsync(request);
}
}
```

## Gerenciar o ciclo de vida de um objeto usando o AWS SDK para Ruby

Você pode usar o AWS SDK para Ruby para gerenciar a configuração do ciclo de vida em um bucket com a classe [AWS::S3::BucketLifecycleConfiguration](#). Para obter mais informações sobre o uso do AWS SDK para Ruby com o Amazon S3, consulte [Usar o AWS SDK para Ruby - versão 3 \(p. 650\)](#). Para obter mais informações sobre o gerenciamento da configuração do ciclo de vida, consulte [Gerenciamento do ciclo de vida de objetos \(p. 122\)](#).

## Gerenciar o ciclo de vida de um objeto usando a API REST

Você pode usar o Console de gerenciamento da AWS para definir a configuração do ciclo de vida no bucket. Se o seu aplicativo exigir, você também pode enviar solicitações REST diretamente. As seções a seguir no Amazon Simple Storage Service API Reference descrevem a API REST relacionada à configuração do ciclo de vida.

- Ciclo de vida de PUT Bucket
- Ciclo de vida de GET Bucket
- DELETE Bucket lifecycle

## Cross-Origin Resource Sharing (CORS, Compartilhamento de recursos de origem cruzada)

O compartilhamento de recursos de origem cruzada (CORS) define uma maneira de os aplicativos web clientes carregados em um domínio interagirem com recursos em outro domínio. Com suporte ao CORS, você pode criar aplicativos web avançados do lado do cliente com o Amazon S3 e permitir seletivamente o acesso de origem cruzada aos recursos do Amazon S3.

Esta seção fornece uma visão geral do CORS. Os subtópicos descrevem como você pode habilitar o CORS usando o console do Amazon S3 ou, programaticamente, usando a API REST do Amazon S3 e os SDKs da AWS.

### Tópicos

- [Compartilhamento de recursos de origem cruzada: cenários de caso de uso \(p. 156\)](#)
- [Como faço para configurar CORS no meu bucket? \(p. 156\)](#)
- [Como o Amazon S3 avalia a configuração de CORS em um bucket? \(p. 159\)](#)
- [Ativação do compartilhamento de recursos de origem cruzada \(CORS\) \(p. 159\)](#)
- [Solução de problemas do CORS \(p. 165\)](#)

## Compartilhamento de recursos de origem cruzada: cenários de caso de uso

Veja a seguir exemplos de cenário de uso do CORS:

- Cenário 1: suponha que você esteja hospedando um site em um bucket do Amazon S3 chamado `website`, como descrito em [Hospedagem de um site estático no Amazon S3 \(p. 494\)](#). Os usuários carregam o endpoint de site `http://website.s3-website-us-east-1.amazonaws.com`. Agora você quer usar JavaScript nas páginas da web armazenadas nesse bucket para fazer solicitações GET e PUT autenticadas no mesmo bucket usando o endpoint da API do Amazon S3 para o bucket `website.s3.amazonaws.com`. Um navegador normalmente impediria que o JavaScript permitisse essas solicitações. No entanto, com CORS, é possível configurar seu bucket para permitir explicitamente solicitações de origem cruzada de `website.s3-website-us-east-1.amazonaws.com`.
- Cenário 2: suponha que você queira hospedar uma fonte web de seu bucket do S3. Mais uma vez, os navegadores exigem uma verificação de CORS (também chamada de verificação de simulação) para carregar fontes web. Assim, é preciso configurar o bucket que está hospedando a fonte web para permitir que qualquer origem faça essas solicitações.

## Como faço para configurar CORS no meu bucket?

Para configurar seu bucket para permitir solicitações de origem cruzada, crie uma configuração de CORS, que é um documento XML com regras que identificam as origens que poderão acessar seu bucket, as operações (métodos HTTP) que oferecerão suporte para cada origem e outras informações específicas da operação.

Você pode adicionar até 100 regras à configuração. Adicione o documento XML como o sub-recurso `cors` ao bucket programaticamente ou usando o console do Amazon S3. Para obter mais informações, consulte [Ativação do compartilhamento de recursos de origem cruzada \(CORS\) \(p. 159\)](#).

Em vez de acessar um site usando um endpoint do Amazon S3, você pode usar seu próprio domínio, como `example1.com`, para distribuir seu conteúdo. Para obter informações sobre como usar seu próprio domínio, consulte [Exemplo: configurar um site estático usando um domínio personalizado \(p. 511\)](#). A configuração de exemplo `cors` a seguir tem três regras especificadas como elementos `CORSRule`:

- A primeira regra permite solicitações `PUT`, `POST` e `DELETE` de origem cruzada da origem `http://www.example1.com`. A regra também permite todos os cabeçalhos em uma solicitação `OPTIONS` de simulação por meio do cabeçalho `Access-Control-Request-Headers`. Em resposta a solicitações `OPTIONS` de simulação, o Amazon S3 retorna cabeçalhos solicitados.
- A segunda regra permite as mesmas solicitações de origem cruzada da primeira regra, mas a regra se aplica a outra origem, `http://www.example2.com`.
- A terceira regra permite solicitações `GET` de origem cruzada de todas as origens. O caractere curinga `*` refere-se a todas as origens.

```
<CORSConfiguration>
  <CORSRule>
    <AllowedOrigin>http://www.example1.com</AllowedOrigin>

    <AllowedMethod>PUT</AllowedMethod>
    <AllowedMethod>POST</AllowedMethod>
    <AllowedMethod>DELETE</AllowedMethod>

    <AllowedHeader>*</AllowedHeader>
  </CORSRule>
  <CORSRule>
    <AllowedOrigin>http://www.example2.com</AllowedOrigin>

    <AllowedMethod>PUT</AllowedMethod>
    <AllowedMethod>POST</AllowedMethod>
    <AllowedMethod>DELETE</AllowedMethod>

    <AllowedHeader>*</AllowedHeader>
  </CORSRule>
  <CORSRule>
    <AllowedOrigin>*</AllowedOrigin>
    <AllowedMethod>GET</AllowedMethod>
  </CORSRule>
</CORSConfiguration>
```

A configuração de CORS também permite parâmetros de configuração opcionais, conforme exibido na configuração de CORS a seguir. Neste exemplo, a configuração de CORS permite solicitações `PUT`, `POST` e `DELETE` de origem cruzada da origem `http://www.example.com`.

```
<CORSConfiguration>
  <CORSRule>
    <AllowedOrigin>http://www.example.com</AllowedOrigin>
    <AllowedMethod>PUT</AllowedMethod>
    <AllowedMethod>POST</AllowedMethod>
    <AllowedMethod>DELETE</AllowedMethod>
    <AllowedHeader>*</AllowedHeader>
    <MaxAgeSeconds>3000</MaxAgeSeconds>
    <ExposeHeader>x-amz-server-side-encryption</ExposeHeader>
    <ExposeHeader>x-amz-request-id</ExposeHeader>
    <ExposeHeader>x-amz-id-2</ExposeHeader>
```

```
</CORSRule>  
</CORSConfiguration>
```

O elemento `CORSRule` na configuração anterior inclui os seguintes elementos opcionais:

- `MaxAgeSeconds` — especifica o tempo em segundos (neste exemplo, 3000) que o navegador armazena em cache uma resposta do Amazon S3 a uma solicitação OPTIONS de simulação para o recurso especificado. Armazenando a resposta em cache, o navegador não precisará enviar solicitações de simulação ao Amazon S3 caso a solicitação original seja repetida.
- `ExposeHeader` — Identifica os cabeçalhos de resposta (neste exemplo, `x-amz-server-side-encryption`, `x-amz-request-id` e `x-amz-id-2`) que os clientes podem acessar de seus aplicativos (por exemplo, de um objeto JavaScript XMLHttpRequest).

## Elemento AllowedMethod

Na configuração de CORS, você pode especificar os seguintes valores para o elemento `AllowedMethod`.

- GET
- PUT
- POST
- DELETE
- HEAD

## Elemento AllowedOrigin

No elemento `AllowedOrigin`, você especifica as origens das quais deseja permitir solicitações de domínio cruzado, por exemplo, `http://www.example.com`. A string de origem pode conter somente um caractere curinga \*, como `http://*.example.com`. É possível especificar \* como a origem para permitir que todas as origens enviem solicitações de origem cruzada. Você também pode especificar `https` para permitir somente origens confiáveis.

## Elemento AllowedHeader

O elemento `AllowedHeader` especifica quais cabeçalhos são permitidos em uma solicitação de simulação por meio do cabeçalho `Access-Control-Request-Headers`. Cada nome no cabeçalho `Access-Control-Request-Headers` deve corresponder a uma entrada correspondente na regra. O Amazon S3 enviará somente os cabeçalhos permitidos em uma resposta que tiver sido solicitada. Para ver uma lista de cabeçalhos que podem ser usados no Amazon S3, acesse [Cabeçalhos de solicitação comuns](#) no guia Amazon Simple Storage Service API Reference.

Cada string de `AllowedHeader` na regra pode conter no máximo um caractere curinga \*. Por exemplo, `<AllowedHeader>x-amz-*</AllowedHeader>` permitirá todos os cabeçalhos específicos da Amazon.

## Elemento ExposeHeader

Cada elemento `ExposeHeader` identifica um cabeçalho na resposta que você deseja que os clientes acessem de seus aplicativos (por exemplo, de um objeto JavaScript XMLHttpRequest). Para ver uma lista de cabeçalhos de resposta comuns do Amazon S3, acesse [Cabeçalhos de resposta comuns](#) no guia Amazon Simple Storage Service API Reference.

## Elemento MaxAgeSeconds

O elemento `MaxAgeSeconds` especifica o tempo em segundos que seu navegador pode armazenar em cache a resposta para uma solicitação de simulação conforme identificado pelo recurso, pelo método HTTP e pela origem.

## Como o Amazon S3 avalia a configuração de CORS em um bucket?

Quando o Amazon S3 recebe uma solicitação de simulação de um navegador, ele avalia a configuração de CORS para o bucket e usa a primeira regra `CORSRule` que corresponde à solicitação de entrada do navegador para permitir uma solicitação de origem cruzada. Para que uma regra seja correspondente, as seguintes condições devem ser satisfeitas:

- O cabeçalho `Origin` da solicitação deve corresponder a um elemento `AllowedOrigin`.
- O método de solicitação (por exemplo, GET ou PUT) ou o cabeçalho `Access-Control-Request-Method` no caso da solicitação de simulação OPTIONS deve ser um dos elementos `AllowedMethod`.
- Cada cabeçalho listado no cabeçalho `Access-Control-Request-Headers` da solicitação de simulação deve corresponder a um elemento `AllowedHeader`.

### Note

As ACLs e políticas continuam sendo aplicadas quando você ativa o CORS no bucket.

## Ativação do compartilhamento de recursos de origem cruzada (CORS)

Ative o compartilhamento de recursos de origem cruzada definindo uma configuração de CORS em seu bucket usando o Console de gerenciamento da AWS, a API REST ou os SDKs da AWS.

### Tópicos

- [Habilitar o compartilhamento de recursos de origem cruzada \(CORS\) usando o Console de gerenciamento da AWS \(p. 159\)](#)
- [Habilitar o compartilhamento de recursos de origem cruzada \(CORS\) usando o AWS SDK for Java \(p. 160\)](#)
- [Habilitar o compartilhamento de recursos de origem cruzada \(CORS\) usando o AWS SDK para .NET \(p. 162\)](#)
- [Ativação do compartilhamento de recursos de origem cruzada \(CORS\) usando a API REST \(p. 165\)](#)

## Habilitar o compartilhamento de recursos de origem cruzada (CORS) usando o Console de gerenciamento da AWS

Use o Console de gerenciamento da AWS para definir uma configuração de CORS no bucket. Para obter instruções, consulte [Como ativo o compartilhamento de recursos entre domínios com CORS?](#) no Guia do usuário do console do Amazon Simple Storage Service.

## Habilitar o compartilhamento de recursos de origem cruzada (CORS) usando o AWS SDK for Java

Você pode usar o AWS SDK for Java para gerenciar o compartilhamento de recursos de origem cruzada (CORS) para um bucket. Para obter mais informações sobre CORS, consulte [Cross-Origin Resource Sharing \(CORS, Compartilhamento de recursos de origem cruzada\) \(p. 156\)](#).

### Example

O exemplo a seguir:

- Cria uma configuração do CORS e define a configuração em um bucket
- Recupera a configuração e a altera adicionando uma regra
- Adiciona a configuração modificada ao bucket
- Exclui a configuração

Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

import java.io.IOException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.BucketCrossOriginConfiguration;
import com.amazonaws.services.s3.model.CORSRule;

public class CORS {

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

        // Create two CORS rules.
        List<CORSRule.AllowedMethods> rule1AM = new ArrayList<CORSRule.AllowedMethods>();
        rule1AM.add(CORSRule.AllowedMethods.PUT);
        rule1AM.add(CORSRule.AllowedMethods.POST);
        rule1AM.add(CORSRule.AllowedMethods.DELETE);
        CORSRule rule1 = new CORSRule().withId("CORSRule1").withAllowedMethods(rule1AM)
            .withAllowedOrigins(Arrays.asList(new String[] { "http://
*.example.com" }));

        List<CORSRule.AllowedMethods> rule2AM = new ArrayList<CORSRule.AllowedMethods>();
        rule2AM.add(CORSRule.AllowedMethods.GET);
        CORSRule rule2 = new CORSRule().withId("CORSRule2").withAllowedMethods(rule2AM)
            .withAllowedOrigins(Arrays.asList(new String[]
{ "*" }).withMaxAgeSeconds(3000)
            .withExposedHeaders(Arrays.asList(new String[] { "x-amz-server-side-
encryption" })));

        List<CORSRule> rules = new ArrayList<CORSRule>();
```

```
rules.add(rule1);
rules.add(rule2);

// Add the rules to a new CORS configuration.
BucketCrossOriginConfiguration configuration = new
BucketCrossOriginConfiguration();
configuration.setRules(rules);

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

    // Add the configuration to the bucket.
    s3Client.setBucketCrossOriginConfiguration(bucketName, configuration);

    // Retrieve and display the configuration.
    configuration = s3Client.getBucketCrossOriginConfiguration(bucketName);
    printCORSConfiguration(configuration);

    // Add another new rule.
    List<CORSRule.AllowedMethods> rule3AM = new
ArrayList<CORSRule.AllowedMethods>();
    rule3AM.add(CORSRule.AllowedMethods.HEAD);
    CORSRule rule3 = new CORSRule().withId("CORSRule3").withAllowedMethods(rule3AM)
        .withAllowedOrigins(Arrays.asList(new String[] { "http://
www.example.com" }));
    rules = configuration.getRules();
    rules.add(rule3);
    configuration.setRules(rules);
    s3Client.setBucketCrossOriginConfiguration(bucketName, configuration);

    // Verify that the new rule was added by checking the number of rules in the
    configuration.
    configuration = s3Client.getBucketCrossOriginConfiguration(bucketName);
    System.out.println("Expected # of rules = 3, found " +
configuration.getRules().size());

    // Delete the configuration.
    s3Client.deleteBucketCrossOriginConfiguration(bucketName);
    System.out.println("Removed CORS configuration.");

    // Retrieve and display the configuration to verify that it was
    // successfully deleted.
    configuration = s3Client.getBucketCrossOriginConfiguration(bucketName);
    printCORSConfiguration(configuration);
}

catch(AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
}
catch(SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}

private static void printCORSConfiguration(BucketCrossOriginConfiguration
configuration) {
    if (configuration == null) {
        System.out.println("Configuration is null.");
    } else {
```

```
        System.out.println("Configuration has " + configuration.getRules().size() + " rules\n");

        for (CORSRule rule : configuration.getRules()) {
            System.out.println("Rule ID: " + rule.getId());
            System.out.println("MaxAgeSeconds: " + rule.getMaxAgeSeconds());
            System.out.println("AllowedMethod: " + rule.getAllowedMethods());
            System.out.println("AllowedOrigins: " + rule.getAllowedOrigins());
            System.out.println("AllowedHeaders: " + rule.getAllowedHeaders());
            System.out.println("ExposeHeader: " + rule.getExposedHeaders());
            System.out.println();
        }
    }
}
```

## Habilitar o compartilhamento de recursos de origem cruzada (CORS) usando o AWS SDK para .NET

Para gerenciar o compartilhamento de recursos de origem cruzada (CORS) para um bucket, você pode usar o AWS SDK para .NET. Para obter mais informações sobre CORS, consulte [Cross-Origin Resource Sharing \(CORS, Compartilhamento de recursos de origem cruzada\) \(p. 156\)](#).

### Example

O seguinte código C#:

- Cria uma configuração do CORS e define a configuração em um bucket
- Recupera a configuração e a altera adicionando uma regra
- Adiciona a configuração modificada ao bucket
- Exclui a configuração

Para obter informações sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

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

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

```
private static async Task CORSConfigTestAsync()
{
    try
    {
        // Create a new configuration request and add two rules
        CORSConfiguration configuration = new CORSConfiguration
        {
            Rules = new System.Collections.Generic.List<CORSRule>
            {
                new CORSRule
                {
                    Id = "CORSRule1",
                    AllowedMethods = new List<string> {"PUT", "POST", "DELETE"},
                    AllowedOrigins = new List<string> {"http://*.example.com"}
                },
                new CORSRule
                {
                    Id = "CORSRule2",
                    AllowedMethods = new List<string> {"GET"},
                    AllowedOrigins = new List<string> {"*"},
                    MaxAgeSeconds = 3000,
                    ExposeHeaders = new List<string> {"x-amz-server-side-
encryption"}
                }
            };
        }

        // Add the configuration to the bucket.
        await PutCORSConfigurationAsync(configuration);

        // Retrieve an existing configuration.
        configuration = await RetrieveCORSConfigurationAsync();

        // Add a new rule.
        configuration.Rules.Add(new CORSRule
        {
            Id = "CORSRule3",
            AllowedMethods = new List<string> { "HEAD" },
            AllowedOrigins = new List<string> { "http://www.example.com" }
        });

        // Add the configuration to the bucket.
        await PutCORSConfigurationAsync(configuration);

        // Verify that there are now three rules.
        configuration = await RetrieveCORSConfigurationAsync();
        Console.WriteLine();
        Console.WriteLine("Expected # of rules=3; found:{0}",
        configuration.Rules.Count);
        Console.WriteLine();
        Console.WriteLine("Pause before configuration delete. To continue, click
Enter...");  

        Console.ReadKey();

        // Delete the configuration.
        await DeleteCORSConfigurationAsync();

        // Retrieve a nonexistent configuration.
        configuration = await RetrieveCORSConfigurationAsync();
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine("Error encountered on server. Message:'{0}' when writing
an object", e.Message);
    }
    catch (Exception e)
```

```
        {
            Console.WriteLine("Unknown encountered on server. Message:{0}' when
writing an object", e.Message);
        }
    }

    static async Task PutCORSConfigurationAsync(CORSConfiguration configuration)
    {

        PutCORSConfigurationRequest request = new PutCORSConfigurationRequest
        {
            BucketName = bucketName,
            Configuration = configuration
        };

        var response = await s3Client.PutCORSConfigurationAsync(request);
    }

    static async Task<CORSConfiguration> RetrieveCORSConfigurationAsync()
    {
        GetCORSConfigurationRequest request = new GetCORSConfigurationRequest
        {
            BucketName = bucketName
        };

        var response = await s3Client.GetCORSConfigurationAsync(request);
        var configuration = response.Configuration;
        PrintCORSRules(configuration);
        return configuration;
    }

    static async Task DeleteCORSConfigurationAsync()
    {
        DeleteCORSConfigurationRequest request = new DeleteCORSConfigurationRequest
        {
            BucketName = bucketName
        };
        await s3Client.DeleteCORSConfigurationAsync(request);
    }

    static void PrintCORSRules(CORSConfiguration configuration)
    {
        Console.WriteLine();

        if (configuration == null)
        {
            Console.WriteLine("\nConfiguration is null");
            return;
        }

        Console.WriteLine("Configuration has {0} rules:", configuration.Rules.Count);
        foreach (CORSRule rule in configuration.Rules)
        {
            Console.WriteLine("Rule ID: {0}", rule.Id);
            Console.WriteLine("MaxAgeSeconds: {0}", rule.MaxAgeSeconds);
            Console.WriteLine("AllowedMethod: {0}", string.Join(", ",
rule.AllowedMethods.ToArray()));
            Console.WriteLine("AllowedOrigins: {0}", string.Join(", ",
rule.AllowedOrigins.ToArray()));
            Console.WriteLine("AllowedHeaders: {0}", string.Join(", ",
rule.AllowedHeaders.ToArray()));
            Console.WriteLine("ExposeHeader: {0}", string.Join(", ",
rule.ExposeHeaders.ToArray()));
        }
    }
}
```

}

## Ativação do compartilhamento de recursos de origem cruzada (CORS) usando a API REST

Para definir uma configuração de CORS no bucket, use o Console de gerenciamento da AWS. Se o seu aplicativo exigir, você também pode enviar solicitações REST diretamente. As seções a seguir no Amazon Simple Storage Service API Reference descrevem as ações da API REST relacionadas à configuração de CORS:

- [PUT bucket cors](#)
- [Cors de GET Bucket](#)
- [DELETE bucket cors](#)
- [Objeto OPTIONS](#)

## Solução de problemas do CORS

Se você encontrar um comportamento inesperado ao acessar os buckets definidos com a configuração de CORS, tente usar as etapas a seguir para resolver o problema:

1. Verifique se a configuração de CORS está definida no bucket.

Para obter instruções, consulte [Editar permissões de bucket](#) no Guia do usuário do console do Amazon Simple Storage Service. Se a configuração de CORS estiver definida, o console exibirá um link Edit CORS Configuration (Editar configuração de CORS) na seção Permissions (Permissões) do bucket Properties (Propriedades).

2. Capture a solicitação e a resposta completas usando uma ferramenta de sua escolha. Para cada solicitação que o Amazon S3 recebe, deve existir uma regra de CORS que corresponda aos dados na solicitação, da seguinte maneira:

- a. Verifique se a solicitação tem o cabeçalho de origem.

Se o cabeçalho estiver ausente, o Amazon S3 não tratará a solicitação como uma solicitação de origem cruzada e não enviará cabeçalhos de resposta de CORS na resposta.

- b. Verifique se o cabeçalho de origem na solicitação corresponde a pelo menos um dos elementos `AllowedOrigin` na `CORSRule` especificada.

O esquema, o host e os valores de porta no cabeçalho da solicitação de origem devem corresponder a elementos `AllowedOrigin` na `CORSRule`. Por exemplo, se você tiver definido a `CORSRule` para permitir a origem `http://www.example.com`, as origens `https://www.example.com` e `http://www.example.com:80` da solicitação não corresponderão à origem permitida na configuração.

- c. Verifique se o método na solicitação (ou, em uma solicitação de simulação, o método especificado em `Access-Control-Request-Method`) é um dos elementos `AllowedMethod` na mesma `CORSRule`.

- d. Para uma solicitação de simulação, se a solicitação incluir um cabeçalho `Access-Control-Request-Headers`, verifique se `CORSRule` inclui as entradas `AllowedHeader` para cada valor no cabeçalho `Access-Control-Request-Headers` header.

## Operações em objetos

O Amazon S3 permite que você armazene, recupere e exclua objetos. Você pode recuperar um objeto inteiro ou parte de um objeto. Se o seu bucket tem versionamento habilitado, você pode recuperar uma

versão específica do objeto. Você também pode recuperar um sub-recurso associado com seu objeto e atualizá-lo onde for aplicável. Você pode fazer uma cópia do seu objeto existente. Dependendo do tamanho de objeto, as seguintes considerações relacionadas a upload e cópia são aplicáveis:

- Fazer upload de objetos —você pode fazer upload de objetos de até 5 GB em uma única operação. Para objetos maiores do que 5 GB, você deve usar a API de multipart upload.

Usando a API multipart upload, você pode fazer upload de objetos de até 5 TB cada um. Para obter mais informações, consulte [Upload de objetos usando a API de multipart upload \(p. 181\)](#).

- Cópia de objetos —a operação de cópia cria uma cópia de um objeto que já está armazenado no Amazon S3.

Você pode criar uma cópia do objeto de até 5 GB em uma única operação atômica. Contudo, para copiar objetos maiores do que 5 GB, você deve usar a API de multipart upload. Para obter mais informações, consulte [Cópia de objetos \(p. 219\)](#).

Você pode usar a API REST (veja [Fazer solicitações usando a API REST \(p. 45\)](#)) para trabalhar com objetos ou usar uma das seguintes bibliotecas do AWS SDK:

- [AWS SDK for Java](#)
- [AWS SDK para .NET](#)
- [AWS SDK para PHP](#)

Essas bibliotecas fornecem uma abstração de alto nível que facilita o trabalho com objetos. Contudo, se o seu aplicativo exigir, você pode usar a API REST diretamente.

## Obtenção de objetos

### Tópicos

- [Recursos relacionados \(p. 167\)](#)
- [Obtenha um objeto usando o AWS SDK for Java \(p. 167\)](#)
- [Obtenha um objeto usando o AWS SDK para .NET \(p. 169\)](#)
- [Obtenha um objeto usando o AWS SDK para PHP \(p. 171\)](#)
- [Obtenha um objeto usando a API REST \(p. 172\)](#)
- [Compartilhe um objeto \(p. 172\)](#)

Você pode recuperar objetos diretamente do Amazon S3. Você tem as seguintes opções para recuperar um objeto:

- Recupere um objeto inteiro —uma única operação GET pode retornar o objeto inteiro armazenado no Amazon S3.
- Recupere o objeto em parte —usando o cabeçalho HTTP Range em uma solicitação GET, você recupera uma faixa específica de bytes em um objeto armazenado no Amazon S3.

Você recomeça a buscar outras partes do objeto sempre que seu aplicativo estiver pronto. Este download retomável é útil quando você precisa apenas de partes dos dados do seu objeto. Ele também é útil onde a conectividade de rede é deficiente e você precisa reagir a falhas.

### Note

O Amazon S3 não suporta a recuperação de múltiplas faixas de dados por solicitação GET.

Quando você recupera um objeto, os metadados são retornados nos cabeçalhos da resposta. Às vezes, você deseja substituir certos valores no cabeçalho da resposta retornados em uma resposta do GET. Por exemplo, você pode substituir o valor `Content-Disposition` no cabeçalho da resposta em sua solicitação GET. O objeto GET da API REST (consulte [Objeto GET](#)) permite que você especifique parâmetros na string da query na sua solicitação GET para substituir esses valores.

Os AWS SDKs para Java, .NET e PHP também fornecem os objetos necessários que você pode usar para especificar valores para esses cabeçalhos na resposta à sua solicitação GET.

Ao recuperar objetos que são armazenados com criptografia usando criptografia do lado do servidor, você precisará fornecer os cabeçalhos apropriados de solicitação. Para obter mais informações, consulte [Proteção de dados usando criptografia \(p. 409\)](#).

## Recursos relacionados

- [Usar os AWS SDKs, a CLI e os Explorers \(p. 639\)](#)

## Obtenha um objeto usando o AWS SDK for Java

Ao fazer download de um objeto com AWS SDK for Java, o Amazon S3 retorna todos os metadados do objeto e um fluxo de entrada de onde ler os conteúdos do objeto.

Para recuperar um objeto, faça o seguinte:

- Execute o método `AmazonS3Client.getObject()`, fornecendo o nome do bucket e a chave de objeto na solicitação.
- Execute um dos métodos de instância `S3Object` para processar o fluxo de entrada.

### Note

A conexão de rede permanece aberta até que você leia todos os dados ou feche o fluxo de entrada. Recomendamos que você leia o conteúdo de fluxo o mais rápido possível.

Veja a seguir algumas variações que você pode usar:

- Em vez de ler o objeto inteiro, você pode ler apenas uma parte dos dados do objeto especificando o intervalo de bytes que você deseja na solicitação.
- Se desejar, é possível substituir os valores do cabeçalho da resposta (consulte [Obtenção de objetos \(p. 166\)](#)) usando um objeto `ResponseHeaderOverrides` e configurando a propriedade da solicitação correspondente. Por exemplo, você pode usar esse recurso para indicar que o objeto deve ser baixado em um arquivo com um nome de arquivo diferente do nome da chave do objeto.

O exemplo a seguir recupera o objeto de um bucket do Amazon S3 de três maneiras: primeiro, como objeto concluído, ou como intervalo de bytes do objeto, ou ainda, como objeto completo com os valores do cabeçalho da resposta substituídos. Para obter mais informações sobre obter objetos do Amazon S3, consulte [GET Object](#). Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.  
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)  
  
import java.io.BufferedReader;  
import java.io.IOException;  
import java.io.InputStream;
```

```
import java.io.InputStream;
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.GetObjectRequest;
import com.amazonaws.services.s3.model.ResponseHeaderOverrides;
import com.amazonaws.services.s3.model.S3Object;

public class GetObject {

    public static void main(String[] args) throws IOException {
        String clientRegion = "*** Client region ***";
        String bucketName = "*** Bucket name ***";
        String key = "*** Object key ***";

        S3Object fullObject = null, objectPortion = null, headerOverrideObject = null;
        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .withRegion(clientRegion)
                .withCredentials(new ProfileCredentialsProvider())
                .build();

            // Get an object and print its contents.
            System.out.println("Downloading an object");
            fullObject = s3Client.getObject(new GetObjectRequest(bucketName, key));
            System.out.println("Content-Type: " +
                fullObject.getObjectMetadata().getContentType());
            System.out.println("Content: ");
            displayTextInputStream(fullObject.getObjectContent());

            // Get a range of bytes from an object and print the bytes.
            GetObjectRequest rangeObjectRequest = new GetObjectRequest(bucketName, key)
                .withRange(0,9);
            objectPortion = s3Client.getObject(rangeObjectRequest);
            System.out.println("Printing bytes retrieved.");
            displayTextInputStream(objectPortion.getObjectContent());

            // Get an entire object, overriding the specified response headers, and print
            // the object's content.
            ResponseHeaderOverrides headerOverrides = new ResponseHeaderOverrides()
                .withCacheControl("No-cache")

            .withContentDisposition("attachment; filename=example.txt");
            GetObjectRequest getObjectRequestHeaderOverride = new
            GetObjectRequest(bucketName, key)

            .withResponseHeaders(headerOverrides);
            headerOverrideObject = s3Client.getObject(getObjectRequestHeaderOverride);
            displayTextInputStream(headerOverrideObject.getObjectContent());
        }
        catch(AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        }
        catch(SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
        finally {
            // To ensure that the network connection doesn't remain open, close any open
            input streams.
        }
    }
}
```

```
        if(fullObject != null) {
            fullObject.close();
        }
        if(objectPortion != null) {
            objectPortion.close();
        }
        if(headerOverrideObject != null) {
            headerOverrideObject.close();
        }
    }
}

private static void displayTextInputStream(InputStream input) throws IOException {
    // Read the text input stream one line at a time and display each line.
    BufferedReader reader = new BufferedReader(new InputStreamReader(input));
    String line = null;
    while ((line = reader.readLine()) != null) {
        System.out.println(line);
    }
    System.out.println();
}
}
```

## Obtenha um objeto usando o AWS SDK para .NET

Ao fazer download de um objeto, você obtém todos os metadados do objeto e um fluxo do qual você lê o conteúdo. Você deve ler conteúdo de fluxo o mais rápido possível porque os dados são trazidos em um fluxo diretamente do Amazon S3 e sua conexão de rede permanecerá aberta até que você leia todos os dados ou feche o fluxo de entrada. Para obter um objeto, faça o seguinte:

- Execute o método `GetObject`, fornecendo o nome do bucket e a chave de objeto na solicitação.
- Execute um dos métodos `GetObjectResponse` para processar o fluxo.

Veja a seguir algumas variações que você pode usar:

- Em vez de ler o objeto inteiro, você pode ler apenas a parte dos dados do objeto especificando o intervalo de bytes na solicitação, conforme exibido no seguinte exemplo do C#:

### Example

```
GetObjectRequest request = new GetObjectRequest
{
    BucketName = bucketName,
    Key = keyName,
    ByteRange = new ByteRange(0, 10)
};
```

- Ao recuperar um objeto, você pode substituir os valores do cabeçalho da resposta (consulte [Obtenção de objetos \(p. 166\)](#)) usando o objeto `ResponseHeaderOverrides` e configurando a propriedade da solicitação correspondente. O exemplo de código C# a seguir mostra como fazer isso. Por exemplo, você pode usar esse recurso para indicar que o objeto deve ser baixado em um arquivo com um nome de arquivo diferente do nome da chave do objeto.

### Example

```
GetObjectRequest request = new GetObjectRequest
{
    BucketName = bucketName,
    Key = keyName
};
```

```
ResponseHeaderOverrides responseHeaders = new ResponseHeaderOverrides();
responseHeaders.CacheControl = "No-cache";
responseHeaders.ContentDisposition = "attachment; filename=testing.txt";

request.ResponseHeaderOverrides = responseHeaders;
```

### Example

O exemplo de código C# a seguir recupera um objeto de um bucket do Amazon S3 especificado. A partir da resposta, o exemplo lê os dados do objeto usando a propriedade `GetObjectResponse.ResponseStream`. O exemplo também mostra como você pode usar a coleção `GetObjectResponse.Metadata` para ler os metadados do objeto. Se o objeto que você recupera têm os metadados `x-amz-meta-title`, o código imprime o valor dos metadados.

Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.IO;
using System.Threading.Tasks;

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

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            ReadObjectDataAsync().Wait();
        }

        static async Task ReadObjectDataAsync()
        {
            string responseBody = "";
            try
            {
                GetObjectRequest request = new GetObjectRequest
                {
                    BucketName = bucketName,
                    Key = keyName
                };
                using (GetObjectResponse response = await client.GetObjectAsync(request))
                using (Stream responseStream = response.ResponseStream)
                using (StreamReader reader = new StreamReader(responseStream))
                {
                    string title = response.Metadata["x-amz-meta-title"]; // Assume you
have "title" as metadata added to the object.
                    string contentType = response.Headers["Content-Type"];
                    Console.WriteLine("Object metadata, Title: {0}", title);
                    Console.WriteLine("Content type: {0}", contentType);
                }
            }
        }
    }
}
```

```
        responseBody = reader.ReadToEnd(); // Now you process the response
body.
    }
}
catch (AmazonS3Exception e)
{
    Console.WriteLine("Error encountered ***. Message:{0}" when writing an
object", e.Message);
}
catch (Exception e)
{
    Console.WriteLine("Unknown encountered on server. Message:{0}" when
writing an object", e.Message);
}
}
}
}
```

## Obtenha um objeto usando o AWS SDK para PHP

Este tópico explica como usar uma classe do AWS SDK para PHP para recuperar um objeto do Amazon S3. Você pode recuperar um objeto inteiro ou um intervalo de bytes do objeto. Partimos do princípio de que você já está seguindo as instruções para [Usar o AWS SDK para PHP e executar exemplos do PHP \(p. 649\)](#) e tem o AWS SDK para PHP devidamente instalado.

Ao recuperar um objeto, você pode substituir os valores do cabeçalho da resposta adicionando as chaves da resposta, `ResponseContentType`, `ResponseContentLanguage`, `ResponseContentDisposition`, `ResponseCacheControl`, e `ResponseExpires`, ao `getObject()` método, conforme exibido no seguinte exemplo de código PHP:

### Example

```
$result = $s3->getObject([
    'Bucket'                  => $bucket,
    'Key'                     => $keyname,
    'ResponseContentType'     => 'text/plain',
    'ResponseContentLanguage' => 'en-US',
    'ResponseContentDisposition' => 'attachment; filename=testing.txt',
    'ResponseCacheControl'    => 'No-cache',
    'ResponseExpires'          => gmdate(DATE_RFC2822, time() + 3600),
]);
```

Para obter mais informações sobre recuperação de objetos, consulte [Obtenção de objetos \(p. 166\)](#).

O exemplo de PHP a seguir recupera um objeto e exibe o conteúdo do objeto no navegador. O exemplo mostra como usar o método `getObject()`. Para obter informações sobre a execução dos exemplos de PHP neste guia, consulte [Executar exemplos do PHP \(p. 649\)](#).

```
<?php

require 'vendor/autoload.php';

use Aws\S3\S3Client;
use Aws\S3\Exception\S3Exception;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

$s3 = new S3Client([
```

```
'version' => 'latest',
'region'  => 'us-east-1'
]);

try {
    // Get the object.
    $result = $s3->getObject([
        'Bucket' => $bucket,
        'Key'     => $keyname
    ]);

    // Display the object in the browser.
    header("Content-Type: {$result['ContentType']}");
    echo $result['Body'];
} catch (S3Exception $e) {
    echo $e->getMessage() . PHP_EOL;
}
```

## Recursos relacionados

- [AWS SDK para PHP para classe Aws\S3\S3Client do Amazon S3](#)
- [Documentação do AWS SDK para PHP](#)

## Obtenha um objeto usando a API REST

Você pode usar o AWS SDK para recuperar chaves de objeto de um bucket. Contudo, se o seu aplicativo exigir, você pode enviar solicitações REST diretamente. Você pode enviar uma solicitação GET para recuperar chaves de objeto. Para obter mais informações sobre o formato de solicitação e de resposta, acesse [Objeto GET](#).

## Compartilhe um objeto

### Tópicos

- [Gerar um pre-signed URL usando o AWS Explorer for Visual Studio \(p. 173\)](#)
- [Gerar um pre-signed URL de objeto usando o AWS SDK for Java \(p. 173\)](#)
- [Gerar um pre-signed URL de objeto usando o AWS SDK para .NET \(p. 174\)](#)

Todos os objetos são privados por padrão. Somente o proprietário do objeto tem permissão para acessar esses objetos. Contudo, o proprietário do objeto pode compartilhar objetos com os outros criando um pre-signed URL, usando suas próprias credenciais de segurança para conceder permissão de prazo limitado para download de objetos.

Quando cria um pre-signed URL para seu objeto, você deve fornecer as credenciais de segurança, especificar um nome de bucket, uma chave de objeto, especificar o método HTTP (GET para download do objeto) e data e hora de expiração. Os pre-signed URLs são válidos apenas pela duração especificada.

Qualquer um que recebe o pre-signed URL pode acessar o objeto. Por exemplo, se você tiver um vídeo em seu bucket e o bucket e o objeto forem privados, será possível compartilhar o vídeo gerando um pre-signed URL.

### Note

Qualquer um com credenciais de segurança válidas pode criar um pre-signed URL. No entanto, para acessar com êxito um objeto, o pre-signed URL deve ter sido criado por alguém que tenha permissão para executar a operação na qual o pre-signed URL está baseado.

Você pode gerar o pre-signed URL com programação usando AWS SDK for Java e .NET.

## Gerar um pre-signed URL usando o AWS Explorer for Visual Studio

Se você estiver usando o Visual Studio, poderá gerar um pre-signed URL para um objeto sem gravar nenhum código usando o AWS Explorer for Visual Studio. Qualquer um com esse URL pode fazer download do objeto. Para obter mais informações, acesse [Usar o Amazon S3 no AWS Explorer](#).

Para instruções sobre como instalar o AWS Explorer, consulte [Usar os AWS SDKs, a CLI e os Explorers \(p. 639\)](#).

## Gerar um pre-signed URL de objeto usando o AWS SDK for Java

### Example

O exemplo a seguir gera um pre-signed URL que você pode compartilhar para recuperar um objeto de um bucket do S3. Para obter mais informações, consulte [Compartilhe um objeto \(p. 172\)](#).

Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

import java.io.IOException;
import java.net.URL;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.HttpMethod;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.GeneratePresignedUrlRequest;

public class GeneratePresignedURL {

    public static void main(String[] args) throws IOException {
        String clientRegion = "*** Client region ***";
        String bucketName = "*** Bucket name ***";
        String objectKey = "*** Object key ***";

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

            // Set the presigned URL to expire after one hour.
            java.util.Date expiration = new java.util.Date();
            long expTimeMillis = expiration.getTime();
            expTimeMillis += 1000 * 60 * 60;
            expiration.setTime(expTimeMillis);

            // Generate the presigned URL.
            System.out.println("Generating pre-signed URL.");
            GeneratePresignedUrlRequest generatePresignedUrlRequest =
                new GeneratePresignedUrlRequest(bucketName, objectKey)
                    .withMethod(HttpMethod.GET)
                    .withExpiration(expiration);
            URL url = s3Client.generatePresignedUrl(generatePresignedUrlRequest);
        }
    }
}
```

```
        System.out.println("Pre-Signed URL: " + url.toString());
    }
    catch(AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    }
    catch(SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
```

## Gerar um pre-signed URL de objeto usando o AWS SDK para .NET

### Example

O exemplo a seguir gera um pre-signed URL que você pode compartilhar para recuperar um objeto. Para obter mais informações, consulte [Compartilhe um objeto \(p. 172\)](#).

Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-
developer-guide/blob/master/LICENSE-SAMPLECODE.)

using Amazon.S3;
using Amazon.S3.Model;
using System;

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

        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            string urlString = GeneratePreSignedURL();
        }
        static string GeneratePreSignedURL()
        {
            string urlString = "";
            try
            {
                GetPreSignedUrlRequest request1 = new GetPreSignedUrlRequest
                {
                    BucketName = bucketName,
                    Key = objectKey,
                    Expires = DateTime.Now.AddMinutes(5)
                };
                urlString = s3Client.GetPreSignedURL(request1);
            }
            catch (AmazonS3Exception e)
            {
```

```
        Console.WriteLine("Error encountered on server. Message:{0}' when writing
an object", e.Message);
    }
    catch (Exception e)
    {
        Console.WriteLine("Unknown encountered on server. Message:{0}' when
writing an object", e.Message);
    }
    return urlString;
}
}
```

## Upload de objetos

Dependendo do tamanho de dados que você está fazendo upload, o Amazon S3 oferece as seguintes opções:

- Upload de objetos em uma única operação —com uma única operação PUT, você pode fazer upload de objetos de até 5 GB.

Para obter mais informações, consulte [Fazer upload de objetos em uma única operação \(p. 176\)](#).

- Faça upload de objetos em partes —ao usar a API multipart upload, você pode fazer upload de objetos grandes, de até 5 TB.

A API multipart upload API foi projetada para melhorar a experiência de upload de objetos maiores.

Você pode fazer upload de objetos em partes. O upload dessas partes de objetos pode ser feito independentemente, em qualquer ordem, e em paralelo. Você pode usar um multipart upload de objetos de 5 MB a 5 TB. Para obter mais informações, consulte [Upload de objetos usando a API de multipart upload \(p. 181\)](#).

Recomendamos que você use o multipart uploader das seguintes formas:

- Se você estiver fazendo upload de grandes objetos em uma rede de banda larga estável, use o multipart upload para maximizar o uso da banda larga disponível, fazendo upload de partes do objeto em paralelo para desempenho com multi-thread.
- Se você estiver fazendo upload em uma rede lenta, use o multipart upload para aumentar a resiliência dos erros de rede, evitando reinícios de upload. Ao usar o multipart upload, tente novamente apenas as partes que foram interrompidas durante o upload. Você não precisa reiniciar o upload do seu objeto do começo.

Para obter mais informações sobre multipart uploads, consulte [Visão geral do multipart upload \(p. 181\)](#).

### Tópicos

- [Fazer upload de objetos em uma única operação \(p. 176\)](#)
- [Upload de objetos usando a API de multipart upload \(p. 181\)](#)
- [Fazer upload de objetos usando pre-signed URLs \(p. 214\)](#)

Ao fazer upload de um objeto, você pode pedir que o Amazon S3 criptografe o objeto antes de salvá-lo no disco e decodificá-lo quando fizer seu upload. Para obter mais informações, consulte [Proteção de dados usando criptografia \(p. 409\)](#).

### Tópicos relacionados

[Usar os AWS SDKs, a CLI e os Explorers \(p. 639\)](#)

## Fazer upload de objetos em uma única operação

### Tópicos

- [Faça upload de objetos usando o AWS SDK for Java \(p. 176\)](#)
- [Faça upload de objetos usando o AWS SDK para .NET \(p. 177\)](#)
- [Faça upload de objetos usando o AWS SDK para PHP \(p. 178\)](#)
- [Faça upload de objetos usando o AWS SDK para Ruby \(p. 179\)](#)
- [Faça upload de um objeto usando a API REST \(p. 180\)](#)

Você pode usar o AWS SDK para fazer upload de objetos. O SDK fornece bibliotecas wrapper para você para fazer upload de dados com facilidade. Contudo, se o seu aplicativo exigir, você pode usar a API REST diretamente em seu aplicativo.

### Faça upload de objetos usando o AWS SDK for Java

#### Example

O exemplo a seguir cria dois objetos. O primeiro objeto tem uma sequência de texto como dados, e o segundo objeto é um arquivo. O exemplo cria o primeiro objeto especificando o nome de bucket, a chave de objeto, e os dados de texto diretamente em uma chamada para `AmazonS3Client.putObject()`. O exemplo cria um segundo objeto usando um `PutObjectRequest` que especifica o nome de bucket, a chave de objeto, e o caminho do arquivo. O `PutObjectRequest` também especifica o cabeçalho de `ContentType` e os metadados do título.

Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

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

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ObjectMetadata;
import com.amazonaws.services.s3.model.PutObjectRequest;

public class UploadObject {

    public static void main(String[] args) throws IOException {
        String clientRegion = "*** Client region ***";
        String bucketName = "*** Bucket name ***";
        String stringObjKeyName = "*** String object key name ***";
        String fileObjKeyName = "*** File object key name ***";
        String fileName = "*** Path to file to upload ***";

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

            // Upload a text string as a new object.
            s3Client.putObject(bucketName, stringObjKeyName, "Uploaded String Object");
        }
    }
}
```

```
// Upload a file as a new object with ContentType and title specified.
PutObjectRequest request = new PutObjectRequest(bucketName, fileObjKeyName, new
File(fileName));
ObjectMetadata metadata = new ObjectMetadata();
metadata.setContentType("plain/text");
metadata.addUserMetadata("x-amz-meta-title", "someTitle");
request.setMetadata(metadata);
s3Client.putObject(request);
}
catch(AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
}
catch(SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

## Faça upload de objetos usando o AWS SDK para .NET

### Example

O exemplo de código C# a seguir cria dois objetos com as duas solicitações PutObjectRequest:

- A primeira solicitação PutObjectRequest salva uma sequência de texto como exemplo de dados do objeto. Ela também especifica os nomes do bucket e da chave de objeto.
- A segunda solicitação PutObjectRequest faz upload de um arquivo especificando o nome do arquivo. Essa solicitação também especifica o cabeçalho ContentType e os metadados opcionais de objeto (título).

Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-
developer-guide/blob/master/LICENSE-SAMPLECODE.)

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

namespace Amazon.DocSamples.S3
{
    class UploadObjectTest
    {
        private const string bucketName = "**** bucket name ****";
        // Example creates two objects (for simplicity, we upload same file twice).
        // You specify key names for these objects.
        private const string keyName1 = "**** key name for first object created ****";
        private const string keyName2 = "**** key name for second object created ****";
        private const string filePath = @">**** file path ****";
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.EUWest1;

        private static IAmazonS3 client;

        public static void Main()
```

```
{  
    client = new AmazonS3Client(bucketRegion);  
    WritingAnObjectAsync().Wait();  
}  
  
static async Task WritingAnObjectAsync()  
{  
    try  
    {  
        // 1. Put object-specify only key name for the new object.  
        var putRequest1 = new PutObjectRequest  
        {  
            BucketName = bucketName,  
            Key = keyName1,  
            ContentBody = "sample text"  
        };  
  
        PutObjectResponse response1 = await client.PutObjectAsync(putRequest1);  
  
        // 2. Put the object-set ContentType and add metadata.  
        var putRequest2 = new PutObjectRequest  
        {  
            BucketName = bucketName,  
            Key = keyName2,  
            FilePath = filePath,  
            ContentType = "text/plain"  
        };  
        putRequest2.Metadata.Add("x-amz-meta-title", "someTitle");  
    }  
    catch (AmazonS3Exception e)  
    {  
        Console.WriteLine(  
            "Error encountered ***. Message:'{0}' when writing an object"  
            , e.Message);  
    }  
    catch (Exception e)  
    {  
        Console.WriteLine(  
            "Unknown encountered on server. Message:'{0}' when writing an object"  
            , e.Message);  
    }  
}
```

## Faça upload de objetos usando o AWS SDK para PHP

Este tópico oriente sobre o uso de classes do AWS SDK para PHP para fazer upload de um objeto de até 5 GB. Para arquivos maiores, você deve usar a API multipart upload. Para obter mais informações, consulte [Upload de objetos usando a API de multipart upload \(p. 181\)](#).

Este tópico pressupõe que você já está seguindo as instruções para [Usar o AWS SDK para PHP e executar exemplos do PHP \(p. 649\)](#) e tem o AWS SDK para PHP devidamente instalado.

Example de criação de um objeto no bucket do Amazon S3 pelos dados de upload

O exemplo PHP a seguir cria um objeto em um bucket especificado pelo upload de dados usando o método `putObject()`. Para obter informações sobre a execução de exemplos PHP neste guia, vá para [Executar exemplos do PHP \(p. 649\)](#).

```
<?php  
require 'vendor/autoload.php';
```

```
use Aws\S3\S3Client;
use Aws\S3\Exception\S3Exception;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

try {
    // Upload data.
    $result = $s3->putObject([
        'Bucket' => $bucket,
        'Key'     => $keyname,
        'Body'    => 'Hello, world!',
        'ACL'     => 'public-read'
    ]);

    // Print the URL to the object.
    echo $result['ObjectURL'] . PHP_EOL;
} catch (S3Exception $e) {
    echo $e->getMessage() . PHP_EOL;
}
```

#### Recursos relacionados

- [AWS SDK para PHP para classe Aws\S3\S3Client do Amazon S3](#)
- [Documentação do AWS SDK para PHP](#)

### Faça upload de objetos usando o AWS SDK para Ruby

O AWS SDK para Ruby - versão 3 tem duas maneiras de fazer upload de um objeto para o Amazon S3. O primeiro usa um upload gerenciado de arquivo, que facilita o upload de arquivos de qualquer tamanho para disco. Usar o método de upload gerenciado de arquivo:

1. Crie uma instância da classe `Aws::S3::Resource`.
2. Faça referência ao objeto de destino pelo nome e chave do bucket. Os objetos residem em um bucket e têm chaves exclusivas que identificam cada objeto.
3. Chame `#upload_file` no objeto.

#### Example

```
require 'aws-sdk-s3'

s3 = Aws::S3::Resource.new(region:'us-west-2')
obj = s3.bucket('bucket-name').object('key')
obj.upload_file('/path/to/source/file')
```

A segunda forma que o AWS SDK para Ruby - Versão 3 pode fazer upload de um objeto usa o método `#put` do `Aws::S3::Object`. Isso é útil se o objeto for uma string ou um objeto de E/S que não seja um arquivo em disco. Para usar este método:

1. Crie uma instância da classe `Aws::S3::Resource`.
2. Faça referência ao objeto de destino pelo nome e chave do bucket.
3. Chame `#put`, passando a sequência ou objeto de E/S.

### Example

```
require 'aws-sdk-s3'

s3 = Aws::S3::Resource.new(region:'us-west-2')
obj = s3.bucket('bucket-name').object('key')

# string data
obj.put(body: 'Hello World!')

# I/O object
File.open('/path/to/source.file', 'rb') do |file|
  obj.put(body: file)
end
```

### Faça upload de um objeto usando a API REST

Você pode usar o AWS SDK para fazer upload de um objeto. Contudo, se o seu aplicativo exigir, você pode enviar solicitações REST diretamente. Você pode enviar uma solicitação PUT para fazer upload de dados em uma única operação. Para obter mais informações, consulte [Objeto PUT](#).

## Upload de objetos usando a API de multipart upload

### Tópicos

- [Visão geral do multipart upload \(p. 181\)](#)
- [Usar o AWS Java SDK para multipart upload \(API de alto nível\) \(p. 188\)](#)
- [Usar o AWS Java SDK para um multipart upload \(API de baixo nível\) \(p. 192\)](#)
- [Usar o AWS SDK para .NET para multipart upload \(API de alto nível\) \(p. 197\)](#)
- [Usar o AWS SDK para .NET para multipart upload \(API de nível baixo\) \(p. 204\)](#)
- [Usar o SDK PHP da AWS para multipart upload \(p. 209\)](#)
- [Usar o SDK PHP da AWS para multipart upload \(API de baixo nível\) \(p. 211\)](#)
- [Usar o AWS SDK para Ruby para multipart upload \(p. 214\)](#)
- [Usar a API REST para multipart upload \(p. 214\)](#)

O multipart upload permite que você faça upload de um único objeto como um conjunto de partes. Cada parte é uma parte contígua de dados do objeto. O upload dessas partes de objetos pode ser feito de maneira independente e em qualquer ordem. Se a transmissão de alguma parte falhar, você poderá retransmitir essa parte sem afetar outras partes. Depois que todas as partes do objeto são carregadas, o Amazon S3 monta essas partes e cria o objeto. Geralmente, quando seu objeto alcança 100 MB de tamanho, você deve considerar o uso de multipart uploads em vez de fazer upload do objeto em uma única operação.

Usar o multipart upload fornece as seguintes vantagens:

- Transferência aprimorada - Você pode fazer upload de partes em paralelo para melhorar a transferência.
- Recuperação rápida de alguns problemas de rede - Partes de tamanho menor minimizam o impacto de reiniciar um upload que tenha falhado devido a um erro de rede.
- Pausar e retomar uploads de objeto - Você pode fazer upload de partes do objeto ao longo do tempo. Uma vez iniciado um multipart upload, não há expiração; você deverá concluir ou anular explicitamente o multipart upload.
- Começar um upload antes de saber o tamanho final do objeto - Você pode fazer upload de um objeto à medida que ele for criado.

Para obter mais informações, consulte [Visão geral do multipart upload \(p. 181\)](#).

## Visão geral do multipart upload

### Tópicos

- [Operações simultâneas de multipart upload \(p. 183\)](#)
- [Multipart upload e definição de preço \(p. 183\)](#)
- [Anular multipart uploads incompletos usando uma política de ciclo de vida de bucket \(p. 183\)](#)
- [Limites do Multipart upload do Amazon S3 \(p. 185\)](#)
- [Suporte de API para multipart upload \(p. 185\)](#)
- [API de multipart upload e permissões \(p. 186\)](#)

A API de multipart upload permite que você faça upload, em partes, de objetos grandes. Você pode usar essa API para fazer upload de novos objetos grandes ou para fazer uma cópia de um objeto existente (consulte [Operações em objetos \(p. 165\)](#)).

O multipart upload é um processo de três etapas: você inicia o upload, faz upload de partes do objeto e, depois de fazer upload de todas as partes, conclui o multipart upload. Ao receber a solicitação de

conclusão do multipart upload, o Amazon S3 cria o objeto a partir das partes carregadas, e você pode então acessar o objeto como qualquer outro objeto em seu bucket.

Você pode listar todos os seus multipart uploads em andamento ou obter uma lista das partes que carregou para um multipart upload específico. Cada uma dessas operações é explicada nesta seção.

#### Iniciação do multipart upload

Quando você envia uma solicitação para iniciar um multipart upload, o Amazon S3 retorna uma resposta com um ID de upload, que é um identificador exclusivo do seu multipart upload. Você deve incluir esse ID de upload sempre que fizer upload de partes, listar as partes, concluir um upload ou anular um upload. Se você desejar fornecer metadados que descrevem o objeto que está sendo carregado, deverá fornecê-los na solicitação para iniciar o multipart upload.

#### Upload de partes

Ao fazer upload de uma parte, além do ID de upload, você deve especificar um número de parte. Você pode escolher qualquer número de parte entre 1 e 10.000. Um número de parte identifica com exclusividade a parte e sua posição no objeto do qual você está fazendo upload. O número de parte que você escolhe não precisa ser uma sequência consecutiva (por exemplo, ele pode ser 1, 5 e 14). Se você fizer upload de uma nova parte usando o mesmo número da parte anteriormente carregada, a parte anteriormente carregada será substituída. Sempre que você faz upload uma parte, o Amazon S3 retorna um cabeçalho ETag na resposta. Para cada upload de parte, você deve registrar o número de parte e o valor de ETag. Você precisa incluir esses valores na solicitação subsequente para concluir o multipart upload.

#### Note

Depois de iniciar um multipart upload e fazer upload de uma ou mais partes, você deve concluir ou anular o multipart upload para parar de ser cobrado pelo armazenamento de peças carregadas. Somente depois que você concluir ou anular um multipart upload é que o Amazon S3 liberará o armazenamento das partes e parará de cobrar pelo armazenamento das partes.

#### Conclusão do multipart upload (ou anulação)

Quando você conclui um multipart upload, o Amazon S3 cria um objeto concatenando as partes em ordem crescente com base no número de parte. Se algum metadado de objeto tiver sido fornecido na solicitação de criação do multipart upload, o Amazon S3 associará esses metadados ao objeto. Depois de uma solicitação de conclusão bem-sucedida, as partes não existem mais. Sua solicitação de conclusão de multipart upload deve incluir o ID do upload e uma lista dos números de parte e os valores ETag correspondentes. A resposta do Amazon S3 inclui uma ETag que identifica de maneira exclusiva os dados do objeto combinado. Esse ETag não será necessariamente um hash MD5 de dados de objeto. Opcionalmente, você pode anular o multipart upload. Depois de anular um multipart upload, você não pode fazer upload de nenhuma parte usando esse ID de upload novamente. Todo armazenamento de partes do multipart upload anulado consumido é liberado. Se algum upload de parte estiver em andamento, ele ainda poderá ser bem-sucedido ou até mesmo falhar depois da anulação. Para liberar todo o armazenamento consumido por todas as partes, você deve anular um multipart upload somente depois que todos os uploads de parte tiverem sido concluídos.

#### Listagens de multipart upload

Você pode listar as partes de um multipart upload específico ou de todos os multipart uploads em andamento. A operação de listagem de partes retorna as informações das partes que você fez upload em um multipart upload específico. Para cada solicitação de listagem de partes, o Amazon S3 retorna informações das partes do multipart upload especificado, até no máximo 1.000 partes. Se houver mais de 1.000 partes no multipart upload, você deverá enviar uma série de solicitações de listagem para recuperar todas as partes. Observe que a lista de partes retornada não inclui partes que não tiveram o upload concluído. Usando a operação listar multipart uploads, você pode obter uma lista de multipart uploads em andamento. Um multipart upload em andamento é um upload que você iniciou, mas que ainda

não concluiu ou anulou. Cada solicitação retorna no máximo 1.000 multipart uploads. Se houver mais de 1.000 multipart uploads em andamento, você precisará enviar solicitações adicionais para recuperar os multipart uploads restantes. Use a listagem retornada apenas para verificação. Você não deve usar o resultado dessa listagem ao enviar uma solicitação de conclusão de multipart upload. Em vez disso, mantenha sua própria lista de números de parte que você especificou ao fazer upload das partes e valores correspondentes de ETag que o Amazon S3 retorna.

### Operações simultâneas de multipart upload

Em um ambiente de desenvolvimento distribuído, é possível que seu aplicativo inicie várias atualizações no mesmo objeto ao mesmo tempo. Seu aplicativo pode iniciar vários multipart uploads usando a mesma chave de objeto. Para cada um desses uploads, seu aplicativo pode fazer upload das partes e enviar uma solicitação de conclusão de upload ao Amazon S3 para criar o objeto. Quando os buckets têm o versionamento habilitado, concluir um multipart upload sempre cria uma nova versão. Para os buckets que não têm o versionamento habilitado, é possível que alguma outra solicitação recebida entre o momento em que um multipart upload é iniciado e quando ele é concluído tenha precedência.

#### Note

É possível que alguma outra solicitação recebida entre o momento em que você iniciou um multipart upload e o concluiu tenha precedência. Por exemplo, se outra operação excluir uma chave depois que você iniciar um multipart upload com essa chave, mas antes de o concluir, a resposta de conclusão do multipart upload poderá indicar a criação bem-sucedida de um objeto sem você nunca ter visto o objeto.

### Multipart upload e definição de preço

Após iniciar um multipart upload, o Amazon S3 retém todas as partes até você concluir ou anular o upload. Durante todo o ciclo de vida, você será cobrado por armazenamento, largura de banda e solicitações desse multipart upload e das partes associadas. Se você anular o multipart upload, o Amazon S3 excluirá os artefatos de upload e as partes carregadas, e você não mais será cobrado por eles. Para obter mais informações sobre a definição de preço, consulte [Definição de preço do Amazon S3](#).

### Anular multipart uploads incompletos usando uma política de ciclo de vida de bucket

Depois de iniciar um multipart upload, você inicia o upload das partes. O Amazon S3 armazena essas partes, mas cria o objeto a partir delas apenas depois do upload de todas elas e envia uma solicitação `successful` para concluir o multipart upload (é necessário verificar se a sua solicitação para concluir o multipart upload teve êxito). Ao receber a solicitação para concluir o multipart upload, o Amazon S3 monta as partes e cria um objeto.

Se você não enviar a solicitação de conclusão do multipart upload, o Amazon S3 não montará as partes e não criará nenhum objeto. Portanto, as partes permanecem no Amazon S3 e você paga pelas partes que são armazenadas no Amazon S3. Como melhor prática, recomendamos que você configure uma regra de ciclo de vida (usando a ação `AbortIncompleteMultipartUpload`) para minimizar os custos de armazenamento.

O Amazon S3 oferece suporte a uma regra de ciclo de vida de bucket que você pode usar para levar o Amazon S3 a anular multipart uploads que não são concluídos dentro de um número especificado de dias após sua inicialização. Quando um multipart upload não é concluído no prazo, ele se torna qualificado para uma operação de anulação e o Amazon S3 interrompe o multipart upload (e exclui as partes associadas ao multipart upload).

Veja a seguir um exemplo de configuração de ciclo de vida que especifica uma regra com a ação `AbortIncompleteMultipartUpload`.

```
<LifecycleConfiguration>
  <Rule>
```

```
<ID>sample-rule</ID>
<Prefix></Prefix>
<Status>Enabled</Status>
<AbortIncompleteMultipartUpload>
    <DaysAfterInitiation>7</DaysAfterInitiation>
</AbortIncompleteMultipartUpload>
</Rule>
</LifecycleConfiguration>
```

No exemplo, a regra não especifica um valor para o elemento `Prefix` (prefixo de nome de chave do objeto) e, portanto, aplica-se a todos os objetos no bucket para os quais você iniciou multipart uploads. Todos os multipart uploads iniciados e não concluídos no prazo de sete dias tornam-se qualificados para uma operação de anulação (a ação não afeta multipart uploads concluídos).

Para obter mais informações sobre a configuração do ciclo de vida de bucket, consulte [Gerenciamento do ciclo de vida de objetos \(p. 122\)](#).

Note

Se o multipart upload for concluído no número de dias especificado na regra, a ação de ciclo de vida `AbortIncompleteMultipartUpload` não se aplicará (ou seja, o Amazon S3 não tomará nenhuma ação). Além disso, essa ação não se aplica a objetos; nenhum objeto é excluído por essa ação de ciclo de vida.

O seguinte comando da CLI `put-bucket-lifecycle` adiciona a configuração de ciclo de vida do bucket especificado.

```
$ aws s3api put-bucket-lifecycle \
  --bucket bucketname \
  --lifecycle-configuration filename-containing-lifecycle-configuration
```

Para testar o comando da CLI, faça o seguinte:

- Configure a CLI da AWS. Para obter instruções, consulte [Configurar a CLI da AWS \(p. 645\)](#).
- Salve a seguinte configuração de ciclo de vida de exemplo em um arquivo (`lifecycle.json`). A configuração de exemplo especifica o prefixo vazio e, portanto, aplica-se a todos os objetos no bucket. Você pode especificar um prefixo para restringir a política a um subconjunto de objetos.

```
{
    "Rules": [
        {
            "ID": "Test Rule",
            "Status": "Enabled",
            "Prefix": "",
            "AbortIncompleteMultipartUpload": {
                "DaysAfterInitiation": 7
            }
        }
    ]
}
```

- Execute o comando da CLI a seguir para definir a configuração do ciclo de vida no seu bucket.

```
aws s3api put-bucket-lifecycle \
  --bucket bucketname \
  --lifecycle-configuration file://lifecycle.json
```

- Para verificar, recupere a configuração de ciclo de vida usando o comando da CLI `get-bucket-lifecycle`.

```
aws s3api get-bucket-lifecycle \
--bucket bucketname
```

5. Para excluir a configuração de ciclo de vida, use o comando da CLI `delete-bucket-lifecycle`.

```
aws s3api delete-bucket-lifecycle \
--bucket bucketname
```

## Limites do Multipart upload do Amazon S3

A tabela a seguir fornece especificações básicas do multipart upload. Para obter mais informações, consulte [Visão geral do multipart upload \(p. 181\)](#).

Item	Especificação
Tamanho máximo do objeto	5 TB
Número máximo de partes por upload	10.000
Números de parte	1 a 10.000 (inclusive)
Tamanho da parte	De 5 MB a 5 GB; a última parte pode ser de < 5 MB
Número máximo de partes retornadas em uma solicitação de listagem de partes	1000
Número máximo de multipart uploads retornados em uma solicitação de listagem de multipart uploads	1000

## Suporte de API para multipart upload

Você pode usar um SDK da AWS para fazer upload de um objeto em partes. As bibliotecas de SDK da AWS a seguir oferecem suporte a multipart upload:

- [AWS SDK for Java](#)
- [AWS SDK para .NET](#)
- [AWS SDK para PHP](#)

Essas bibliotecas fornecem uma abstração de alto nível que facilita o multipart upload de objetos. Contudo, se o seu aplicativo exigir, você pode usar a API REST diretamente. As seções a seguir no Amazon Simple Storage Service API Reference descrevem a API REST para multipart upload.

- [Iniciar multipart upload](#)
- [Upload de parte](#)
- [Upload de parte \(Copiar\)](#)
- [Concluir multipart upload](#)
- [Anular multipart upload](#)
- [Listar partes](#)
- [Listar multipart uploads](#)

## API de multipart upload e permissões

Um indivíduo deve ter as permissões necessárias para usar as operações do multipart upload. Você pode usar ACLs, a política de bucket ou a política de usuário para conceder permissões às pessoas para realizar essas operações. A tabela a seguir lista as permissões necessárias para várias operações de multipart upload ao usar ACLs, a política de bucket ou a política de usuário.

Ação	Permissões obrigatórias
Iniciar multipart upload	Você deve ter permissão para realizar a ação <code>s3:PutObject</code> em um objeto para iniciar o multipart upload.  O proprietário do bucket pode permitir que outros principais realizem a ação <code>s3:PutObject</code> .
Iniciador	O elemento de contêiner que identifica quem iniciou o multipart upload. Se o iniciador for uma conta da AWS, esse elemento fornecerá as mesmas informações que o elemento do proprietário. Se o iniciador for um usuário do IAM, esse elemento fornecerá o ARN e o nome da exibição do usuário.
Upload de parte	Você deve ter permissão para realizar a ação <code>s3:PutObject</code> em um objeto para fazer upload de uma parte.  Somente o iniciador de um multipart upload pode fazer upload de partes. O proprietário do bucket deve permitir que o iniciador realize a ação <code>s3:PutObject</code> em um objeto para que o iniciador possa fazer upload de uma parte desse objeto.
Upload de parte (Copiar)	Você deve ter permissão para realizar a ação <code>s3:PutObject</code> em um objeto para fazer upload de uma parte. Como você está fazendo upload de uma parte a partir de um objeto existente, deverá ter permissão <code>s3:GetObject</code> no objeto de origem.  Somente o iniciador de um multipart upload pode fazer upload de partes. Para iniciador fazer upload de uma parte para um objeto, o proprietário do bucket deve permitir que o iniciador realize a ação <code>s3:PutObject</code> no objeto.
Concluir multipart upload	Você deve ter permissão para realizar a ação <code>s3:PutObject</code> em um objeto para concluir o multipart upload.  Somente o iniciador de um multipart upload pode concluir esse multipart upload. O proprietário do bucket deve permitir que o iniciador realize a ação <code>s3:PutObject</code> em um objeto para que o iniciador possa concluir um multipart upload desse objeto.
Anular multipart upload	Você deve ter permissão para realizar a ação <code>s3:AbortMultipartUpload</code> em um objeto para anular um multipart upload.  Por padrão, o proprietário do bucket e o iniciador do multipart upload têm permissão para executar essa ação. Se o iniciador for um usuário do IAM, a conta da AWS desse usuário também terá permissão para anular esse multipart upload.  Além desses padrões, o proprietário do bucket pode permitir que outros principais executem a ação <code>s3:AbortMultipartUpload</code> em um objeto. O proprietário do bucket pode negar que qualquer principal realize a ação <code>s3:AbortMultipartUpload</code> .
Listar partes	Você deve ter permissão para realizar a ação <code>s3&gt;ListMultipartUploadParts</code> para listar partes em um multipart upload.  Por padrão, o proprietário do bucket tem permissão para listar as partes de qualquer multipart upload para o bucket. O iniciador do multipart upload tem permissão para

Ação	Permissões obrigatórias
	<p>listar partes do multipart upload específico. Se o iniciador do multipart upload for um usuário do IAM, a conta da AWS que controla o usuário do IAM também terá permissão para listar partes desse upload.</p> <p>Além desses padrões, o proprietário do bucket pode permitir que outros principais executem a ação <code>s3&gt;ListMultipartUploadParts</code> em um objeto. O proprietário do bucket também pode negar que qualquer principal realize a ação <code>s3&gt;ListMultipartUploadParts</code>.</p>
Listar multipart uploads	<p>Você deve ter permissão para realizar a ação <code>s3&gt;ListBucketMultipartUploads</code> em um bucket para listar multipart uploads em andamento no bucket.</p> <p>Além desse padrão, o proprietário do bucket pode permitir que outros principais executem a ação <code>s3&gt;ListBucketMultipartUploads</code> no bucket.</p>

Para obter informações sobre a relação entre permissões de ACL e permissões em políticas de acesso, consulte [Mapeamento das permissões da ACL e das permissões da política de acesso \(p. 393\)](#). Para obter informações sobre usuários do IAM, acesse [Trabalho com usuários e grupos](#).

## Usar o AWS Java SDK para multipart upload (API de alto nível)

### Tópicos

- [Fazer upload de um arquivo \(p. 188\)](#)
- [Anular multipart uploads \(p. 189\)](#)
- [Acompanhar o progresso do multipart upload \(p. 190\)](#)

O AWS SDK for Java expõe uma API de alto nível, chamada `TransferManager`, que simplifica multipart uploads (consulte [Upload de objetos usando a API de multipart upload \(p. 181\)](#)). Você pode fazer upload de dados de um arquivo ou de um fluxo. Você também pode definir opções avançadas, como o tamanho da parte que você deseja usar para o multipart upload, ou o número de threads simultâneas que você quer usar quando fizer o upload das partes. Você também pode definir propriedades de objeto opcionais, a classe de armazenamento ou a ACL. Você usa as classes `PutObjectRequest` e `TransferManagerConfiguration` para definir essas opções avançadas.

Quando possível, a classe `TransferManager` tenta usar vários threads para fazer upload de várias partes de um upload único de uma vez só. Ao lidar com tamanhos grandes de conteúdo e com alta banda larga, isso pode representar um aumento significativo na transferência.

Além da funcionalidade de upload de arquivos, a classe `TransferManager` possibilita a você anular um multipart upload em andamento. Um upload é considerado como em andamento depois que você o inicia até ser concluído ou anulado. O `TransferManager` anula todos os multipart uploads em andamento em um bucket especificado que foi iniciado antes de uma data e hora especificadas.

Para obter mais informações sobre multipart upload, incluindo a funcionalidade adicional oferecida por métodos API de nível baixo, consulte [Upload de objetos usando a API de multipart upload \(p. 181\)](#).

### Fazer upload de um arquivo

#### Example

O exemplo a seguir mostra como fazer upload de um objeto usando a API Java de alto nível de multipart upload (a classe `TransferManager`). Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

import java.io.File;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.transfer.TransferManager;
import com.amazonaws.services.s3.transfer.TransferManagerBuilder;
import com.amazonaws.services.s3.transfer.Upload;

public class HighLevelMultipartUpload {

    public static void main(String[] args) throws Exception {
        String clientRegion = "*** Client region ***";
        String bucketName = "*** Bucket name ***";
        String keyName = "*** Object key ***";
        String filePath = "*** Path for file to upload ***";

        try {
            TransferManager transferManager = TransferManagerBuilder.create()
                .withRegion(clientRegion)
                .withS3Client(AmazonS3ClientBuilder.create()
                    .withRegion(clientRegion)
                    .build())
                .withProfileCredentialsProvider(ProfileCredentialsProvider.create())
                .build();

            Upload upload = transferManager.upload(bucketName, keyName, new File(filePath));
            System.out.println("Upload initiated with ID: " + upload.getId());
        } catch (AmazonServiceException | SdkClientException e) {
            System.out.println("Error occurred while performing multipart upload: " + e.getMessage());
        }
    }
}
```

```
AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
    .withRegion(clientRegion)
    .withCredentials(new ProfileCredentialsProvider())
    .build();
TransferManager tm = TransferManagerBuilder.standard()
    .withS3Client(s3Client)
    .build();

// TransferManager processes all transfers asynchronously,
// so this call returns immediately.
Upload upload = tm.upload(bucketName, keyName, new File(filePath));
System.out.println("Object upload started");

// Optionally, wait for the upload to finish before continuing.
upload.waitForCompletion();
System.out.println("Object upload complete");
}

catch(AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
}
catch(SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
}
```

## Anular multipart uploads

### Example

O exemplo a seguir usa a API Java de alto nível (a classe `TransferManager`) para anular todos os multipart uploads em andamento que foram iniciados em um bucket específico há uma semana. Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

import java.util.Date;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.transfer.TransferManager;
import com.amazonaws.services.s3.transfer.TransferManagerBuilder;

public class HighLevelAbortMultipartUpload {

    public static void main(String[] args) {
        String clientRegion = "**** Client region ****";
        String bucketName = "**** Bucket name ****";

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

```
TransferManager tm = TransferManagerBuilder.standard()
    .withS3Client(s3Client)
    .build();

    // sevenDays is the duration of seven days in milliseconds.
    long sevenDays = 1000 * 60 * 60 * 24 * 7;
    Date oneWeekAgo = new Date(System.currentTimeMillis() - sevenDays);
    tm.abortMultipartUploads(bucketName, oneWeekAgo);
}
catch(AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
}
catch(SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client couldn't
    // parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

### Acompanhar o progresso do multipart upload

A API Java de alto nível de multipart upload fornece uma interface de escuta, `ProgressListener`, para acompanhar o progresso ao fazer upload de um objeto no Amazon S3. Os eventos de progresso notificam periodicamente o ouvinte sobre a transferência dos bytes.

O exemplo a seguir demonstra como assinar um evento `ProgressEvent` e gravar um handler:

#### Example

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-
developer-guide/blob/master/LICENSE-SAMPLECODE.)

import java.io.File;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.event.ProgressEvent;
import com.amazonaws.event.ProgressListener;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.PutObjectRequest;
import com.amazonaws.services.s3.transfer.TransferManager;
import com.amazonaws.services.s3.transfer.TransferManagerBuilder;
import com.amazonaws.services.s3.transfer.Upload;

public class HighLevelTrackMultipartUpload {

    public static void main(String[] args) throws Exception {
        String clientRegion = "**** Client region ****";
        String bucketName = "**** Bucket name ****";
        String keyName = "**** Object key ****";
        String filePath = "**** Path to file to upload ***";

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

```
        .withS3Client(s3Client)
        .build();
    PutObjectRequest request = new PutObjectRequest(bucketName, keyName, new
File(filePath));

    // To receive notifications when bytes are transferred, add a
    // ProgressListener to your request.
    request.setGeneralProgressListener(new ProgressListener() {
        public void progressChanged(ProgressEvent progressEvent) {
            System.out.println("Transferred bytes: " +
progressEvent.getBytesTransferred());
        }
    });
    // TransferManager processes all transfers asynchronously,
    // so this call returns immediately.
    Upload upload = tm.upload(request);

    // Optionally, you can wait for the upload to finish before continuing.
    upload.waitForCompletion();
}
catch(AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
}
catch(SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

## Usar o AWS Java SDK para um multipart upload (API de baixo nível)

### Tópicos

- [Fazer upload de um arquivo \(p. 192\)](#)
- [Listar multipart uploads \(p. 194\)](#)
- [Anular um multipart upload \(p. 195\)](#)

O AWS SDK for Java expõe uma API de baixo nível muito semelhante à API REST do Amazon S3 para os multipart uploads (consulte [Upload de objetos usando a API de multipart upload \(p. 181\)](#)). Use a API de baixo nível quando precisar pausar e retomar multipart uploads, variar os tamanhos das partes durante o upload ou não souber o tamanho dos dados com antecedência. Quando não houver esses requisitos, use a API de nível alto (consulte [Usar o AWS Java SDK para multipart upload \(API de alto nível\) \(p. 188\)](#)).

### Fazer upload de um arquivo

O exemplo a seguir mostra como usar as classes Java de baixo nível para fazer upload de um arquivo. Ele realiza as seguintes etapas:

- Inicia um multipart upload usando o método `AmazonS3Client.initiateMultipartUpload()`, e transmite a um objeto `InitiateMultipartUploadRequest`.
- Salva o ID de upload retornado pelo método `AmazonS3Client.initiateMultipartUpload()`. Você fornece esse ID de upload para cada operação de multipart upload subsequente.
- Faz upload das partes do objeto. Para cada parte, chame o método `AmazonS3Client.uploadPart()`. Você fornece informações sobre o upload da parte usando um objeto `UploadPartRequest`.
- Para cada parte, você salva a ETag da resposta do método `AmazonS3Client.uploadPart()` em uma lista. Você usa os valores de ETag para concluir o multipart upload.
- Chama o método `AmazonS3Client.completeMultipartUpload()` para concluir o multipart upload.

### Example

Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

import java.io.File;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.CompleteMultipartUploadRequest;
import com.amazonaws.services.s3.model.InitiateMultipartUploadRequest;
import com.amazonaws.services.s3.model.InitiateMultipartUploadResult;
import com.amazonaws.services.s3.model.PartETag;
import com.amazonaws.services.s3.model.UploadPartRequest;
import com.amazonaws.services.s3.model.UploadPartResult;

public class LowLevelMultipartUpload {
```

```
public static void main(String[] args) throws IOException {
    String clientRegion = "**** Client region ****";
    String bucketName = "**** Bucket name ****";
    String keyName = "**** Key name ****";
    String filePath = "**** Path to file to upload ****";

    File file = new File(filePath);
    long contentLength = file.length();
    long partSize = 5 * 1024 * 1024; // Set part size to 5 MB.

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

        // Create a list of ETag objects. You retrieve ETags for each object part
uploaded,
        // then, after each individual part has been uploaded, pass the list of ETags
to
        // the request to complete the upload.
        List<PartETag> partETags = new ArrayList<PartETag>();

        // Initiate the multipart upload.
        InitiateMultipartUploadRequest initRequest = new
InitiateMultipartUploadRequest(bucketName, keyName);
        InitiateMultipartUploadResult initResponse =
s3Client.initiateMultipartUpload(initRequest);

        // Upload the file parts.
        long filePosition = 0;
        for (int i = 1; filePosition < contentLength; i++) {
            // Because the last part could be less than 5 MB, adjust the part size as
needed.
            partSize = Math.min(partSize, (contentLength - filePosition));

            // Create the request to upload a part.
            UploadPartRequest uploadRequest = new UploadPartRequest()
                .withBucketName(bucketName)
                .withKey(keyName)
                .withUploadId(initResponse.getUploadId())
                .withPartNumber(i)
                .withFileOffset(filePosition)
                .withFile(file)
                .withPartSize(partSize);

            // Upload the part and add the response's ETag to our list.
            UploadPartResult uploadResult = s3Client.uploadPart(uploadRequest);
            partETags.add(uploadResult.getPartETag());

            filePosition += partSize;
        }

        // Complete the multipart upload.
        CompleteMultipartUploadRequest compRequest = new
CompleteMultipartUploadRequest(bucketName, keyName,
                initResponse.getUploadId(), partETags);
        s3Client.completeMultipartUpload(compRequest);
    }
    catch(AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    }
    catch(SdkClientException e) {
```

```
// Amazon S3 couldn't be contacted for a response, or the client
// couldn't parse the response from Amazon S3.
e.printStackTrace();
}
}
```

## Listar multipart uploads

### Example

O exemplo a seguir mostra como recuperar uma lista de multipart uploads em andamento usando a API Java de baixo nível:

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-
// developer-guide/blob/master/LICENSE-SAMPLECODE.)

import java.util.List;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ListMultipartUploadsRequest;
import com.amazonaws.services.s3.model.MultipartUpload;
import com.amazonaws.services.s3.model.MultipartUploadListing;

public class ListMultipartUploads {

    public static void main(String[] args) {
        String clientRegion = "**** Client region ****";
        String bucketName = "**** Bucket name ****";

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

            // Retrieve a list of all in-progress multipart uploads.
            ListMultipartUploadsRequest allMultipartUploadsRequest = new
ListMultipartUploadsRequest(bucketName);
            MultipartUploadListing multipartUploadListing =
s3Client.listMultipartUploads(allMultipartUploadsRequest);
            List<MultipartUpload> uploads = multipartUploadListing.getMultipartUploads();

            // Display information about all in-progress multipart uploads.
            System.out.println(uploads.size() + " multipart upload(s) in progress.");
            for (MultipartUpload u : uploads) {
                System.out.println("Upload in progress: Key = " + u.getKey() + "\", id =
" + u.getUploadId());
            }
        }
        catch(AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        }
        catch(SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

```
        }
    }
```

### Anular um multipart upload

Você pode abortar um multipart upload em andamento chamando o método `AmazonS3Client.abortMultipartUpload()`. Esse método exclui todas as partes que foram carregadas no Amazon S3 e libera os recursos. Você fornece o ID do upload, o nome do bucket e o nome da chave.

#### Example

O exemplo a seguir mostra como anular multipart uploads usando a API Java de baixo nível.

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-
developer-guide/blob/master/LICENSE-SAMPLECODE.)

import java.util.List;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.AbortMultipartUploadRequest;
import com.amazonaws.services.s3.model.ListMultipartUploadsRequest;
import com.amazonaws.services.s3.model.MultipartUpload;
import com.amazonaws.services.s3.model.MultipartUploadListing;

public class LowLevelAbortMultipartUpload {

    public static void main(String[] args) {
        String clientRegion = "**** Client region ****";
        String bucketName = "**** Bucket name ****";

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

            // Find all in-progress multipart uploads.
            ListMultipartUploadsRequest allMultipartUploadsRequest = new
ListMultipartUploadsRequest(bucketName);
            MultipartUploadListing multipartUploadListing =
s3Client.listMultipartUploads(allMultipartUploadsRequest);

            List<MultipartUpload> uploads = multipartUploadListing.getMultipartUploads();
            System.out.println("Before deletions, " + uploads.size() + " multipart uploads
in progress.");

            // Abort each upload.
            for (MultipartUpload u : uploads) {
                System.out.println("Upload in progress: Key = \'" + u.getKey() + "\', id =
" + u.getUploadId());
                s3Client.abortMultipartUpload(new AbortMultipartUploadRequest(bucketName,
u.getKey(), u.getUploadId()));
                System.out.println("Upload deleted: Key = \'" + u.getKey() + "\', id = " +
u.getUploadId());
            }

            // Verify that all in-progress multipart uploads have been aborted.
        }
    }
}
```

```
        multipartUploadListing =
s3Client.listMultipartUploads(allMultipartUploadsRequest);
        uploads = multipartUploadListing.getMultipartUploads();
        System.out.println("After aborting uploads, " + uploads.size() + " multipart
uploads in progress.");
    }
    catch(AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    }
    catch(SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}
```

#### Note

Em vez de anular multipart uploads individualmente, você pode anular todos os multipart uploads em andamento que foram iniciados antes de uma determinada hora. Essa operação de limpeza é útil para anular os multipart uploads que você iniciou, mas que não foram concluídos nem anulados. Para obter mais informações, consulte [Anular multipart uploads \(p. 189\)](#).

## Usar o AWS SDK para .NET para multipart upload (API de alto nível)

### Tópicos

- [Faça upload de um arquivo para um Bucket do S3 usando o AWS SDK para .NET \(API de alto nível\). \(p. 197\)](#)
- [Fazer upload de um diretório \(p. 199\)](#)
- [Anular multipart uploads para um bucket do S3 usando o AWS SDK para .NET \(API de alto nível\) \(p. 200\)](#)
- [Acompanhe o andamento de um Multipart Upload para um Bucket do S3 usando o AWS SDK para .NET \(API de alto nível\) \(p. 202\)](#)

O AWS SDK para .NET expõe uma API de alto nível que simplifica os multipart uploads (consulte [Upload de objetos usando a API de multipart upload \(p. 181\)](#)). Você pode carregar dados de um arquivo, de um diretório ou de um fluxo. Para obter mais informações sobre multipart uploads do Amazon S3, consulte [Visão geral do multipart upload \(p. 181\)](#).

A classe `TransferUtility` fornece métodos para fazer upload de arquivos e diretórios, monitorando o progresso do upload, e anulando multipart uploads.

[Faça upload de um arquivo para um Bucket do S3 usando o AWS SDK para .NET \(API de alto nível\).](#)

Para fazer upload de um arquivo para um bucket do S3, use a classe `TransferUtility`. Ao fazer o upload de dados de um arquivo, você deve fornecer o nome da chave do objeto. Caso contrário, a API usará o nome do arquivo no lugar do nome da chave. Ao fazer o upload de dados de um fluxo, você deve fornecer o nome da chave do objeto.

Para definir opções de upload avançadas — tais como o tamanho da parte, o número de threads ao fazer upload das partes simultaneamente, os metadados, a classe de armazenamento, ou ACL — use a classe `TransferUtilityUploadRequest`.

O exemplo do C# a seguir faz upload de um arquivo em um bucket do Amazon S3 em várias partes. Ele mostra como usar várias sobrecargas de `TransferUtility.Upload` para fazer upload de um arquivo. Cada chamada sucessiva para upload substitui o upload anterior. Para obter informações sobre a compatibilidade do exemplo com uma versão específica do AWS SDK para .NET e instruções para criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

using Amazon.S3;
using Amazon.S3.Transfer;
using System;
using System.IO;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class UploadFileMPUHighLevelAPITest
    {
        private const string bucketName = "*** provide bucket name ***";
        private const string keyName = "*** provide a name for the uploaded object ***";
        private const string filePath = "*** provide the full path name of the file to
upload ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
```

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

private static async Task UploadFileAsync()
{
    try
    {
        var fileTransferUtility =
            new TransferUtility(s3Client);

        // Option 1. Upload a file. The file name is used as the object key name.
        await fileTransferUtility.UploadAsync(filePath, bucketName);
        Console.WriteLine("Upload 1 completed");

        // Option 2. Specify object key name explicitly.
        await fileTransferUtility.UploadAsync(filePath, bucketName, keyName);
        Console.WriteLine("Upload 2 completed");

        // Option 3. Upload data from a type of System.IO.Stream.
        using (var fileToUpload =
            new FileStream(filePath, FileMode.Open, FileAccess.Read))
        {
            await fileTransferUtility.UploadAsync(fileToUpload,
                bucketName, keyName);
        }
        Console.WriteLine("Upload 3 completed");

        // Option 4. Specify advanced settings.
        var fileTransferUtilityRequest = new TransferUtilityUploadRequest
        {
            BucketName = bucketName,
            FilePath = filePath,
            StorageClass = S3StorageClass.StandardInfrequentAccess,
            PartSize = 6291456, // 6 MB.
            Key = keyName,
            CannedACL = S3CannedACL.PublicRead
        };
        fileTransferUtilityRequest.Metadata.Add("param1", "Value1");
        fileTransferUtilityRequest.Metadata.Add("param2", "Value2");

        await fileTransferUtility.UploadAsync(fileTransferUtilityRequest);
        Console.WriteLine("Upload 4 completed");
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine("Error encountered on server. Message:{0}' when writing
an object", e.Message);
    }
    catch (Exception e)
    {
        Console.WriteLine("Unknown encountered on server. Message:{0}' when
writing an object", e.Message);
    }
}
}
```

[Mais informações](#)

[AWS SDK para .NET](#)

## Fazer upload de um diretório

Você pode usar a classe `TransferUtility` para fazer upload de um diretório inteiro. Por padrão, a API faz upload somente dos arquivos na raiz do diretório especificado. No entanto, você pode especificar um upload recursivo em todos os subdiretórios.

Para selecionar arquivos no diretório especificado, com base nos critérios de filtragem, especifique expressões de filtragem. Por exemplo, para carregar somente os arquivos .pdf de um diretório, você especifica a expressão de filtragem de "`*.pdf`".

Ao fazer upload de arquivos de um diretório, você não especifica os nomes de chaves para os objetos resultantes. O Amazon S3 cria os nomes de chave usando o caminho do arquivo original. Por exemplo, suponha que você tem um diretório denominado `c:\myfolder` com a seguinte estrutura:

### Example

```
C:\myfolder
    \a.txt
    \b.pdf
    \media\
        An.mp3
```

Ao fazer upload desse diretório, o Amazon S3 usa os seguintes nomes de chaves:

### Example

```
a.txt
b.pdf
media/An.mp3
```

### Example

O exemplo de C# a seguir faz upload de um diretório em um bucket do Amazon S3. Ele mostra como usar várias sobrecargas de `TransferUtility.UploadDirectory` para fazer upload do diretório. Cada chamada sucessiva para upload substitui o upload anterior. Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

using Amazon.S3;
using Amazon.S3.Transfer;
using System;
using System.IO;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class UploadDirMPUHighLevelAPITest
    {
        private const string existingBucketName = "**** bucket name ****";
        private const string directoryPath = @"**** directory path ****";
        // The example uploads only .txt files.
        private const string wildCard = "*.txt";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
        static void Main()
        {
```

```
s3Client = new AmazonS3Client(bucketRegion);
UploadDirAsync().Wait();
}

private static async Task UploadDirAsync()
{
    try
    {
        var directoryTransferUtility =
            new TransferUtility(s3Client);

        // 1. Upload a directory.
        await directoryTransferUtility.UploadDirectoryAsync(directoryPath,
            existingBucketName);
        Console.WriteLine("Upload statement 1 completed");

        // 2. Upload only the .txt files from a directory
        //     and search recursively.
        await directoryTransferUtility.UploadDirectoryAsync(
            directoryPath,
            existingBucketName,
            wildCard,
            SearchOption.AllDirectories);
        Console.WriteLine("Upload statement 2 completed");

        // 3. The same as Step 2 and some optional configuration.
        //     Search recursively for .txt files to upload.
        var request = new TransferUtilityUploadDirectoryRequest
        {
            BucketName = existingBucketName,
            Directory = directoryPath,
            SearchOption = SearchOption.AllDirectories,
            SearchPattern = wildCard
        };

        await directoryTransferUtility.UploadDirectoryAsync(request);
        Console.WriteLine("Upload statement 3 completed");
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine(
            "Error encountered ***. Message:'{0}' when writing an object",
            e.Message);
    }
    catch (Exception e)
    {
        Console.WriteLine(
            "Unknown encountered on server. Message:'{0}' when writing an object",
            e.Message);
    }
}
}
```

#### Anular multipart uploads para um bucket do S3 usando o AWS SDK para .NET (API de alto nível)

Para anular multipart uploads em andamento, use a classe `TransferUtility` de AWS SDK para .NET. Forneça um valor `DateTime`. A API, em seguida, anulará todos os multipart uploads que foram iniciados antes da data e hora especificadas e removerá as partes carregadas. Um upload é considerado como em andamento depois que você o inicia e até ser concluído ou anulado.

Como você será cobrado por todo armazenamento associado às partes carregadas, é importante que você conclua o multipart upload para terminar de criar o objeto criado ou anular o multipart upload para remover as partes carregadas. Para obter mais informações sobre multipart uploads do Amazon S3, consulte [Visão](#)

geral do multipart upload (p. 181). Para obter informações sobre definição de preço, consulte [Multipart upload e definição de preço \(p. 183\)](#).

O exemplo do C# a seguir anula todos os multipart uploads em andamento que foram iniciados em um bucket específico há uma semana. Para obter informações sobre a compatibilidade do exemplo com uma versão específica do AWS SDK para .NET e instruções para criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-
developer-guide/blob/master/LICENSE-SAMPLECODE.)

using Amazon.S3;
using Amazon.S3.Transfer;
using System;
using System.Threading.Tasks;

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

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

        private static async Task AbortMPUAsync()
        {
            try
            {
                var transferUtility = new TransferUtility(s3Client);

                // Abort all in-progress uploads initiated before the specified date.
                await transferUtility.AbortMultipartUploadsAsync(
                    bucketName, DateTime.Now.AddDays(-7));
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:{0} when writing
an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:{0} when
writing an object", e.Message);
            }
        }
    }
}
```

#### Note

Você também pode anular um multipart upload específico. Para obter mais informações, consulte [Listar multipart uploads para um Bucket do S3 usando o AWS SDK para .NET \(baixo nível\) \(p. 206\)](#).

## Mais informações

### AWS SDK para .NET

#### Acompanhe o andamento de um Multipart Upload para um Bucket do S3 usando o AWS SDK para .NET (API de alto nível)

O seguinte exemplo do C# faz upload de um arquivo em um bucket do S3 usando a classe `TransferUtility` e monitora o andamento do upload. Para obter informações sobre a compatibilidade do exemplo com uma versão específica do AWS SDK para .NET e instruções para criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-
developer-guide/blob/master/LICENSE-SAMPLECODE.)

using Amazon.S3;
using Amazon.S3.Transfer;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class TrackMPUUsingHighLevelAPITest
    {
        private const string bucketName = "*** provide the bucket name ***";
        private const string keyName = "*** provide the name for the uploaded object ***";
        private const string filePath = " *** provide the full path name of the file to
upload ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

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

        private static async Task TrackMPUAsync()
        {
            try
            {
                var fileTransferUtility = new TransferUtility(s3Client);

                // Use TransferUtilityUploadRequest to configure options.
                // In this example we subscribe to an event.
                var uploadRequest =
                    new TransferUtilityUploadRequest
                {
                    BucketName = bucketName,
                    FilePath = filePath,
                    Key = keyName
                };

                uploadRequest.UploadProgressEvent +=
                    new EventHandler<UploadProgressArgs>
                    (uploadRequest_UncalibratedUploadPartProgressEvent);

                await fileTransferUtility.UploadAsync(uploadRequest);
                Console.WriteLine("Upload completed");
            }
            catch (AmazonS3Exception e)
```

```
        {
            Console.WriteLine("Error encountered on server. Message:'{0}' when writing
an object", e.Message);
        }
        catch (Exception e)
        {
            Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
        }
    }

    static void uploadRequest_UploadPartProgressEvent(object sender, UploadProgressArgs
e)
{
    // Process event.
    Console.WriteLine("{0}/{1}", e.TransferredBytes, e.TotalBytes);
}
}
```

[Mais informações](#)

[AWS SDK para .NET](#)

## Usar o AWS SDK para .NET para multipart upload (API de nível baixo)

O AWS SDK para .NET expõe uma API de baixo nível muito semelhante à API REST do Amazon S3 para multipart upload (consulte [Usar a API REST para multipart upload \(p. 214\)](#)). Use a API de nível baixo quando precisar pausar e retomar multipart uploads, variar os tamanhos das partes durante o upload ou quando você não souber o tamanho necessário dos dados com antecedência. Use a API de nível alto (consulte [Usar o AWS SDK para .NET para multipart upload \(API de alto nível\) \(p. 197\)](#)) sempre que não houver esses requisitos.

### Tópicos

- [Faça upload de um arquivo para um Bucket do S3 usando o AWS SDK para .NET \(API de nível baixo\). \(p. 204\)](#)
- [Listar multipart uploads para um Bucket do S3 usando o AWS SDK para .NET \(baixo nível\) \(p. 206\)](#)
- [Acompanhe o andamento de um Multipart Upload para um Bucket do S3 usando o AWS SDK para .NET \(de baixo nível\) \(p. 207\)](#)
- [Anular multipart uploads para um Bucket do S3 usando o AWS SDK para .NET \(baixo nível\) \(p. 207\)](#)

[Faça upload de um arquivo para um Bucket do S3 usando o AWS SDK para .NET \(API de nível baixo\).](#)

O exemplo do C# a seguir mostra como usar a API de multipart upload do AWS SDK para .NET de nível baixo para fazer upload de um arquivo para um bucket do S3. Para obter informações sobre multipart uploads do Amazon S3, consulte [Visão geral do multipart upload \(p. 181\)](#).

### Note

Ao usar a API do AWS SDK para .NET para fazer upload de objetos grandes, pode ocorrer um tempo limite mesmo quando os dados são gravados para o fluxo de solicitação. Você pode definir um tempo limite explícito usando o `UploadPartRequest`.

O exemplo do C# a seguir faz upload de um arquivo para um bucket do S3 usando a API de multipart upload de nível baixo. Para obter informações sobre a compatibilidade do exemplo com uma versão específica do AWS SDK para .NET e instruções para criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

using Amazon.Runtime;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.IO;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class UploadFileMPULowLevelAPITest
    {
        private const string bucketName = "*** provide bucket name ***";
        private const string keyName = "*** provide a name for the uploaded object ***";
        private const string filePath = "*** provide the full path name of the file to
upload ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
```

```
public static void Main()
{
    s3Client = new AmazonS3Client(bucketRegion);
    Console.WriteLine("Uploading an object");
    UploadObjectAsync().Wait();
}

private static async Task UploadObjectAsync()
{
    // Create list to store upload part responses.
    List<UploadPartResponse> uploadResponses = new List<UploadPartResponse>();

    // Setup information required to initiate the multipart upload.
    InitiateMultipartUploadRequest initiateRequest = new
    InitiateMultipartUploadRequest
    {
        BucketName = bucketName,
        Key = keyName
    };

    // Initiate the upload.
    InitiateMultipartUploadResponse initResponse =
        await s3Client.InitiateMultipartUploadAsync(initiateRequest);

    // Upload parts.
    long contentLength = new FileInfo(filePath).Length;
    long partSize = 5 * (long)Math.Pow(2, 20); // 5 MB

    try
    {
        Console.WriteLine("Uploading parts");

        long filePosition = 0;
        for (int i = 1; filePosition < contentLength; i++)
        {
            UploadPartRequest uploadRequest = new UploadPartRequest
            {
                BucketName = bucketName,
                Key = keyName,
                UploadId = initResponse.UploadId,
                PartNumber = i,
                PartSize = partSize,
                FilePosition = filePosition,
                FilePath = filePath
            };

            // Track upload progress.
            uploadRequest.StreamTransferProgress +=
                new
                EventHandler<StreamTransferProgressArgs>(UploadPartProgressCallback);

            // Upload a part and add the response to our list.
            uploadResponses.Add(await s3Client.UploadPartAsync(uploadRequest));

            filePosition += partSize;
        }

        // Setup to complete the upload.
        CompleteMultipartUploadRequest completeRequest = new
        CompleteMultipartUploadRequest
        {
            BucketName = bucketName,
            Key = keyName,
            UploadId = initResponse.UploadId
        };
        completeRequest.AddPartETags(uploadResponses);
    }
}
```

```
// Complete the upload.  
CompleteMultipartUploadResponse completeUploadResponse =  
    await s3Client.CompleteMultipartUploadAsync(completeRequest);  
}  
catch (Exception exception)  
{  
    Console.WriteLine("An AmazonS3Exception was thrown: { 0 }",  
exception.Message);  
  
    // Abort the upload.  
    AbortMultipartUploadRequest abortMPURequest = new  
AbortMultipartUploadRequest  
{  
    BucketName = bucketName,  
    Key = keyName,  
    UploadId = initResponse.UploadId  
};  
    await s3Client.AbortMultipartUploadAsync(abortMPURequest);  
}  
}  
public static void UploadPartProgressEventCallback(object sender,  
StreamTransferProgressArgs e)  
{  
    // Process event.  
    Console.WriteLine("{0}/{1}", e.TransferredBytes, e.TotalBytes);  
}  
}  
}
```

## Mais informações

### AWS SDK para .NET

#### [Listar multipart uploads para um Bucket do S3 usando o AWS SDK para .NET \(baixo nível\)](#)

Para listar todos os multipart uploads em andamento em um bucket específico, use a classe `ListMultipartUploadsRequest` da API de multipart upload do AWS SDK para .NET de nível baixo. O `AmazonS3Client.ListMultipartUploads` método retorna uma instância da classe `ListMultipartUploadsResponse` que fornece informações sobre multipart uploads em andamento.

Multipart upload em andamento é um multipart upload que foi iniciado com o uso da solicitação para iniciar o multipart upload, mas que ainda não foi concluído ou anulado. Para obter mais informações sobre multipart uploads do Amazon S3, consulte [Visão geral do multipart upload \(p. 181\)](#).

O exemplo do C# a seguir mostra como usar o AWS SDK para .NET para listar todos os multipart uploads em andamento em um bucket. Para obter informações sobre a compatibilidade do exemplo com uma versão específica do AWS SDK para .NET e instruções sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

```
ListMultipartUploadsRequest request = new ListMultipartUploadsRequest  
{  
    BucketName = bucketName // Bucket receiving the uploads.  
};  
  
ListMultipartUploadsResponse response = await  
    AmazonS3Client.ListMultipartUploadsAsync(request);
```

## Mais informações

### AWS SDK para .NET

## Acompanhe o andamento de um Multipart Upload para um Bucket do S3 usando o AWS SDK para .NET (de baixo nível)

Para acompanhar o andamento de um multipart upload, use o evento `UploadPartRequest.StreamTransferProgress` fornecido pela API de multipart upload do AWS SDK para .NET de nível baixo. O evento ocorre periodicamente. Ele retorna informações como o número total de bytes a transferir e o número de bytes transferidos.

O exemplo do C# a seguir mostra como acompanhar o andamento de multipart uploads. Para um exemplo do C# completo que inclui o código seguinte, consulte [Faça upload de um arquivo para um Bucket do S3 usando o AWS SDK para .NET \(API de nível baixo\).](#) (p. 204).

```
UploadPartRequest uploadRequest = new UploadPartRequest
{
    // Provide the request data.
};

uploadRequest.StreamTransferProgress +=
    new EventHandler<StreamTransferProgressArgs>(UploadPartProgressEventCallback);

...
public static void UploadPartProgressEventCallback(object sender,
    StreamTransferProgressArgs e)
{
    // Process the event.
    Console.WriteLine("{0}/{1}", e.TransferredBytes, e.TotalBytes);
}
```

### Mais informações

#### [AWS SDK para .NET](#)

## [Anular multipart uploads para um Bucket do S3 usando o AWS SDK para .NET \(baixo nível\)](#)

Você pode abortar um multipart upload em andamento chamando o método `AmazonS3Client.AbortMultipartUploadAsync`. Além de anular o upload, esse método exclui todas as partes que foram carregadas no Amazon S3.

Para anular um multipart upload, é preciso fornecer o ID de upload e nomes do bucket e da chave usados no upload. Depois de anular um multipart upload, não é possível usar o ID de upload para fazer upload de partes adicionais. Para obter mais informações sobre multipart uploads do Amazon S3, consulte [Visão geral do multipart upload](#) (p. 181).

O exemplo do C# a seguir mostra como anular um multipart upload. Para um exemplo do C# completo que inclui o código seguinte, consulte [Faça upload de um arquivo para um Bucket do S3 usando o AWS SDK para .NET \(API de nível baixo\).](#) (p. 204).

```
AbortMultipartUploadRequest abortMPURequest = new AbortMultipartUploadRequest
{
    BucketName = existingBucketName,
    Key = keyName,
    UploadId = initResponse.UploadId
};
await AmazonS3Client.AbortMultipartUploadAsync(abortMPURequest);
```

Você também pode anular todos os multipart uploads em andamento que foram iniciados antes de um horário específico. Essa operação de limpeza é útil para anular os multipart uploads que não foram concluídos nem anulados. Para obter mais informações, consulte [Anular multipart uploads para um bucket do S3 usando o AWS SDK para .NET \(API de alto nível\)](#) (p. 200).

[Mais informações](#)

[AWS SDK para .NET](#)

## Usar o SDK PHP da AWS para multipart upload

Você pode fazer upload de arquivos grandes para o Amazon S3 em várias partes. Você deve usar o multipart upload para arquivos maiores que 5 GB. O AWS SDK para PHP expõe a classe [MultipartUploader](#) que simplifica multipart uploads.

O método `upload` da classe `MultipartUploader` funciona melhor para um multipart upload simples. Se precisar pausar e retomar multipart uploads, variar os tamanhos das partes durante o upload ou não souber o tamanho necessário dos dados com antecedência, use a API de nível baixo PHP. Para obter mais informações, consulte [Usar o SDK PHP da AWS para multipart upload \(API de baixo nível\) \(p. 211\)](#).

Para obter mais informações sobre multipart uploads, consulte [Upload de objetos usando a API de multipart upload \(p. 181\)](#). Para obter informações sobre como fazer upload de arquivos menores que 5 GB em tamanho, consulte [Faça upload de objetos usando o AWS SDK para PHP \(p. 178\)](#).

### Fazer upload de um arquivo usando multipart upload de alto nível

Este tópico explica como usar a classe `Aws\S3\Model\MultipartUpload\UploadBuilder` de alto nível do AWS SDK para PHP para multipart uploads de arquivos. Pressupõe-se que você já esteja seguindo as instruções para [Usar o AWS SDK para PHP e executar exemplos do PHP \(p. 649\)](#) e tenha o AWS SDK para PHP devidamente instalado.

O exemplo de PHP a seguir faz upload de um arquivo em um bucket do Amazon S3. O exemplo demonstra como definir parâmetros para o objeto `MultipartUploader`.

Para obter informações sobre a execução dos exemplos de PHP neste guia, consulte [Executar exemplos do PHP \(p. 649\)](#).

```
<?php

require 'vendor/autoload.php';

use Aws\Common\Exception\MultipartUploadException;
use Aws\S3\MultipartUploader;
use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// Prepare the upload parameters.
uploader = new MultipartUploader($s3, '/path/to/large/file.zip', [
    'bucket' => $bucket,
    'key'     => $keyname
]);

// Perform the upload.
try {
    $result = $uploader->upload();
    echo "Upload complete: {$result['ObjectURL']}" . PHP_EOL;
} catch (MultipartUploadException $e) {
    echo $e->getMessage() . PHP_EOL;
}
```

### Recursos relacionados

- [AWS SDK para PHP para classe Aws\S3\S3Client do Amazon S3](#)

- [Multipart uploads do Amazon S3](#)
- [Documentação do AWS SDK para PHP](#)

## Usar o SDK PHP da AWS para multipart upload (API de baixo nível)

### Tópicos

- [Carregar um arquivo em várias partes usando a API de baixo nível do SDK PHP \(p. 211\)](#)
- [Listar multipart uploads usando a API de baixo nível do AWS SDK para PHP \(p. 212\)](#)
- [Anular um multipart upload \(p. 213\)](#)

O AWS SDK para PHP expõe uma API de baixo nível muito semelhante à API REST do Amazon S3 para multipart upload (consulte [Usar a API REST para multipart upload \(p. 214\)](#)). Use a API de nível baixo quando precisar pausar e retomar multipart uploads, variar os tamanhos das partes durante o upload ou se você não souber o tamanho necessário dos dados com antecedência. Use as abstrações de alto nível do AWS SDK para PHP (consulte [Usar o SDK PHP da AWS para multipart upload \(p. 209\)](#)) sempre que não possuir esses requisitos.

### Carregar um arquivo em várias partes usando a API de baixo nível do SDK PHP

Este tópico mostra como usar o método de baixo nível `uploadPart` da versão 3 do AWS SDK para PHP para carregar um arquivo em várias partes. Pressupõe-se que você já esteja seguindo as instruções para [Usar o AWS SDK para PHP e executar exemplos do PHP \(p. 649\)](#) e tenha o AWS SDK para PHP devidamente instalado.

O exemplo de PHP a seguir carrega um arquivo para um bucket do Amazon S3 usando o multipart upload da API de baixo nível PHP. Para obter informações sobre a execução dos exemplos de PHP neste guia, consulte [Executar exemplos do PHP \(p. 649\)](#).

```
<?php

require 'vendor/autoload.php';

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';
$filename = '*** Path to and Name of the File to Upload ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

$result = $s3->createMultipartUpload([
    'Bucket'      => $bucket,
    'Key'         => $keyname,
    'StorageClass' => 'REDUCED_REDUNDANCY',
    'ACL'          => 'public-read',
    'Metadata'    => [
        'param1' => 'value 1',
        'param2' => 'value 2',
        'param3' => 'value 3'
    ]
]);
$uploadId = $result['UploadId'];

// Upload the file in parts.
try {
    $file = fopen($filename, 'r');
    $partNumber = 1;
    while (!feof($file)) {
        $result = $s3->uploadPart([
            'Bucket'      => $bucket,
```

```
'Key'          => $keyname,
'UploadId'     => $uploadId,
'PartNumber'   => $partNumber,
'Body'         => fread($file, 5 * 1024 * 1024),
]);
$parts['Parts'][$partNumber] = [
'PartNumber'  => $partNumber,
'ETag'        => $result['ETag'],
];
$partNumber++;

echo "Uploading part {$partNumber} of {$filename}." . PHP_EOL;
}
fclose($file);
} catch (S3Exception $e) {
$result = $s3->abortMultipartUpload([
'Bucket'      => $bucket,
'Key'         => $keyname,
'UploadId'    => $uploadId
]);

echo "Upload of {$filename} failed." . PHP_EOL;
}

// Complete the multipart upload.
$result = $s3->completeMultipartUpload([
'Bucket'      => $bucket,
'Key'         => $keyname,
'UploadId'    => $uploadId,
'MultipartUpload' => $parts,
]);
$url = $result['Location'];

echo "Uploaded {$filename} to {$url}." . PHP_EOL;
```

## Recursos relacionados

- [AWS SDK para PHP para classe Aws\S3\S3Client do Amazon S3](#)
- [Multipart uploads do Amazon S3](#)
- [Documentação do AWS SDK para PHP](#)

## Listar multipart uploads usando a API de baixo nível do AWS SDK para PHP

Este tópico mostra como usar as classes da API de baixo nível da versão 3 do AWS SDK para PHP para listar todos os multipart uploads em andamento em um bucket. Pressupõe-se que você já esteja seguindo as instruções para [Usar o AWS SDK para PHP e executar exemplos do PHP \(p. 649\)](#) e tenha o AWS SDK para PHP devidamente instalado.

O exemplo de PHP a seguir demonstra a listagem de todos os multipart uploads em andamento em um bucket.

```
<?php

require 'vendor/autoload.php';

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';

$s3 = new S3Client([
'version' => 'latest',
```

```
    'region' => 'us-east-1'
]);

// Retrieve a list of the current multipart uploads.
$result = $s3->listMultipartUploads([
    'Bucket' => $bucket
]);

// Write the list of uploads to the page.
print_r($result->toArray());
```

## Recursos relacionados

- [AWS SDK para PHP para classe Aws\S3\S3Client do Amazon S3](#)
- [Multipart uploads do Amazon S3](#)
- [Documentação do AWS SDK para PHP](#)

## Anular um multipart upload

Este tópico descreve como usar uma classe da versão 3 do AWS SDK para PHP para anular um multipart upload que está em andamento. Pressupõe-se que você já esteja seguindo as instruções para [Usar o AWS SDK para PHP e executar exemplos do PHP \(p. 649\)](#) e tenha o AWS SDK para PHP devidamente instalado.

O exemplo de PHP a seguir mostra como anular um multipart upload em andamento usando o método `abortMultipartUpload()`. Para obter informações sobre a execução dos exemplos de PHP neste guia, consulte [Executar exemplos do PHP \(p. 649\)](#).

```
<?php

require 'vendor/autoload.php';

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';
$uploadId = '*** Upload ID of upload to Abort ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region' => 'us-east-1'
]);

// Abort the multipart upload.
$s3->abortMultipartUpload([
    'Bucket' => $bucket,
    'Key' => $keyname,
    'UploadId' => $uploadId,
]);
```

## Recursos relacionados

- [AWS SDK para PHP para classe Aws\S3\S3Client do Amazon S3](#)
- [Multipart uploads do Amazon S3](#)
- [Documentação do AWS SDK para PHP](#)

## Usar o AWS SDK para Ruby para multipart upload

O AWS SDK para Ruby versão 3 oferece suporte a multipart uploads do Amazon S3 de duas formas. Para a primeira opção, você pode usar um assistente de upload de arquivo gerenciado. Este é o método recomendado para fazer upload de arquivos em um bucket e fornece os seguintes benefícios:

- Gerencia multipart uploads para objetos com mais de 15 MB.
- Abre corretamente arquivos em modo binário para evitar problemas de codificação.
- Usa vários threads para upload de partes de grandes objetos em paralelo.

Para obter mais informações, consulte [Upload de arquivos no Amazon S3](#) no blog do desenvolvedor da AWS.

Como alternativa, você pode usar as seguintes operações de cliente do multipart upload diretamente:

- [create\\_multipart\\_upload](#) – Inicia um multipart upload e retorna um ID de upload.
- [upload\\_part](#) – Faz upload de uma parte em um multipart upload.
- [upload\\_part\\_copy](#) – Faz upload de uma parte copiando dados de um objeto existente como a fonte de dados.
- [complete\\_multipart\\_upload](#) – Conclui um multipart upload montando as partes anteriormente carregadas.
- [abort\\_multipart\\_upload](#) – Anula um multipart upload.

Para obter mais informações, consulte [Usar o AWS SDK para Ruby - versão 3 \(p. 650\)](#).

## Usar a API REST para multipart upload

As seções a seguir no Amazon Simple Storage Service API Reference descrevem a API REST para multipart upload.

- [Iniciar multipart upload](#)
- [Upload de parte](#)
- [Concluir multipart upload](#)
- [Anular multipart upload](#)
- [Listar partes](#)
- [Listar multipart uploads](#)

Use essas APIs para fazer suas solicitações REST ou use um dos SDKs que fornecemos. Para obter mais informações sobre o recurso SDKs, consulte [Suporte de API para multipart upload \(p. 185\)](#).

## Fazer upload de objetos usando pre-signed URLs

### Tópicos

- [Upload de um objeto usando um pre-signed URL \(AWS SDK for Java\) \(p. 215\)](#)
- [Fazer upload de um objeto para um bucket do S3 usando um pre-signed URL \(AWS SDK para .NET\) \(p. 217\)](#)
- [Upload de um objeto usando um pre-signed URL \(AWS SDK para Ruby\) \(p. 218\)](#)

Um pre-signed URL fornece acesso ao objeto identificado no URL, desde que o criador do pre-signed URL tenha permissões para acessar esse objeto. Isto é, se você receber um pre-signed URL para fazer upload de um objeto, poderá fazer upload do objeto somente se o criador do pre-signed URL tiver as permissões necessárias para fazer upload desse objeto.

Por padrão, todos os objetos e buckets são privados. Os pre-signed URLs serão úteis se você desejar que o usuário/cliente seja capaz de fazer upload de um objeto específico no seu bucket, mas não exigir que ele tenha credenciais ou permissões de segurança da AWS. Quando cria um pre-signed URL, você deve fornecer as credenciais de segurança, e então especificar um nome de bucket, uma chave de objeto, um método HTTP (PUT para upload de objetos) e data e hora de expiração. Os pre-signed URLs são válidos apenas pela duração especificada.

Você pode gerar um pre-signed URL programaticamente usando o AWS SDK for Java ou AWS SDK para .NET. Se você estiver usando o Microsoft Visual Studio, também poderá usar o AWS Explorer para gerar um pre-signed URL de objeto sem gravar nenhum código. Qualquer pessoa que receber um pre-signed URL válido poderá fazer upload de um objeto programaticamente.

Para obter mais informações, acesse [Usar o Amazon S3 no AWS Explorer](#).

Para obter instruções sobre como instalar o AWS Explorer, consulte [Usar os AWS SDKs, a CLI e os Explorers \(p. 639\)](#).

#### Note

Qualquer um com credenciais de segurança válidas pode criar um pre-signed URL. No entanto, para fazer upload de um objeto, o pre-signed URL deve ter sido criado por alguém que tenha permissão para executar a operação na qual o pre-signed URL está baseado.

## Upload de um objeto usando um pre-signed URL (AWS SDK for Java)

Você pode usar o AWS SDK for Java para gerar um presigned URL que você, ou qualquer pessoa a quem você der o URL, poderá usar para fazer upload de um objeto no Amazon S3. Ao usar o URL para fazer o upload de um objeto, o Amazon S3 cria o objeto no bucket especificado. Se um objeto com a mesma chave especificada no pre-signed URL já existir no bucket, o Amazon S3 substituirá o objeto existente pelo objeto carregado. Para concluir um upload com êxito, você deve fazer o seguinte:

- Especificar o verbo HTTP PUT ao criar os objetos `GeneratePresignedUrlRequest` e `HttpURLConnection`.
- Interagir com o objeto `HttpURLConnection` de alguma forma após concluir o upload. O exemplo a seguir faz isso usando o objeto `HttpURLConnection` para verificar o código de resposta HTTP.

#### Example

Esse exemplo gera um pre-signed URL e o usa para fazer upload dos dados de amostra como um objeto. Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

import java.io.IOException;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.HttpMethod;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.GeneratePresignedUrlRequest;
import com.amazonaws.services.s3.model.S3Object;

public class GeneratePresignedUrlAndUploadObject {
```

```
public static void main(String[] args) throws IOException {
    String clientRegion = "**** Client region ****";
    String bucketName = "**** Bucket name ****";
    String objectKey = "**** Object key ****";

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

        // Set the pre-signed URL to expire after one hour.
        java.util.Date expiration = new java.util.Date();
        long expTimeMillis = expiration.getTime();
        expTimeMillis += 1000 * 60 * 60;
        expiration.setTime(expTimeMillis);

        // Generate the pre-signed URL.
        System.out.println("Generating pre-signed URL.");
        GeneratePresignedUrlRequest generatePresignedUrlRequest = new
GeneratePresignedUrlRequest(bucketName, objectKey)
            .withMethod(HttpMethod.PUT)
            .withExpiration(expiration);
        URL url = s3Client.generatePresignedUrl(generatePresignedUrlRequest);

        // Create the connection and use it to upload the new object using the pre-
        signed URL.
        HttpURLConnection connection = (HttpURLConnection) url.openConnection();
        connection.setDoOutput(true);
        connection.setRequestMethod("PUT");
        OutputStreamWriter out = new OutputStreamWriter(connection.getOutputStream());
        out.write("This text uploaded as an object via presigned URL.");
        out.close();

        // Check the HTTP response code. To complete the upload and make the object
        available,
        // you must interact with the connection object in some way.
        connection.getResponseCode();
        System.out.println("HTTP response code: " + connection.getResponseCode());

        // Check to make sure that the object was uploaded successfully.
        S3Object object = s3Client.getObject(bucketName, objectKey);
        System.out.println("Object " + object.getKey() + " created in bucket " +
object.getBucketName());
    }
    catch(AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    }
    catch(SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}
```

## Fazer upload de um objeto para um bucket do S3 usando um pre-signed URL (AWS SDK para .NET)

O exemplo do C# a seguir mostra como usar o AWS SDK para .NET para fazer upload de um objeto para um bucket do S3 usando um pre-signed URL. Para obter mais informações sobre pre-signed URLs, consulte [Fazer upload de objetos usando pre-signed URLs \(p. 214\)](#).

Esse exemplo gera um pre-signed URL para um objeto específico e o utiliza para fazer upload de um arquivo. Para obter informações sobre a compatibilidade do exemplo com uma versão específica do AWS SDK para .NET e instruções sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.IO;
using System.Net;

namespace Amazon.DocSamples.S3
{
    class UploadObjectUsingPresignedURLTest
    {
        private const string bucketName = "**** provide bucket name ****";
        private const string objectKey = "**** provide the name for the uploaded object ****";
        private const string filePath = "**** provide the full path name of the file to upload ****";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            var url = GeneratePreSignedURL();
            UploadObject(url);
        }

        private static void UploadObject(string url)
        {
            HttpWebRequest httpRequest = WebRequest.Create(url) as HttpWebRequest;
            httpRequest.Method = "PUT";
            using (Stream dataStream = httpRequest.GetRequestStream())
            {
                var buffer = new byte[8000];
                using (FileStream fileStream = new FileStream(filePath, FileMode.Open,
                    FileAccess.Read))
                {
                    int bytesRead = 0;
                    while ((bytesRead = fileStream.Read(buffer, 0, buffer.Length)) > 0)
                    {
                        dataStream.Write(buffer, 0, bytesRead);
                    }
                }
            }
            HttpWebResponse response = httpRequest.GetResponse() as HttpWebResponse;
        }

        private static string GeneratePreSignedURL()
        {
```

```
var request = new GetPreSignedUrlRequest
{
    BucketName = bucketName,
    Key        = objectKey,
    Verb       = HttpVerb.PUT,
    Expires    = DateTime.Now.AddMinutes(5)
};

string url = s3Client.GetPreSignedURL(request);
return url;
}
}
```

[Mais informações](#)

[AWS SDK para .NET](#)

## Upload de um objeto usando um pre-signed URL (AWS SDK para Ruby)

As tarefas a seguir orientam você na utilização de um script Ruby para fazer upload de um objeto usando um pre-signed URL para SDK para Ruby - Versão 3.

### Upload de objetos - SDK para Ruby - Versão 3

1	Crie uma instância da classe <code>Aws::S3::Resource</code> .
2	Forneça um nome de bucket e uma chave de objeto chamando os métodos <code>#bucket[]</code> e <code>#object[]</code> da instância da classe <code>Aws::S3::Resource</code> .  Gere um pre-signed URL criando uma instância da classe <code>URI</code> e use-a para analisar o método <code>.presigned_url</code> da instância da classe <code>Aws::S3::Resource</code> . Você deve especificar <code>:put</code> como um argumento para <code>.presigned_url</code> e especificar <code>PUT</code> como <code>Net::HTTP::Session#send_request</code> se quiser fazer upload de um objeto.
3	Qualquer pessoa com o pre-signed URL pode fazer upload de um objeto.  O upload cria um objeto ou substitui qualquer objeto existente pela mesma chave especificada no pre-signed URL.

O exemplo de código Ruby a seguir demonstra as tarefas precedentes para SDK para Ruby - Versão 3.

### Example

```
#Uploading an object using a presigned URL for SDK para Ruby - Version 3.

require 'aws-sdk-s3'
require 'net/http'

s3 = Aws::S3::Resource.new(region:'us-west-2')

obj = s3.bucket('BucketName').object('KeyName')
# Replace BucketName with the name of your bucket.
# Replace KeyName with the name of the object you are creating or replacing.

url = URI.parse(obj.presigned_url(:put))

body = "Hello World!"
# This is the contents of your object. In this case, it's a simple string.

Net::HTTP.start(url.host) do |http|
```

```
http.send_request("PUT", url.request_uri, body, {
# This is required, or Net::HTTP will add a default unsigned content-type.
    "content-type" => "",
})
end

puts obj.get.body.read
# This will print out the contents of your object to the terminal window.
```

## Cópia de objetos

### Tópicos

- [Recursos relacionados \(p. 220\)](#)
- [Cópia de objetos em uma única operação \(p. 220\)](#)
- [Copiar objetos usando a API de multipart upload \(p. 226\)](#)

A operação de cópia cria uma cópia de um objeto que já está armazenado no Amazon S3. Você pode criar uma cópia do seu objeto de até 5 GB em uma única operação atômica. Contudo, para copiar objetos maiores do que 5 GB, você deve usar a API de multipart upload. Usando a operação copy você pode:

- Criar cópias adicionais de objetos
- Renomear objetos copiando-os e excluindo os originais
- Mover objetos por locais do Amazon S3 (por ex. us-west-1 and EU)
- Alterar metadados do objeto

Cada objeto do Amazon S3 tem metadados. É um conjunto de pares nome-valor. Você pode definir metadados de objeto no momento em que fizer seu upload. Após fazer upload do objeto, você não pode modificar seus metadados. A única forma de modificar metadados de objeto é fazer uma cópia do objeto e definir os metadados. Na operação de cópia, você define o mesmo objeto como origem e destino.

Cada objeto tem metadados. Alguns deles são metadados de sistema e outros são definidos pelo usuário. Usuários controlam alguns dos metadados de sistema, como a configuração de classe de armazenamento para usar o objeto e configurar a criptografia do lado do servidor. Quando você copia um objeto, os metadados de sistema controlados pelo usuário e os metadados definidos pelo usuário também são copiados. O Amazon S3 redefine os metadados controlados pelo sistema. Por exemplo, quando você copia um objeto, o Amazon S3 redefine a data de criação do objeto copiado. Você não precisa definir nenhum desses valores na sua solicitação de cópia.

Ao copiar um objeto, você pode decidir atualizar alguns dos valores de metadados. Por exemplo, se o objeto de origem é configurado para usar o armazenamento padrão, você pode escolher usar o armazenamento com redundância reduzida para a cópia de objeto. Você também pode modificar os valores de metadados definidos pelo usuário presentes no objeto de origem. Se optar por atualizar metadados configuráveis do usuário do objeto (definidos pelo sistema ou pelo usuário) durante a cópia, você deverá especificar explicitamente todos os metadados configuráveis pelo usuário presentes no objeto de origem na solicitação, mesmo se estiver apenas alterando um dos valores de metadados.

Para obter mais informações sobre metadados de objeto, consulte [Chave de objeto e metadados \(p. 102\)](#).

### Note

Cópia de objetos por locais incorre em alterações de banda larga.

### Note

Se o objeto de origem estiver arquivado no Amazon Glacier (a classe de armazenamento de objeto for GLACIER), você deverá, primeiro, restaurar uma cópia temporária antes de copiar o

objeto para outro bucket. Para obter informações sobre objetos de arquivo, consulte [Transição para a classe de armazenamento GLACIER \(arquivamento de objeto\) \(p. 127\)](#).

Ao copiar objetos, você pode solicitar que o Amazon S3 salve o objeto de destino criptografado usando uma chave de criptografia AWS Key Management Service (KMS), uma chave de criptografia gerenciada pelo Amazon S3 ou uma chave de criptografia fornecida pelo cliente. Da mesma forma, você deve especificar informações de criptografia na solicitação. Se a origem da cópia for um objeto armazenado no Amazon S3 usando criptografia do lado do servidor com chave fornecida pelo cliente, você precisará fornecer informações de criptografia na solicitação, de maneira que o Amazon S3 possa decodificar o objeto para a cópia. Para obter mais informações, consulte [Proteção de dados usando criptografia \(p. 409\)](#).

## Recursos relacionados

- [Usar os AWS SDKs, a CLI e os Explorers \(p. 639\)](#)

## Cópia de objetos em uma única operação

O exemplo nesta seção mostra como copiar objetos de até 5 GB em uma única operação. Para copiar objetos maiores do que 5 GB, você deve usar a API de multipart upload. Para obter mais informações, consulte [Copiar objetos usando a API de multipart upload \(p. 226\)](#).

### Tópicos

- [Copie um objeto usando o AWS SDK for Java \(p. 220\)](#)
- [Copiar um objeto do Amazon S3 em uma única operação usando o AWS SDK para .NET \(p. 221\)](#)
- [Copie um objeto usando o AWS SDK para PHP \(p. 222\)](#)
- [Copie um objeto usando o AWS SDK para Ruby \(p. 223\)](#)
- [Copie um objeto usando a API REST \(p. 224\)](#)

## Copie um objeto usando o AWS SDK for Java

### Example

O exemplo a seguir mostra como copiar um objeto no Amazon S3 usando o AWS SDK for Java. Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

import java.io.IOException;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.CopyObjectRequest;

public class CopyObjectSingleOperation {

    public static void main(String[] args) throws IOException {
        String clientRegion = "*** Client region ***";
        String bucketName = "*** Bucket name ***";
        String sourceKey = "*** Source object key *** ";
    }
}
```

```
String destinationKey = "*** Destination object key ***";

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

    // Copy the object into a new object in the same bucket.
    CopyObjectRequest copyObjRequest = new CopyObjectRequest(bucketName, sourceKey,
bucketName, destinationKey);
    s3Client.copyObject(copyObjRequest);
}
catch(AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
}
catch(SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

## Copiar um objeto do Amazon S3 em uma única operação usando o AWS SDK para .NET

O exemplo do C# a seguir mostra como usar a AWS SDK para .NET de alto nível para copiar objetos de até 5 GB em uma única operação. Para objetos maiores do que 5 GB, use o exemplo de cópia do multipart upload descrito em [Copiar um objeto do Amazon S3 usando a API de multipart upload do AWS SDK para .NET \(p. 228\)](#).

Esse exemplo faz a cópia de um objeto de até 5 GB. Para obter informações sobre a compatibilidade do exemplo com uma versão específica do AWS SDK para .NET e instruções sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

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

namespace Amazon.DocSamples.S3
{
    class CopyObjectTest
    {
        private const string sourceBucket = "*** provide the name of the bucket with source
object ***";
        private const string destinationBucket = "*** provide the name of the bucket to
copy the object to ***";
        private const string objectKey = "*** provide the name of object to copy ***";
        private const string destObjectKey = "*** provide the destination object key name
***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main()
    }
}
```

```
{  
    s3Client = new AmazonS3Client(bucketRegion);  
    Console.WriteLine("Copying an object");  
    CopyingObjectAsync().Wait();  
}  
  
private static async Task CopyingObjectAsync()  
{  
    try  
    {  
        CopyObjectRequest request = new CopyObjectRequest  
        {  
            SourceBucket = sourceBucket,  
            SourceKey = objectKey,  
            DestinationBucket = destinationBucket,  
            DestinationKey = destObjectKey  
        };  
        CopyObjectResponse response = await s3Client.CopyObjectAsync(request);  
    }  
    catch (AmazonS3Exception e)  
    {  
        Console.WriteLine("Error encountered on server. Message:'{0}' when writing  
an object", e.Message);  
    }  
    catch (Exception e)  
    {  
        Console.WriteLine("Unknown encountered on server. Message:'{0}' when  
writing an object", e.Message);  
    }  
}
```

## Mais informações

[AWS SDK para .NET](#)

## Copie um objeto usando o AWS SDK para PHP

Este tópico orienta sobre o uso de classes da versão 3 do AWS SDK para PHP para copiar um único objeto e múltiplos objetos com o Amazon S3 de um bucket para outro ou no mesmo bucket.

Este tópico pressupõe que você já está seguindo as instruções para [Usar o AWS SDK para PHP e executar exemplos do PHP \(p. 649\)](#) e tem o AWS SDK para PHP devidamente instalado.

As tarefas a seguir orientam sobre a utilização de classes PHP SDK para gerar um objeto que já está armazenado no Amazon S3.

As tarefas a seguir orientam sobre o uso de classes PHP para fazer múltiplas cópias de um objeto no Amazon S3.

### Cópia de objetos

1	Crie uma instância de um cliente do Amazon S3 usando o construtor da classe <code>Aws\S3\S3Client</code> .
2	Para fazer múltiplas cópias de um objeto, você executa um lote de chamadas para o método <code>getCommand()</code> do cliente do Amazon S3 do que é herdado da classe <code>Aws\CommandInterface</code> . Você fornece o comando <code>CopyObject</code> como o primeiro argumento e uma matriz contendo o bucket de origem, o nome de uma chave de origem, o bucket de destino e o nome do destino como segundo argumento.

### Example de cópia de objetos no Amazon S3

O exemplo PHP a seguir ilustra o uso do método `copyObject()` para copiar um único objeto no Amazon S3 usando um lote de chamadas para `CopyObject`, usando o método `getCommand()` para fazer várias cópias de um objeto.

```
<?php

require 'vendor/autoload.php';

use Aws\S3\S3Client;

$sourceBucket = '*** Your Source Bucket Name ***';
$sourceKeyname = '*** Your Source Object Key ***';
$targetBucket = '*** Your Target Bucket Name ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// Copy an object.
$s3->copyObject([
    'Bucket'      => $targetBucket,
    'Key'         => "{$sourceKeyname}-copy",
    'CopySource'  => "{$sourceBucket}/{$sourceKeyname}",
]);

// Perform a batch of CopyObject operations.
$batch = array();
for ($i = 1; $i <= 3; $i++) {
    $batch[] = $s3->getCommand('CopyObject', [
        'Bucket'      => $targetBucket,
        'Key'         => "{$targetKeyname}-{$i}",
        'CopySource'  => "{$sourceBucket}/{$sourceKeyname}",
    ]);
}
try {
    $results = CommandPool::batch($s3, $batch);
    foreach($results as $result) {
        if ($result instanceof ResultInterface) {
            // Result handling here
        }
        if ($result instanceof AwsException) {
            // AwsException handling here
        }
    }
} catch (\Exception $e) {
    // General error handling here
}
```

### Recursos relacionados

- [AWS SDK para PHP para classe Aws\S3\S3Client do Amazon S3](#)
- [Documentação do AWS SDK para PHP](#)

## Copie um objeto usando o AWS SDK para Ruby

As tarefas a seguir orientam sobre o uso de classes Ruby para copiar um objeto no Amazon S3 de um bucket para outro ou para copiar um objeto no mesmo bucket.

## Cópia de objetos

- |   |   |
|---|---|
| 1 | Use o gem modularizado do Amazon S3 para versão 3 do AWS SDK para Ruby, exija "aws-sdk-s3" e forneça suas credenciais da AWS. Para obter mais informações sobre como fornecer suas credenciais, consulte <a href="#">Fazer solicitações usando credenciais de usuário do IAM ou da conta da AWS (p. 18)</a> . |
| 2 | Forneça as informações da solicitação, como o nome do bucket de origem, o nome da chave de origem, o nome do bucket de destino e a chave de destino.  |

O exemplo de código Ruby a seguir demonstra as tarefas precedentes usando o método `#copy_object` para copiar um objeto de um bucket para outro.

### Example

```
require 'aws-sdk-s3'

source_bucket_name = '*** Provide bucket name ***'
target_bucket_name = '*** Provide bucket name ***'
source_key = '*** Provide source key ***'
target_key = '*** Provide target key ***'

s3 = Aws::S3::Client.new(region: 'us-west-2')
s3.copy_object({bucket: target_bucket_name, copy_source: source_bucket_name + '/' +
  source_key, key: target_key})

puts "Copying file #{source_key} to #{target_key}."
```

## Copie um objeto usando a API REST

Este exemplo descreve como copiar um objeto usando REST. Para obter mais informações sobre a API REST, consulte [Objeto \(Cópia\) PUT](#).

Este exemplo copia o objeto `flotsam` do bucket `pacific` para o objeto `jetsam` do bucket `atlantic`, preservando seus metadados.

```
PUT /jetsam HTTP/1.1
Host: atlantic.s3.amazonaws.com
x-amz-copy-source: /pacific/flotsam
Authorization: AWS AKIAIOSFODNN7EXAMPLE:ENoSbxYByFA0UGLZUqJN5EUnLDg=
Date: Wed, 20 Feb 2008 22:12:21 +0000
```

A assinatura foi gerada a partir das informações a seguir.

```
PUT\r\n
\r\n
\r\n
Wed, 20 Feb 2008 22:12:21 +0000\r\n

x-amz-copy-source:/pacific/flotsam\r\n
/atlantic/jetsam
```

O Amazon S3 retorna a seguinte resposta que especifica a ETag do objeto e quando foi modificado pela última vez.

```
HTTP/1.1 200 OK
x-amz-id-2: Vyaxt7qEbzb34BnSu5hctyyNSlHTYZFMWK4FtzO+ix8JQNyalaLdTshL0Kxatba0zt
x-amz-request-id: 6B13C3C5B34AF333
```

```
Date: Wed, 20 Feb 2008 22:13:01 +0000
Content-Type: application/xml
Transfer-Encoding: chunked
Connection: close
Server: AmazonS3
<?xml version="1.0" encoding="UTF-8"?>

<CopyObjectResult>
    <LastModified>2008-02-20T22:13:01</LastModified>
    <ETag>"7e9c608af58950deeb370c98608ed097"</ETag>
</CopyObjectResult>
```

## Copiar objetos usando a API de multipart upload

Os exemplos nesta seção mostram como copiar objetos maiores que 5 GB usando a API de multipart upload. Copie objetos menores que 5 GB em uma única operação. Para obter mais informações, consulte [Cópia de objetos em uma única operação \(p. 220\)](#).

### Tópicos

- [Copiar um objeto usando a API de multipart upload do AWS SDK for Java \(p. 226\)](#)
- [Copiar um objeto do Amazon S3 usando a API de multipart upload do AWS SDK para .NET \(p. 228\)](#)
- [Copiar objetos usando a API de multipart upload REST \(p. 230\)](#)

### Copiar um objeto usando a API de multipart upload do AWS SDK for Java

Para copiar um objeto do Amazon S3 com mais de 5 GB com o AWS SDK for Java, use a API Java de baixo nível. Para objetos menores que 5 GB, use a cópia de operação única descrita em [Copie um objeto usando o AWS SDK for Java \(p. 220\)](#).

Para copiar um objeto usando a API Java de baixo nível, faça o seguinte:

- Inicie um multipart upload executando o método `AmazonS3Client.initiateMultipartUpload()`.
- Salve o ID de upload do objeto de resposta retornado pelo método `AmazonS3Client.initiateMultipartUpload()`. Você fornece esse ID de upload para cada operação do upload de parte.
- Copie todas as partes. Para cada parte que você precisar copiar, crie uma nova instância da classe `CopyPartRequest`. Forneça as informações da parte, incluindo a origem e os nomes do bucket de destino, as chaves de objeto de origem e de destino, o ID de upload, os locais dos primeiros e últimos bytes da parte, e o número da parte.
- Salve as respostas das chamadas de método `AmazonS3Client.copyPart()`. Cada resposta inclui o valor `ETag` e o número da parte para a parte cujo upload foi feito. Você precisa dessas informações para concluir o multipart upload.
- Chame o método `AmazonS3Client.completeMultipartUpload()` para concluir a operação de cópia.

### Example

O exemplo a seguir mostra como usar a API Java de baixo nível do Amazon S3 para realizar uma cópia multipart. Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.  
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)  
  
import java.io.IOException;  
import java.util.ArrayList;  
import java.util.List;  
  
import com.amazonaws.AmazonServiceException;  
import com.amazonaws.SdkClientException;  
import com.amazonaws.auth.profile.ProfileCredentialsProvider;  
import com.amazonaws.services.s3.*;  
import com.amazonaws.services.s3.model.*;  
  
public class LowLevelMultipartCopy {
```

```
public static void main(String[] args) throws IOException {
    String clientRegion = "**** Client region ****";
    String sourceBucketName = "**** Source bucket name ****";
    String sourceObjectKey = "**** Source object key ****";
    String destBucketName = "**** Target bucket name ****";
    String destObjectKey = "**** Target object key ****";

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

        // Initiate the multipart upload.
        InitiateMultipartUploadRequest initRequest = new
InitiateMultipartUploadRequest(destBucketName, destObjectKey);
        InitiateMultipartUploadResult initResult =
s3Client.initiateMultipartUpload(initRequest);

        // Get the object size to track the end of the copy operation.
        GetObjectMetadataRequest metadataRequest = new
GetObjectMetadataRequest(sourceBucketName, sourceObjectKey);
        ObjectMetadata metadataResult = s3Client.getObjectMetadata(metadataRequest);
        long objectSize = metadataResult.getContentLength();

        // Copy the object using 5 MB parts.
        long partSize = 5 * 1024 * 1024;
        long bytePosition = 0;
        int partNum = 1;
        List<CopyPartResult> copyResponses = new ArrayList<CopyPartResult>();
        while (bytePosition < objectSize) {
            // The last part might be smaller than partSize, so check to make sure
            // that lastByte isn't beyond the end of the object.
            long lastByte = Math.min(bytePosition + partSize - 1, objectSize - 1);

            // Copy this part.
            CopyPartRequest copyRequest = new CopyPartRequest()
                .withSourceBucketName(sourceBucketName)
                .withSourceKey(sourceObjectKey)
                .withDestinationBucketName(destBucketName)
                .withDestinationKey(destObjectKey)
                .withUploadId(initResult.getUploadId())
                .withFirstByte(bytePosition)
                .withLastByte(lastByte)
                .withPartNumber(partNum++);
            copyResponses.add(s3Client.copyPart(copyRequest));
            bytePosition += partSize;
        }

        // Complete the upload request to concatenate all uploaded parts and make the
copied object available.
        CompleteMultipartUploadRequest completeRequest = new
CompleteMultipartUploadRequest(
            destBucketName,
            destObjectKey,
            initResult.getUploadId(),
            getETags(copyResponses));
        s3Client.completeMultipartUpload(completeRequest);
        System.out.println("Multipart copy complete.");
    }
    catch(AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    }
}
```

```
        }
        catch(SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }

    // This is a helper function to construct a list of ETags.
    private static List<PartETag> getETags(List<CopyPartResult> responses) {
        List<PartETag> etags = new ArrayList<PartETag>();
        for (CopyPartResult response : responses) {
            etags.add(new PartETag(response.getPartNumber(), response.getETag()));
        }
        return etags;
    }
}
```

## Copiar um objeto do Amazon S3 usando a API de multipart upload do AWS SDK para .NET

O exemplo C# a seguir mostra como usar o AWS SDK para .NET para copiar um objeto do Amazon S3 com mais de 5 GB de um local de origem para outro, como de um bucket para outro. Para copiar objetos menores que 5 GB, use um procedimento de cópia de operação única descrita em [Copiar um objeto do Amazon S3 em uma única operação usando o AWS SDK para .NET \(p. 221\)](#). Para obter mais informações sobre multipart uploads do Amazon S3, consulte [Visão geral do multipart upload \(p. 181\)](#).

Este exemplo mostra como copiar um objeto do Amazon S3 com mais de 5 GB de um bucket do S3 para outro usando a API de multipart upload do AWS SDK para .NET. Para obter informações sobre a compatibilidade com o SDK e instruções para criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class CopyObjectUsingMPUapiTest
    {
        private const string sourceBucket = "*** provide the name of the bucket with source object ***";
        private const string targetBucket = "*** provide the name of the bucket to copy the object to ***";
        private const string sourceObjectKey = "*** provide the name of object to copy ***";
        private const string targetObjectKey = "*** provide the name of the object copy ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            Console.WriteLine("Copying an object");
            MPUCopyObjectAsync().Wait();
        }
}
```

```
}

private static async Task MPUCopyObjectAsync()
{
    // Create a list to store the upload part responses.
    List<UploadPartResponse> uploadResponses = new List<UploadPartResponse>();
    List<CopyPartResponse> copyResponses = new List<CopyPartResponse>();

    // Setup information required to initiate the multipart upload.
    InitiateMultipartUploadRequest initiateRequest =
        new InitiateMultipartUploadRequest
    {
        BucketName = targetBucket,
        Key = targetObjectKey
    };

    // Initiate the upload.
    InitiateMultipartUploadResponse initResponse =
        await s3Client.InitiateMultipartUploadAsync(initiateRequest);

    // Save the upload ID.
    String uploadId = initResponse.UploadId;

    try
    {
        // Get the size of the object.
        GetObjectMetadataRequest metadataRequest = new GetObjectMetadataRequest
        {
            BucketName = sourceBucket,
            Key = sourceObjectKey
        };

        GetObjectMetadataResponse metadataResponse =
            await s3Client.GetObjectMetadataAsync(metadataRequest);
        long objectSize = metadataResponse.ContentLength; // Length in bytes.

        // Copy the parts.
        long partSize = 5 * (long)Math.Pow(2, 20); // Part size is 5 MB.

        long bytePosition = 0;
        for (int i = 1; bytePosition < objectSize; i++)
        {
            CopyPartRequest copyRequest = new CopyPartRequest
            {
                DestinationBucket = targetBucket,
                DestinationKey = targetObjectKey,
                SourceBucket = sourceBucket,
                SourceKey = sourceObjectKey,
                UploadId = uploadId,
                FirstByte = bytePosition,
                LastByte = bytePosition + partSize - 1 >= objectSize ? objectSize -
1 : bytePosition + partSize - 1,
                PartNumber = i
            };

            copyResponses.Add(await s3Client.CopyPartAsync(copyRequest));

            bytePosition += partSize;
        }

        // Set up to complete the copy.
        CompleteMultipartUploadRequest completeRequest =
            new CompleteMultipartUploadRequest
        {
            BucketName = targetBucket,
            Key = targetObjectKey,
            UploadId = initResponse.UploadId
        };
    }
}
```

```
};

completeRequest.AddPartETags(copyResponses);

// Complete the copy.
CompleteMultipartUploadResponse completeUploadResponse =
    await s3Client.CompleteMultipartUploadAsync(completeRequest);
}

catch (AmazonS3Exception e)
{
    Console.WriteLine("Error encountered on server. Message:{0} when writing
an object", e.Message);
}
catch (Exception e)
{
    Console.WriteLine("Unknown encountered on server. Message:{0} when
writing an object", e.Message);
}
}
```

## Mais informações

[AWS SDK para .NET](#)

## Copiar objetos usando a API de multipart upload REST

As seções a seguir no Amazon Simple Storage Service API Reference descrevem a API REST para multipart upload. Para copiar um objeto existente use a API para upload de parte (cópia) e especifique o objeto de origem adicionando o cabeçalho de solicitação `x-amz-copy-source` na solicitação.

- [Iniciar multipart upload](#)
- [Upload de parte](#)
- [Upload de parte \(Copiar\)](#)
- [Concluir multipart upload](#)
- [Anular multipart upload](#)
- [Listar partes](#)
- [Listar multipart uploads](#)

Use essas APIs para fazer suas solicitações REST ou use um dos SDKs que fornecemos. Para obter mais informações sobre o recurso SDKs, consulte [Suporte de API para multipart upload \(p. 185\)](#).

## Lista de chaves de objeto

As chaves podem ser listadas por prefixo. Escolhendo um prefixo comum para os nomes de chaves relacionadas e marcando essas chaves com um caractere especial que limite a hierarquia, você pode usar a operação de lista para selecionar e navegar pelas chaves hierarquicamente. Isso é semelhante à forma como arquivos são armazenados em diretórios em um sistema de arquivos.

O Amazon S3 expõe uma operação de lista que permite que você enumere as chaves contidas em um bucket. As chaves são selecionadas para a listagem pelo bucket e pelo prefixo. Por exemplo, considere um bucket chamado “dicionário” que contém uma chave para cada palavra inglesa. Você pode fazer uma chamada para listar todas as chaves no bucket iniciadas com a letra “q”. Os resultados da lista são obtidos sempre em ordem binária UTF-8.

As operações de lista SOAP e REST retornam um documento XML que contém nomes de chaves correspondentes e informações sobre o objeto identificado em cada chave.

#### Note

O suporte de SOAP via HTTP está obsoleto, mas continua disponível via HTTPS. Novos recursos do Amazon S3 não serão compatíveis com o SOAP. Recomendamos que você use o REST API ou os SDKs da AWS.

Grupos de chaves que compartilham um prefixo terminado por um delimitador especial podem ser rolados para cima pelo prefixo comum para fins de listagem. Isso permite que aplicativos organizem e naveguem por suas chaves hierarquicamente, de maneira muito similar a como você organizaria seus arquivos em diretórios em um sistema de arquivos. Por exemplo, para estender o bucket de dicionário para conter mais do que apenas palavras inglesas, você poderia formar chaves prefixando cada palavra com seu idioma e um delimitador, tal como “francês/lógico”. Usando esse esquema de nomeação e o recurso hierárquico de listagem, você poderia recuperar uma lista somente de palavras francesas. Você também poderia pesquisar a parte superior da lista de idiomas sem ter que iterar com todas as chaves de iteração lexicográficas.

Para obter mais informações sobre esse aspecto de listagem, consulte [Lista hierárquica de chaves usando um prefixo e um delimitador \(p. 231\)](#).

#### Implementação eficiente de lista

O desempenho de lista não é substancialmente afetado pelo número total de chaves em seu bucket, nem por presença nem por ausência de prefixo, de marcador, de maxkeys ou de argumentos de delimitador. Para obter informações sobre como melhorar o desempenho geral de bucket, incluindo a operação de lista, consulte [Orientações sobre desempenho e taxa de solicitações \(p. 595\)](#).

## Percorrer os resultados de várias páginas

Como os buckets podem conter um número praticamente ilimitado de chaves, os resultados completos de uma consulta de lista podem ser muito extensos. Para gerenciar grandes conjuntos de resultados, a API do Amazon S3 oferece suporte à paginação para separá-los em várias respostas. Cada resposta de chaves de lista retorna uma página com até 1.000 chaves com um indicador apontando se a resposta está truncada. Você envia uma série de requisições de chaves de lista até que você receba todas as chaves. As bibliotecas wrapper do AWS SDK fornecem a mesma paginação.

Os exemplos de Java e .NET SDK a seguir mostram como usar paginação ao listar chaves em um bucket:

- [Listagem de chaves usando o AWS SDK for Java \(p. 233\)](#)
- [Listagem de chaves usando o AWS SDK para .NET \(p. 234\)](#)

#### Recursos relacionados

- [Usar os AWS SDKs, a CLI e os Explorers \(p. 639\)](#)

## [Lista hierárquica de chaves usando um prefixo e um delimitador](#)

Os parâmetros de prefixo e delimitador limitam o tipo de resultados retornados pela operação de lista. O prefixo limita os resultados para apenas as chaves que começam com o prefixo especificado e o delimitador faz com que a lista role para cima todas as chaves que compartilham um prefixo comum em um único resultado de lista de sumário.

A finalidade dos parâmetros de prefixo e delimitador é ajudá-lo a organizar e navegar, hierarquicamente, por suas chaves. Para fazer isso, escolha primeiro um delimitador para seu bucket, tal como barra (/), que você não prevê que apareça em nomes de chave. Em seguida, crie seus nomes de chave concatenando todos os níveis de conteúdo de hierarquia, separando cada nível com o delimitador.

Por exemplo, se você estava armazenando informações sobre cidades, pode, naturalmente, organizá-las por continente, país, província ou estado. Como esses nomes geralmente não usam pontuação, você pode selecionar a barra (/) como delimitador. Os seguintes exemplos usam uma barra (/) como delimitador.

- Europa/França/Aquitaine/Bordeaux
- América do Norte/Canadá/Quebec/Montreal
- América do Norte /EUA/Washington/Bellevue
- América do Norte /EUA/Washington/Seattle

Se você armazenou dados de cada cidade do mundo desta forma, ficaria estranho gerenciar um namespace plano de chave. Usando `Prefix` e `Delimiter` com a operação de lista, você pode usar a hierarquia que criou para listar seus dados. Por exemplo, para listar todos os estados nos EUA, defina `Delimiter='/'` e `Prefix='América do Norte/USA'`. Para listar todas as províncias no Canadá para as quais você tenha dados, defina `Delimiter='/'` e `Prefix='América do Norte/Canadá'`.

Uma solicitação de lista com um delimitador permite pesquisar a hierarquia em apenas um nível, pulando e resumindo as (possivelmente milhões de) chaves aninhadas em níveis mais profundos. Por exemplo, suponha que você tenha um bucket (`ExampleBucket`) as seguintes chaves.

```
sample.jpg
photos/2006/January/sample.jpg
photos/2006/February/sample2.jpg
photos/2006/February/sample3.jpg
photos/2006/February/sample4.jpg
```

O bucket de exemplo tem somente `sample.jpg` o objeto no nível raiz. Para listar somente os objetos no nível raiz no bucket, você envia uma solicitação GET no bucket com o caractere delimitador "/". Em resposta, o Amazon S3 retorna a chave do objeto `sample.jpg` porque ela não contém o caractere delimitador "/". Toda as outras chaves contêm o caractere delimitador. O Amazon S3 agrupa essas chaves e retorna um único elemento com valor de prefixo `CommonPrefixes photos/` que é uma substring do início dessas chaves até a primeira ocorrência do delimitador especificado.

### Example

```
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>ExampleBucket</Name>
  <Prefix></Prefix>
  <Marker></Marker>
  <MaxKeys>1000</MaxKeys>
  <Delimiter>/</Delimiter>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>sample.jpg</Key>
    <LastModified>2011-07-24T19:39:30.000Z</LastModified>
    <ETag>"d1a7fb5eab1c16cb4f7cf341cf188c3d"</ETag>
    <Size>6</Size>
    <Owner>
      <ID>75cc57f09aa0c8caeab4f8c24e99d10f8e7faebf76c078efc7c6caea54ba06a</ID>
      <DisplayName>displayname</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <CommonPrefixes>
    <Prefix>photos/</Prefix>
  </CommonPrefixes>
</ListBucketResult>
```

## Listagem de chaves usando o AWS SDK for Java

### Example

O exemplo a seguir lista as chaves de objeto em um bucket. O exemplo usa a paginação para recuperar um conjunto de chaves de objeto. Se houver mais chaves a retornar após a primeira página, o Amazon S3 incluirá um token de continuação na resposta. O exemplo usa o token de continuação na solicitação subsequente para buscar o próximo conjunto de chaves de objeto.

Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-
developer-guide/blob/master/LICENSE-SAMPLECODE.)

import java.io.IOException;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ListObjectsV2Request;
import com.amazonaws.services.s3.model.ListObjectsV2Result;
import com.amazonaws.services.s3.model.S3ObjectSummary;

public class ListKeys {

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

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

            System.out.println("Listing objects");

            // maxKeys is set to 2 to demonstrate the use of
            // ListObjectsV2Result.getNextContinuationToken()
            ListObjectsV2Request req = new
ListObjectsV2Request().withBucketName(bucketName).withMaxKeys(2);
            ListObjectsV2Result result;

            do {
                result = s3Client.listObjectsV2(req);

                for (S3ObjectSummary objectSummary : result.getObjectSummaries()) {
                    System.out.printf(" - %s (size: %d)\n",
objectSummary.getKey(),
objectSummary.getSize());
                }
                // If there are more than maxKeys keys in the bucket, get a continuation
token
                // and list the next objects.
                String token = result.getNextContinuationToken();
                System.out.println("Next Continuation Token: " + token);
                req.setContinuationToken(token);
            } while (result.isTruncated());
        }
        catch(AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
```

```
// it, so it returned an error response.  
e.printStackTrace();  
}  
catch(SdkClientException e) {  
    // Amazon S3 couldn't be contacted for a response, or the client  
    // couldn't parse the response from Amazon S3.  
    e.printStackTrace();  
}  
}  
}
```

## Listagem de chaves usando o AWS SDK para .NET

### Example

O exemplo do C# a seguir lista as chaves de objeto para um bucket. No exemplo, usamos a paginação para recuperar um conjunto de chaves de objeto. Se houver mais chaves a retornar, o Amazon S3 incluirá um token de continuação na resposta. O código usa o token de continuação na solicitação subsequente para buscar o próximo conjunto de chaves de objeto.

Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.  
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-  
developer-guide/blob/master/LICENSE-SAMPLECODE.)  
  
using Amazon.S3;  
using Amazon.S3.Model;  
using System;  
using System.Threading.Tasks;  
  
namespace Amazon.DocSamples.S3  
{  
    class ListObjectsTest  
    {  
        private const string bucketName = "*** bucket name ***";  
        // Specify your bucket region (an example region is shown).  
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;  
  
        private static IAmazonS3 client;  
  
        public static void Main()  
        {  
            client = new AmazonS3Client(bucketRegion);  
            ListingObjectsAsync().Wait();  
        }  
  
        static async Task ListingObjectsAsync()  
        {  
            try  
            {  
                ListObjectsV2Request request = new ListObjectsV2Request  
                {  
                    BucketName = bucketName,  
                    MaxKeys = 10  
                };  
                ListObjectsV2Response response;  
                do  
                {  
                    response = await client.ListObjectsV2Async(request);  
                    if (response.IsTruncated)  
                        request.ContinuationToken = response.NextContinuationToken;  
                } while (response.IsTruncated);  
            }  
        }  
    }  
}
```

```
// Process the response.
foreach (S3Object entry in response.S3Objects)
{
    Console.WriteLine("key = {0} size = {1}",
        entry.Key, entry.Size);
}
Console.WriteLine("Next Continuation Token: {0}",
response.NextContinuationToken);
request.ContinuationToken = response.NextContinuationToken;
} while (response.IsTruncated);
}
catch (AmazonS3Exception amazonS3Exception)
{
    Console.WriteLine("S3 error occurred. Exception: " +
amazonS3Exception.ToString());
    Console.ReadKey();
}
catch (Exception e)
{
    Console.WriteLine("Exception: " + e.ToString());
    Console.ReadKey();
}
}
}
}
```

## Listagem de chaves usando o AWS SDK para PHP

Este tópico orienta sobre o uso de classes da versão 3 do AWS SDK para PHP para listar chaves de objeto contidas em um bucket do Amazon S3.

Este tópico pressupõe que você já está seguindo as instruções para [Usar o AWS SDK para PHP e executar exemplos do PHP \(p. 649\)](#) e tem o AWS SDK para PHP devidamente instalado.

Para listar as chaves de objeto contidas em um bucket utilizando o AWS SDK para PHP você deve primeiro listar objetos contidos no bucket e então extrair a chave de cada um dos objetos listados. Ao listar objetos em um bucket, você tem a opção de usar o método de baixo nível [Aws\S3\S3Client::listObjects\(\)](#) ou a classe de alto nível [Aws\ResultPaginator](#).

O método de baixo nível `listObjects()` mapeia para a API REST subjacente do Amazon S3. Cada solicitação `listObjects()` retorna uma página de até 1.000 objetos. Se houver mais de 1.000 objetos no bucket, a resposta será truncada e você precisará enviar outra solicitação `listObjects()` para recuperar o conjunto seguinte de 1.000 objetos.

Você pode usar o paginador de alto nível `ListObjects` para facilitar a tarefa de listar objetos contidos em um bucket. Para usar o paginador `ListObjects` para criar uma lista de objetos, você executa o método `getPaginator()` do cliente do Amazon S3 que é herdado da classe `Aws\AwsClientInterface` com o comando `ListObjects` como o primeiro argumento e uma matriz que contém os objetos retornados do bucket especificado como o segundo argumento. Quando usado como paginador `ListObjects`, o método `getPaginator()` retorna todos os objetos contidos em um bucket. Não há limite de 1.000 objetos, de maneira que você não precisa se preocupar se a resposta é truncada ou não.

As tarefas a seguir orientam sobre a utilização de métodos PHP do cliente do Amazon S3 para listar objetos contidos em um bucket a partir dos quais você pode listar as chaves do objeto.

### Example de listagem de chaves de objeto

O exemplo PHP a seguir demonstra como listar as chaves a partir de um bucket especificado. Ele mostra como usar o método de alto nível `getIterator()` para listar os objetos em um bucket e como extrair

a chave de cada um dos objetos na lista. Ele mostra como usar o método de alto nível `listObjects()` para listar os objetos em um bucket e como extrair a chave de cada um dos objetos na lista retornada. Para obter informações sobre a execução de exemplos PHP neste guia, vá para [Executar exemplos do PHP \(p. 649\)](#).

```
<?php

require 'vendor/autoload.php';

use Aws\S3\S3Client;
use Aws\S3\Exception\S3Exception;

$bucket = '*** Your Bucket Name ***';

// Instantiate the client.
$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// Use the high-level iterators (returns ALL of your objects).
try {
    $results = $s3->getPaginator('ListObjects', [
        'Bucket' => $bucket
    ]);

    foreach ($results as $result) {
        foreach ($result['Contents'] as $object) {
            echo $object['Key'] . PHP_EOL;
        }
    }
} catch (S3Exception $e) {
    echo $e->getMessage() . PHP_EOL;
}

// Use the plain API (returns ONLY up to 1000 of your objects).
try {
    $objects = $s3->listObjects([
        'Bucket' => $bucket
    ]);
    foreach ($objects['Contents'] as $object) {
        echo $object['Key'] . PHP_EOL;
    }
} catch (S3Exception $e) {
    echo $e->getMessage() . PHP_EOL;
}
```

## Recursos relacionados

- [AWS SDK para PHP para classe Aws\S3\S3Client do Amazon S3](#)
- [Paginadores](#)
- [Documentação do AWS SDK para PHP](#)

## Listagem de chaves usando a API REST

Você pode usar o AWS SDK para listar chaves de objeto de um bucket. Contudo, se o seu aplicativo exigir, você pode enviar solicitações REST diretamente. Você pode enviar uma solicitação GET para retornar alguns ou todos os objetos em um bucket ou pode usar critérios de seleção para obter um subconjunto de objetos em um bucket. Para obter mais informações, acesse [GET Bucket \(Lista Objetos\) Versão 2](#)

## Excluir objetos

### Tópicos

- [Excluir de objetos de um bucket habilitado para versão \(p. 237\)](#)
- [Excluir objetos de um bucket com MFA habilitada \(p. 237\)](#)
- [Recursos relacionados \(p. 238\)](#)
- [Excluir um objeto por solicitação \(p. 238\)](#)
- [Excluir vários objetos por solicitação \(p. 244\)](#)

Você pode excluir um ou mais objetos diretamente no Amazon S3. Você tem as seguintes opções para excluir um objeto:

- Excluir um único objeto — o Amazon S3 fornece a API DELETE que você pode usar para excluir um objeto em uma única solicitação HTTP.
- Excluir vários objetos — o Amazon S3 também fornece a API para exclusão de vários objetos que você pode usar para excluir até 1000 objetos em uma única solicitação HTTP.

Ao excluir objetos de um bucket que não está habilitado para versão, você fornece somente o nome da chave do objeto. No entanto, ao excluir objetos de um bucket habilitado para versão, é possível fornecer o ID de versão do objeto para excluir uma versão específica do objeto.

### Excluir objetos de um bucket habilitado para versão

Se seu bucket for habilitado para versão, várias versões do mesmo objeto poderão existir no bucket. Ao trabalhar com buckets habilitados para versão a API de exclusão permite as seguintes opções:

- Especificar uma solicitação de exclusão não versionada — Você especifica somente a chave do objeto, e não o ID de versão. Nesse caso, o Amazon S3 cria um marcador de exclusão e retorna o ID de versão na resposta. Isso faz com que o objeto desapareça do bucket. Para obter informações sobre versionamento de objetos e sobre o conceito de marcador de exclusão, consulte [Versionamento de objeto \(p. 111\)](#).
- Especificar uma solicitação de exclusão versionada — Você especifica a chave e um ID de versão. Nesse caso, os dois resultados a seguir são possíveis:
  - Se o ID de versão for mapeado para uma versão de objeto específica, o Amazon S3 excluirá versão específica do objeto.
  - Se o ID de versão for mapeado para o marcador de exclusão do objeto em questão, o Amazon S3 excluirá o marcador de exclusão. Isso faz com que o objeto reapareça no bucket.

### Excluir objetos de um bucket com MFA habilitada

Ao excluir objetos de um bucket habilitado para autenticação multifator (MFA, Multi-Factor Authentication), observe:

- Se você fornecer um token de MFA inválido, a solicitação falhará sempre.
- Se tiver um bucket habilitado para MFA e fizer uma solicitação de exclusão em versões (fornecendo uma chave de objeto e um ID de versão), a solicitação falhará se você não fornecer um token de MFA válido. Além disso, ao usar a API de exclusão de vários objetos em um bucket habilitado para MFA, se alguma exclusão for uma solicitação de exclusão versionada (se você especificar a chave de objeto e o ID de versão), a solicitação inteira falhará se o token de MFA não for fornecido.

Por outro lado, nos seguintes casos, a solicitação é feita com êxito:

- Se você tiver um bucket habilitado para MFA, fizer uma solicitação de exclusão sem versões (sem excluir um objeto com versões) e não fornecer um token de MFA, a exclusão será concluída.
- Se você tiver uma solicitação de exclusão de vários objetos especificando somente objetos sem versões a serem excluídos de um bucket habilitado para MFA e não fornecer um token de MFA, as exclusões serão feitas com êxito.

Para obter informações sobre exclusão de MFA, consulte [Exclusão de MFA \(p. 449\)](#).

## Recursos relacionados

- [Usar os AWS SDKs, a CLI e os Explorers \(p. 639\)](#)

## Excluir um objeto por solicitação

### Tópicos

- [Excluir objetos usando o AWS SDK for Java \(p. 238\)](#)
- [Excluir objetos usando o AWS SDK para .NET \(p. 240\)](#)
- [Excluir objetos usando o AWS SDK para PHP \(p. 243\)](#)
- [Excluir um objeto usando a API REST \(p. 244\)](#)

Para excluir um objeto por solicitação, use a API `DELETE` (consulte [DELETE objeto](#)). Para saber mais sobre exclusão de objetos, consulte [Excluir objetos \(p. 237\)](#).

Você pode usar a API REST diretamente ou as bibliotecas de wrapper fornecidas pelos SDKs da AWS que simplificam o desenvolvimento de aplicativos.

### Excluir objetos usando o AWS SDK for Java

É possível excluir um objeto de um bucket. Se tiver habilitado o versionamento no bucket, você tem as seguintes opções:

- Excluir uma versão específica do objeto ao especificar um ID de versão.
- Excluir um objeto sem especificar um ID de versão, caso em que o S3 adicionará um marcador de exclusão para o objeto.

Para obter mais informações sobre versionamento, consulte [Versionamento de objeto \(p. 111\)](#).

#### Example Exemplo 1: excluir um objeto (bucket sem versionamento)

O exemplo a seguir exclui um objeto de um bucket. O exemplo pressupõe que o bucket não esteja habilitado para versionamento e o objeto não tenha IDs de versão. Na solicitação de exclusão, especifique somente a chave do objeto e não um ID de versão. Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.  
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)  
  
import java.io.IOException;  
  
import com.amazonaws.AmazonServiceException;  
import com.amazonaws.SdkClientException;  
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
```

```
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.DeleteObjectRequest;

public class DeleteObjectNonVersionedBucket {

    public static void main(String[] args) throws IOException {
        String clientRegion = "*** Client region ***";
        String bucketName = "*** Bucket name ***";
        String keyName = "*** Key name ***";

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

            s3Client.deleteObject(new DeleteObjectRequest(bucketName, keyName));
        }
        catch(AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        }
        catch(SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

#### Example Exemplo 2: excluir um objeto (bucket com versionamento)

O exemplo a seguir exclui um objeto de um bucket com versionamento. O exemplo exclui uma versão específica do objeto ao especificar o nome da chave e o ID de versão do objeto. O exemplo faz o seguinte:

1. Adiciona um objeto de amostra ao bucket. O Amazon S3 apresenta o ID da versão do objeto recém-adicionado. O exemplo usa esse ID de versão na solicitação de exclusão.
2. Exclui a versão do objeto ao especificar o nome da chave e um ID de versão do objeto. Se não houver nenhuma outra versão do objeto, o Amazon S3 excluirá o objeto totalmente. Caso contrário, o Amazon S3 excluirá somente a versão especificada.

#### Note

Você pode obter os IDs de versão de um objeto enviando uma solicitação `ListVersions`.

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-
developer-guide/blob/master/LICENSE-SAMPLECODE.)

import java.io.IOException;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.BucketVersioningConfiguration;
import com.amazonaws.services.s3.model.DeleteVersionRequest;
import com.amazonaws.services.s3.model.PutObjectResult;
```

```
public class DeleteObjectVersionEnabledBucket {

    public static void main(String[] args) throws IOException {
        String clientRegion = "**** Client region ****";
        String bucketName = "**** Bucket name ****";
        String keyName = "**** Key name ****";

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

            // Check to ensure that the bucket is versioning-enabled.
            String bucketVersionStatus =
                s3Client.getBucketVersioningConfiguration(bucketName).getStatus();
            if(!bucketVersionStatus.equals(BucketVersioningConfiguration.ENABLED)) {
                System.out.printf("Bucket %s is not versioning-enabled.", bucketName);
            }
            else {
                // Add an object.
                PutObjectResult putResult = s3Client.putObject(bucketName, keyName, "Sample
content for deletion example.");
                System.out.printf("Object %s added to bucket %s\n", keyName, bucketName);

                // Delete the version of the object that we just created.
                System.out.println("Deleting versioned object " + keyName);
                s3Client.deleteVersion(new DeleteVersionRequest(bucketName, keyName,
putResult.getVersionId()));
                System.out.printf("Object %s, version %s deleted\n", keyName,
putResult.getVersionId());
            }
        }
        catch(AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        }
        catch(SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

## Excluir objetos usando o AWS SDK para .NET

Ao excluir o objeto de um bucket sem versionamento, o objeto é removido. Se tiver habilitado o versionamento no bucket, você tem as seguintes opções:

- Excluir uma versão específica de um objeto ao especificar um ID de versão.
- Excluir um objeto sem especificar um ID de versão. O Amazon S3 adiciona um marcador de exclusão. Para obter mais informações sobre marcadores de exclusão, consulte [Versionamento de objeto \(p. 111\)](#).

Os exemplos a seguir mostram como excluir um objeto de buckets com e sem versionamento. Para obter mais informações sobre versionamento, consulte [Versionamento de objeto \(p. 111\)](#).

### Example Excluir um objeto de um bucket sem versionamento

O exemplo do C# a seguir exclui um objeto de um bucket sem versionamento. O exemplo pressupõe que os objetos não têm IDs de versão, portanto, você não especifica IDs de versão. Especifique somente

a chave do objeto. Para obter informações sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-
developer-guide/blob/master/LICENSE-SAMPLECODE.)

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

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

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            DeleteObjectNonVersionedBucketAsync().Wait();
        }

        private static async Task DeleteObjectNonVersionedBucketAsync()
        {
            try
            {
                var deleteObjectRequest = new DeleteObjectRequest
                {
                    BucketName = bucketName,
                    Key = keyName
                };

                Console.WriteLine("Deleting an object");
                await client.DeleteObjectAsync(deleteObjectRequest);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when writing
an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
        }
    }
}
```

#### Example Excluir um objeto de um bucket com versionamento

O exemplo do C# a seguir exclui um objeto de um bucket com versionamento. Ele exclui uma versão específica do objeto ao especificar o nome da chave e o ID de versão do objeto.

O código realiza as seguintes tarefas:

1. Permite o versionamento no bucket que você especificar (se o versionamento já estiver habilitado, isso não terá efeito).

2. Adiciona um objeto de exemplo ao bucket. Em resposta, o Amazon S3 retorna o ID de versão do objeto recém-adicionado. O exemplo usa esse ID de versão na solicitação de exclusão.
3. Exclui o objeto de exemplo ao especificar o nome da chave e um ID de versão do objeto.

Note

Você também pode obter o ID de versão de um objeto enviando uma solicitação `ListVersions`:

```
var listResponse = client.ListVersions(new ListVersionsRequest { BucketName =  
    bucketName, Prefix = keyName });
```

Para obter informações sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.  
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-  
developer-guide/blob/master/LICENSE-SAMPLECODE.)  
  
using Amazon.S3;  
using Amazon.S3.Model;  
using System;  
using System.Threading.Tasks;  
  
namespace Amazon.DocSamples.S3  
{  
    class DeleteObjectVersion  
    {  
        private const string bucketName = "*** versioning-enabled bucket name ***";  
        private const string keyName = "*** Object Key Name ***";  
        // Specify your bucket region (an example region is shown).  
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;  
        private static IAmazonS3 client;  
  
        public static void Main()  
        {  
            client = new AmazonS3Client(bucketRegion);  
            CreateAndDeleteObjectVersionAsync().Wait();  
        }  
  
        private static async Task CreateAndDeleteObjectVersionAsync()  
        {  
            try  
            {  
                // Add a sample object.  
                string versionID = await PutAnObject(keyName);  
  
                // Delete the object by specifying an object key and a version ID.  
                DeleteObjectRequest request = new DeleteObjectRequest  
                {  
                    BucketName = bucketName,  
                    Key = keyName,  
                    VersionId = versionID  
                };  
                Console.WriteLine("Deleting an object");  
                await client.DeleteObjectAsync(request);  
            }  
            catch (AmazonS3Exception e)  
            {  
                Console.WriteLine("Error encountered on server. Message:'{0}' when writing  
an object", e.Message);  
            }  
        }  
    }  
}
```

```
        catch (Exception e)
        {
            Console.WriteLine("Unknown encountered on server. Message:{0}' when
writing an object", e.Message);
        }
    }

    static async Task<string> PutAnObject(string objectKey)
{
    PutObjectRequest request = new PutObjectRequest
    {
        BucketName = bucketName,
        Key = objectKey,
        ContentBody = "This is the content body!"
    };
    PutObjectResponse response = await client.PutObjectAsync(request);
    return response.VersionId;
}
}
```

## Excluir objetos usando o AWS SDK para PHP

Este tópico mostra como usar classes da versão 3 do AWS SDK para PHP para excluir um objeto de um bucket sem versionamento. Para obter informações sobre como excluir um objeto de um bucket com versões, consulte [Excluir um objeto usando a API REST \(p. 244\)](#).

Este tópico pressupõe que você já está seguindo as instruções para [Usar o AWS SDK para PHP e executar exemplos do PHP \(p. 649\)](#) e tem o AWS SDK para PHP devidamente instalado.

O exemplo de PHP a seguir exclui um objeto de um bucket. Como esse exemplo mostra como excluir objetos de bucket sem versionamento, ele oferece apenas o nome do bucket e a chave do objeto (não um ID de versão) na solicitação de exclusão. Para obter informações sobre a execução dos exemplos de PHP neste guia, consulte [Executar exemplos do PHP \(p. 649\)](#).

```
<?php

require 'vendor/autoload.php';

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// Delete an object from the bucket.
$s3->deleteObject([
    'Bucket' => $bucket,
    'Key'     => $keyname
]);
```

## Recursos relacionados

- [AWS SDK para PHP para classe Aws\S3\S3Client do Amazon S3](#)
- [Documentação do AWS SDK para PHP](#)

## Excluir um objeto usando a API REST

Você pode usar os SDKs da AWS para excluir um objeto. Contudo, se o seu aplicativo exigir, você pode enviar solicitações REST diretamente. Para obter mais informações, visite [DELETE objeto](#) no Amazon Simple Storage Service API Reference.

## Excluir vários objetos por solicitação

### Tópicos

- [Excluir vários objetos usando o AWS SDK for Java \(p. 244\)](#)
- [Excluir vários objetos usando o AWS SDK para .NET \(p. 248\)](#)
- [Excluir vários objetos usando o AWS SDK para PHP \(p. 253\)](#)
- [Excluir vários objetos usando a API REST \(p. 255\)](#)

O Amazon S3 fornece a API de exclusão de vários objetos (consulte [Excluir – Exclusão de vários objetos](#)), que permite excluir vários objetos em uma única solicitação. A API oferece suporte a dois modos para a resposta: detalhado e silencioso. Por padrão, a operação usa o modo detalhado. No modo detalhado, a resposta inclui o resultado da exclusão de cada chave especificada na sua solicitação. No modo silencioso, a resposta inclui apenas as chaves para as quais a operação de exclusão encontrou um erro. Se todas as chaves forem excluídas com êxito ao usar o modo silencioso, o Amazon S3 retornará uma resposta vazia.

Para saber mais sobre exclusão de objetos, consulte [Excluir objetos \(p. 237\)](#).

Você pode usar a API REST diretamente ou usar os AWS SDKs.

## Excluir vários objetos usando o AWS SDK for Java

O AWS SDK for Java fornece o método `AmazonS3Client.deleteObjects()` para excluir vários objetos. Para cada objeto que você deseja excluir, especifique o nome da chave. Se o bucket estiver habilitado para o versionamento, você tem as seguintes opções:

- Especifique somente o nome da chave do objeto. O Amazon S3 adicionará um marcador de exclusão ao objeto.
- Especifique o nome da chave do objeto e o ID de versão a serem excluídos. O Amazon S3 excluirá a versão especificada do objeto.

### Example

O exemplo a seguir usa a API de Exclusão de vários objetos para excluir objetos de um bucket sem versionamento. O exemplo faz upload de objetos de exemplo no bucket e, em seguida, usa o método `AmazonS3Client.deleteObjects()` para excluir os objetos em uma única solicitação. Na `DeleteObjectsRequest`, o exemplo especifica apenas os nomes de chaves de objeto porque os objetos não têm IDs de versão.

Para obter mais informações sobre exclusão de objetos, consulte [Excluir objetos \(p. 237\)](#). Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.  
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)  
  
import java.io.IOException;  
import java.util.ArrayList;  
  
import com.amazonaws.AmazonServiceException;
```

```
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.DeleteObjectsRequest;
import com.amazonaws.services.s3.model.DeleteObjectsRequest.KeyVersion;
import com.amazonaws.services.s3.model.DeleteObjectsResult;

public class DeleteMultipleObjectsNonVersionedBucket {

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

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

            // Upload three sample objects.
            ArrayList<KeyVersion> keys = new ArrayList<KeyVersion>();
            for (int i = 0; i < 3; i++) {
                String keyName = "delete object example " + i;
                s3Client.putObject(bucketName, keyName, "Object number " + i + " to be
deleted.");
                keys.add(new KeyVersion(keyName));
            }
            System.out.println(keys.size() + " objects successfully created.");

            // Delete the sample objects.
            DeleteObjectsRequest multiObjectDeleteRequest = new
DeleteObjectsRequest(bucketName)
                .withKeys(keys)
                .withQuiet(false);

            // Verify that the objects were deleted successfully.
            DeleteObjectsResult delObjRes =
s3Client.deleteObjects(multiObjectDeleteRequest);
            int successfulDeletes = delObjRes.getDeletedObjects().size();
            System.out.println(successfulDeletes + " objects successfully deleted.");
        }
        catch(AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        }
        catch(SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

### Example

O exemplo a seguir usa a API de Exclusão de vários objetos para excluir objetos de um bucket com versionamento. Ela faz o seguinte:

1. Cria objetos de exemplo e em seguida os exclui, especificando o nome da chave e o ID de versão para cada objeto a excluir. A operação exclui somente as versões especificadas do objeto.
2. Cria objetos de exemplo e os exclui especificando somente os nomes de chave. Como o exemplo não especifica os IDs de versão, a operação adiciona um marcador de exclusão para cada objeto, sem

excluir nenhuma versão específica do objeto. Depois de os marcadores de exclusão serem adicionados, esses objetos não aparecerão em Console de gerenciamento da AWS.

3. Remova os marcadores de exclusão especificando as chaves de objeto e os IDs de versão dos marcadores de exclusão. A operação exclui os marcadores de exclusão, o que resulta no reaparecimento de objetos no Console de gerenciamento da AWS.

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.BucketVersioningConfiguration;
import com.amazonaws.services.s3.model.DeleteObjectsRequest;
import com.amazonaws.services.s3.model.DeleteObjectsRequest.KeyVersion;
import com.amazonaws.services.s3.model.DeleteObjectsResult;
import com.amazonaws.services.s3.model.DeleteObjectsResult.DeletedObject;
import com.amazonaws.services.s3.model.PutObjectResult;

public class DeleteMultipleObjectsVersionEnabledBucket {
    private static AmazonS3 S3_CLIENT;
    private static String VERSIONED_BUCKET_NAME;

    public static void main(String[] args) throws IOException {
        String clientRegion = "**** Client region ****";
        VERSIONED_BUCKET_NAME = "**** Bucket name ****";

        try {
            S3_CLIENT = AmazonS3ClientBuilder.standard()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(clientRegion)
                .build();

            // Check to make sure that the bucket is versioning-enabled.
            String bucketVersionStatus =
                S3_CLIENT.getBucketVersioningConfiguration(VERSIONED_BUCKET_NAME).getStatus();
            if (!bucketVersionStatus.equals(BucketVersioningConfiguration.ENABLED)) {
                System.out.printf("Bucket %s is not versioning-enabled.", VERSIONED_BUCKET_NAME);
            }
            else {
                // Upload and delete sample objects, using specific object versions.
                uploadAndDeleteObjectsWithVersions();

                // Upload and delete sample objects without specifying version IDs.
                // Amazon S3 creates a delete marker for each object rather than deleting
                // specific versions.
                DeleteObjectsResult unversionedDeleteResult =
                    uploadAndDeleteObjectsWithoutVersions();

                // Remove the delete markers placed on objects in the non-versioned create/
                delete method.
                multiObjectVersionedDeleteRemoveDeleteMarkers(unversionedDeleteResult);
            }
        }
        catch(AmazonServiceException e) {
```

```
// The call was transmitted successfully, but Amazon S3 couldn't process
// it, so it returned an error response.
e.printStackTrace();
}
catch(SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}

private static void uploadAndDeleteObjectsWithVersions() {
    System.out.println("Uploading and deleting objects with versions specified.");

    // Upload three sample objects.
    ArrayList<KeyVersion> keys = new ArrayList<KeyVersion>();
    for (int i = 0; i < 3; i++) {
        String keyName = "delete object without version ID example " + i;
        PutObjectResult putResult = S3_CLIENT.putObject(VERSIONED_BUCKET_NAME, keyName,
            "Object number " + i + " to be deleted.");
        // Gather the new object keys with version IDs.
        keys.add(new KeyVersion(keyName, putResult.getVersionId()));
    }

    // Delete the specified versions of the sample objects.
    DeleteObjectsRequest multiObjectDeleteRequest = new
DeleteObjectsRequest(VERSIONED_BUCKET_NAME)
    .withKeys(keys)
    .withQuiet(false);

    // Verify that the object versions were successfully deleted.
    DeleteObjectsResult delObjRes = S3_CLIENT.deleteObjects(multiObjectDeleteRequest);
    int successfulDeletes = delObjRes.getDeletedObjects().size();
    System.out.println(successfulDeletes + " objects successfully deleted");
}

private static DeleteObjectsResult uploadAndDeleteObjectsWithoutVersions() {
    System.out.println("Uploading and deleting objects with no versions specified.");

    // Upload three sample objects.
    ArrayList<KeyVersion> keys = new ArrayList<KeyVersion>();
    for (int i = 0; i < 3; i++) {
        String keyName = "delete object with version ID example " + i;
        S3_CLIENT.putObject(VERSIONED_BUCKET_NAME, keyName, "Object number " + i + " to
be deleted.");
        // Gather the new object keys without version IDs.
        keys.add(new KeyVersion(keyName));
    }

    // Delete the sample objects without specifying versions.
    DeleteObjectsRequest multiObjectDeleteRequest = new
DeleteObjectsRequest(VERSIONED_BUCKET_NAME).withKeys(keys)
    .withQuiet(false);

    // Verify that delete markers were successfully added to the objects.
    DeleteObjectsResult delObjRes = S3_CLIENT.deleteObjects(multiObjectDeleteRequest);
    int successfulDeletes = delObjRes.getDeletedObjects().size();
    System.out.println(successfulDeletes + " objects successfully marked for deletion
without versions.");
    return delObjRes;
}

private static void multiObjectVersionedDeleteRemoveDeleteMarkers(DeleteObjectsResult
response) {
    List<KeyVersion> keyList = new ArrayList<KeyVersion>();
    for (DeletedObject deletedObject : response.getDeletedObjects()) {
```

```
// Note that the specified version ID is the version ID for the delete marker.  
keyList.add(new KeyVersion(deletedObject.getKey(),  
deletedObject.getDeleteMarkerVersionId()));  
}  
// Create a request to delete the delete markers.  
DeleteObjectsRequest deleteRequest = new  
DeleteObjectsRequest(VERSIONED_BUCKET_NAME).withKeys(keyList);  
  
// Delete the delete markers, leaving the objects intact in the bucket.  
DeleteObjectsResult delObjRes = S3_CLIENT.deleteObjects(deleteRequest);  
int successfulDeletes = delObjRes.getDeletedObjects().size();  
System.out.println(successfulDeletes + " delete markers successfully deleted");  
}  
}
```

## Excluir vários objetos usando o AWS SDK para .NET

O AWS SDK para .NET fornece um método conveniente para excluir vários objetos: `DeleteObjects`. Para cada objeto que você deseja excluir, especifique o nome da chave e a versão do objeto. Se o bucket não for habilitado para versionamento, você especifica `null` para o ID de versão. Se uma exceção ocorrer, reveja a resposta `DeleteObjectsException` para determinar quais objetos não foram excluídos e por quê.

Example Excluir vários objetos de um bucket sem versionamento

O exemplo do C# a seguir usa a API de Exclusão de vários objetos para excluir objetos de um bucket sem versionamento. O exemplo faz upload dos objetos de exemplo no bucket e, em seguida, usa o método `DeleteObjects` para excluir os objetos em uma única solicitação. Na `DeleteObjectsRequest`, o exemplo especifica apenas os nomes das chaves dos objetos porque os IDs das versões são nulos.

Para obter informações sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.  
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-  
developer-guide/blob/master/LICENSE-SAMPLECODE.)  
  
using Amazon.S3;  
using Amazon.S3.Model;  
using System;  
using System.Collections.Generic;  
using System.Threading.Tasks;  
  
namespace Amazon.DocSamples.S3  
{  
    class DeleteMultipleObjectsNonVersionedBucketTest  
    {  
        private const string bucketName = "*** versioning-enabled bucket name ***";  
        // Specify your bucket region (an example region is shown).  
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;  
        private static IAmazonS3 s3Client;  
  
        public static void Main()  
        {  
            s3Client = new AmazonS3Client(bucketRegion);  
            MultiObjectDeleteAsync().Wait();  
        }  
  
        static async Task MultiObjectDeleteAsync()  
        {  
            // Create sample objects (for subsequent deletion).  
            var keysAndVersions = await PutObjectsAsync(3);  
        }  
    }  
}
```

```
// a. multi-object delete by specifying the key names and version IDs.
DeleteObjectsRequest multiObjectDeleteRequest = new DeleteObjectsRequest
{
    BucketName = bucketName,
    Objects = keysAndVersions // This includes the object keys and null version
IDs.
};

// You can add specific object key to the delete request using the .AddKey.
// multiObjectDeleteRequest.AddKey("TickerReference.csv", null);
try
{
    DeleteObjectsResponse response = await
s3Client.DeleteObjectsAsync(multiObjectDeleteRequest);
    Console.WriteLine("Successfully deleted all the {0} items",
response.DeletedObjects.Count);
}
catch (DeleteObjectsException e)
{
    PrintDeletionErrorStatus(e);
}
}

private static void PrintDeletionErrorStatus(DeleteObjectsException e)
{
    // var errorResponse = e.ErrorResponse;
    DeleteObjectsResponse errorResponse = e.Response;
    Console.WriteLine("x {0}", errorResponse.DeletedObjects.Count);

    Console.WriteLine("No. of objects successfully deleted = {0}",
errorResponse.DeletedObjects.Count);
    Console.WriteLine("No. of objects failed to delete = {0}",
errorResponse.DeleteErrors.Count);

    Console.WriteLine("Printing error data...");
    foreach (DeleteError deleteError in errorResponse.DeleteErrors)
    {
        Console.WriteLine("Object Key: {0}\t{1}\t{2}", deleteError.Key,
deleteError.Code, deleteError.Message);
    }
}

static async Task<List<KeyVersion>> PutObjectsAsync(int number)
{
    List<KeyVersion> keys = new List<KeyVersion>();
    for (int i = 0; i < number; i++)
    {
        string key = "ExampleObject-" + new System.Random().Next();
        PutObjectRequest request = new PutObjectRequest
        {
            BucketName = bucketName,
            Key = key,
            ContentBody = "This is the content body!",
        };

        PutObjectResponse response = await s3Client.PutObjectAsync(request);
        KeyVersion keyVersion = new KeyVersion
        {
            Key = key,
            // For non-versioned bucket operations, we only need object key.
            // VersionId = response.VersionId
        };
        keys.Add(keyVersion);
    }
    return keys;
}
```

```
    }  
}
```

#### Example Exclusão de vários objetos para um bucket com versionamento

O exemplo do C# a seguir usa a API de exclusão de vários objetos para excluir objetos de um bucket com versionamento. O exemplo executa as seguintes ações:

1. Cria objetos de exemplo e os exclui especificando o nome da chave e o ID de versão para cada objeto. A operação exclui versões específicas dos objetos.
2. Cria objetos de exemplo e os exclui especificando somente os nomes de chave. Como o exemplo não especifica IDs de versão, a operação somente adiciona marcadores de exclusão. Ela não exclui nenhuma versão específica dos objetos. Após a exclusão, esses objetos não aparecem no console do Amazon S3.
3. Exclui os marcadores de exclusão especificando as chaves de objeto e os IDs de versão dos marcadores de exclusão. Quando a operação exclui os marcadores de exclusão, os objetos reaparecem no console.

Para obter informações sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.  
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)  
  
using Amazon.S3;  
using Amazon.S3.Model;  
using System;  
using System.Collections.Generic;  
using System.Threading.Tasks;  
  
namespace Amazon.DocSamples.S3  
{  
    class DeleteMultipleObjVersionedBucketTest  
    {  
        private const string bucketName = "*** versioning-enabled bucket name ***";  
        // Specify your bucket region (an example region is shown).  
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;  
        private static IAmazonS3 s3Client;  
  
        public static void Main()  
        {  
            s3Client = new AmazonS3Client(bucketRegion);  
            DeleteMultipleObjectsFromVersionedBucketAsync().Wait();  
        }  
  
        private static async Task DeleteMultipleObjectsFromVersionedBucketAsync()  
        {  
  
            // Delete objects (specifying object version in the request).  
            await DeleteObjectVersionsAsync();  
  
            // Delete objects (without specifying object version in the request).  
            var deletedObjects = await DeleteObjectsAsync();  
  
            // Additional exercise - remove the delete markers S3 returned in the preceding  
            response.  
            // This results in the objects reappearing in the bucket (you can  
            // verify the appearance/disappearance of objects in the console).  
            await RemoveDeleteMarkersAsync(deletedObjects);  
        }  
}
```

```
private static async Task<List<DeletedObject>> DeleteObjectsAsync()
{
    // Upload the sample objects.
    var keysAndVersions2 = await PutObjectsAsync(3);

    // Delete objects using only keys. Amazon S3 creates a delete marker and
    // returns its version ID in the response.
    List<DeletedObject> deletedObjects = await
NonVersionedDeleteAsync(keysAndVersions2);
    return deletedObjects;
}

private static async Task DeleteObjectVersionsAsync()
{
    // Upload the sample objects.
    var keysAndVersions1 = await PutObjectsAsync(3);

    // Delete the specific object versions.
    await VersionedDeleteAsync(keysAndVersions1);
}

private static void PrintDeletionReport(DeleteObjectsException e)
{
    var errorResponse = e.Response;
    Console.WriteLine("No. of objects successfully deleted = {0}",
errorResponse.DeletedObjects.Count);
    Console.WriteLine("No. of objects failed to delete = {0}",
errorResponse.DeleteErrors.Count);
    Console.WriteLine("Printing error data...");
    foreach (var deleteError in errorResponse.DeleteErrors)
    {
        Console.WriteLine("Object Key: {0}\t{1}\t{2}", deleteError.Key,
deleteError.Code, deleteError.Message);
    }
}

static async Task VersionedDeleteAsync(List<KeyVersion> keys)
{
    // a. Perform a multi-object delete by specifying the key names and version
IDs.
    var multiObjectDeleteRequest = new DeleteObjectsRequest
    {
        BucketName = bucketName,
        Objects = keys // This includes the object keys and specific version IDs.
    };
    try
    {
        Console.WriteLine("Executing VersionedDelete...");
        DeleteObjectsResponse response = await
s3Client.DeleteObjectsAsync(multiObjectDeleteRequest);
        Console.WriteLine("Successfully deleted all the {0} items",
response.DeletedObjects.Count);
    }
    catch (DeleteObjectsException e)
    {
        PrintDeletionReport(e);
    }
}

static async Task<List<DeletedObject>> NonVersionedDeleteAsync(List<KeyVersion>
keys)
{
    // Create a request that includes only the object key names.
    DeleteObjectsRequest multiObjectDeleteRequest = new DeleteObjectsRequest();
    multiObjectDeleteRequest.BucketName = bucketName;
```

```
foreach (var key in keys)
{
    multiObjectDeleteRequest.AddKey(key.Key);
}
// Execute DeleteObjects - Amazon S3 add delete marker for each object
// deletion. The objects disappear from your bucket.
// You can verify that using the Amazon S3 console.
DeleteObjectsResponse response;
try
{
    Console.WriteLine("Executing NonVersionedDelete...");
    response = await s3Client.DeleteObjectsAsync(multiObjectDeleteRequest);
    Console.WriteLine("Successfully deleted all the {0} items",
response.DeletedObjects.Count);
}
catch (DeleteObjectsException e)
{
    PrintDeletionReport(e);
    throw; // Some deletes failed. Investigate before continuing.
}
// This response contains the DeletedObjects list which we use to delete the
delete markers.
return response.DeletedObjects;
}

private static async Task RemoveDeleteMarkersAsync(List<DeletedObject>
deletedObjects)
{
    var keyVersionList = new List<KeyVersion>();

    foreach (var deletedObject in deletedObjects)
    {
        KeyVersion keyVersion = new KeyVersion
        {
            Key = deletedObject.Key,
            VersionId = deletedObject.DeleteMarkerVersionId
        };
        keyVersionList.Add(keyVersion);
    }
    // Create another request to delete the delete markers.
    var multiObjectDeleteRequest = new DeleteObjectsRequest
    {
        BucketName = bucketName,
        Objects = keyVersionList
    };

    // Now, delete the delete marker to bring your objects back to the bucket.
    try
    {
        Console.WriteLine("Removing the delete markers .....");
        var deleteObjectResponse = await
s3Client.DeleteObjectsAsync(multiObjectDeleteRequest);
        Console.WriteLine("Successfully deleted all the {0} delete markers",
                deleteObjectResponse.DeletedObjects.Count);
    }
    catch (DeleteObjectsException e)
    {
        PrintDeletionReport(e);
    }
}

static async Task<List<KeyVersion>> PutObjectsAsync(int number)
{
    var keys = new List<KeyVersion>();
```

```
for (var i = 0; i < number; i++)
{
    string key = "ObjectToDelete-" + new System.Random().Next();
    PutObjectRequest request = new PutObjectRequest
    {
        BucketName = bucketName,
        Key = key,
        ContentBody = "This is the content body!",
    };

    var response = await s3Client.PutObjectAsync(request);
    KeyVersion keyVersion = new KeyVersion
    {
        Key = key,
        VersionId = response.VersionId
    };

    keys.Add(keyVersion);
}
return keys;
}
```

## Excluir vários objetos usando o AWS SDK para PHP

Este tópico mostra como usar as classes da versão 3 do AWS SDK para PHP para excluir vários objetos de buckets do Amazon S3 com e sem versionamento. Para obter mais informações sobre versionamento, consulte [Usar versionamento \(p. 448\)](#).

Este tópico pressupõe que você já está seguindo as instruções para [Usar o AWS SDK para PHP e executar exemplos do PHP \(p. 649\)](#) e tem o AWS SDK para PHP devidamente instalado.

### Example Excluir vários objetos de um bucket sem versionamento

O PHP de exemplo a seguir usa o método `deleteObjects()` para excluir vários objetos de um bucket sem versionamento.

Para obter informações sobre a execução dos exemplos de PHP neste guia, consulte [Executar exemplos do PHP \(p. 649\)](#).

```
<?php

require 'vendor/autoload.php';

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// 1. Create a few objects.
for ($i = 1; $i <= 3; $i++) {
    $s3->putObject([
        'Bucket' => $bucket,
        'Key'     => "key{$i}",
        'Body'    => "content {$i}",
    ]);
}
```

```
}

// 2. List the objects and get the keys.
$keys = $s3->listObjects([
    'Bucket' => $bucket
]) ->getPath('Contents/*/Key');

// 3. Delete the objects.
$s3->deleteObjects([
    'Bucket' => $bucket,
    'Delete' => [
        'Objects' => array_map(function ($key) {
            return ['Key' => $key];
        }), $keys
    ],
]);

```

#### Example Excluir vários objetos de um bucket habilitado para versionamento

O PHP de exemplo a seguir usa o método `deleteObjects()` para excluir vários objetos de um bucket habilitado para versionamento.

Para obter informações sobre a execução dos exemplos de PHP neste guia, consulte [Executar exemplos do PHP \(p. 649\)](#).

```
<?php

require 'vendor/autoload.php';

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// 1. Enable object versioning for the bucket.
$s3->putBucketVersioning([
    'Bucket' => $bucket,
    'Status'  => 'Enabled',
]);
 
// 2. Create a few versions of an object.
for ($i = 1; $i <= 3; $i++) {
    $s3->putObject([
        'Bucket' => $bucket,
        'Key'     => $keyname,
        'Body'    => "content {$i}",
    ]);
}

// 3. List the objects versions and get the keys and version IDs.
$versions = $s3->listObjectVersions(['Bucket' => $bucket])
    ->getPath('Versions');

// 4. Delete the object versions.
$s3->deleteObjects([
    'Bucket'  => $bucket,
    'Delete'  => [
        'Objects' => array_map(function ($version) {
            return [

```

```
        'Key'      => $version['Key'],
        'VersionId' => $version['VersionId']
    },
],
]);

echo "The following objects were deleted successfully:". PHP_EOL;
foreach ($result['Deleted'] as $object) {
    echo "Key: {$object['Key']}, VersionId: {$object['VersionId']}". PHP_EOL;
}

echo PHP_EOL . "The following objects could not be deleted:" . PHP_EOL;
foreach ($result['Errors'] as $object) {
    echo "Key: {$object['Key']}, VersionId: {$object['VersionId']}". PHP_EOL;
}

// 5. Suspend object versioning for the bucket.
$s3->putBucketVersioning([
    'Bucket' => $bucket,
    'Status' => 'Suspended',
]);

```

## Recursos relacionados

- [AWS SDK para PHP para classe Aws\S3\S3Client do Amazon S3](#)
- [Documentação do AWS SDK para PHP](#)

## Excluir vários objetos usando a API REST

Você pode usar os AWS SDKs para excluir vários objetos usando a API de Exclusão de vários objetos. Contudo, se o seu aplicativo exigir, você pode enviar solicitações REST diretamente. Para obter mais informações, visite [Excluir vários objetos](#) no Amazon Simple Storage Service API Reference.

## Selecionar conteúdo de objetos

Com o Amazon S3 Select, você pode usar instruções de linguagem de consulta estruturada (SQL) para filtrar o conteúdo de objetos do Amazon S3 e recuperar somente o subconjunto de dados necessário. Ao usar o Amazon S3 Select para filtrar esses dados, você pode reduzir a quantidade de dados transferidos pelo Amazon S3. Isso reduz o custo e a latência de recuperação desses dados.

O Amazon S3Select funciona em objetos armazenados em formato CSV, JSON ou Apache Parquet. Ele também funciona com objetos compactados com GZIP ou BZIP2 (somente para objetos CSV e JSON) e objetos criptografados no lado do servidor. Você pode especificar o formato dos resultados como CSV ou JSON e determinar como os registros do resultado são delimitados.

Expressões SQL são passadas para o Amazon S3 na solicitação. O Amazon S3 Select é compatível com um subconjunto de SQL. Para obter mais informações sobre os elementos SQL compatíveis com o Amazon S3 Select, consulte [Referência SQL para o Amazon S3 Select e o Glacier Select \(p. 685\)](#).

É possível executar consultas SQL usando SDKs da AWS, a API REST SELECT Object Content, a AWS Command Line Interface (AWS CLI) ou o console do Amazon S3. O console do Amazon S3 limita a quantidade de dados retornados para 40 MB. Para recuperar mais dados, use a AWS CLI ou a API.

## Requisitos e limites

Estes são os requisitos para o uso do Amazon S3 Select:

- É necessário ter permissão s3:GetObject para o objeto sendo consultado.

- Se o objeto sendo consultado for criptografado com uma chave de criptografia fornecida pelo cliente (SSE-C), use `https` e forneça a chave na solicitação.

Os seguintes limites se aplicam ao usar o Amazon S3 Select:

- O tamanho máximo de uma expressão SQL é 256 KB.
- O tamanho máximo de um registro no resultado é 1 MB.

Limitações adicionais são aplicáveis no uso do Amazon S3 Select com objetos Parquet:

- O Amazon S3 Select oferece suporte apenas à compactação colunar com GZIP ou Snappy. O Amazon S3 Select não oferece suporte à compactação de objetos inteiros no caso de objetos Parquet.
- O Amazon S3 Select não oferece suporte à saída do Parquet. É necessário especificar o formato de saída como CSV ou JSON.
- O tamanho máximo do bloco de arquivos não compactados é 256 MB.
- O número máximo de colunas é 100.
- É necessário usar os tipos de dados especificados no esquema do objeto.
- A seleção em um campo repetido retorna apenas o último valor.

## Criar uma solicitação

Ao criar uma solicitação, você fornece detalhes do objeto sendo consultado usando um objeto `InputSerialization`. Forneça detalhes sobre como os resultados serão retornados usando um objeto `OutputSerialization`. Inclua também a expressão SQL que o Amazon S3 usa para filtrar a solicitação.

Para obter mais informações sobre como criar uma solicitação do Amazon S3 Select, consulte [Conteúdo do objeto SELECT](#) no Amazon Simple Storage Service API Reference. Também é possível um exemplo de código do SDK nas seções a seguir.

## Erros

O Amazon S3 Select retorna um código de erro e uma mensagem de erro associada quando um problema é encontrado ao tentar executar uma consulta. Para obter uma lista de códigos e de descrições de erros, consulte a seção [Erros especiais](#) da página Conteúdo de objetos SELECT no Amazon Simple Storage Service API Reference.

### Tópicos

- [Recursos relacionados \(p. 256\)](#)
- [Selecionar conteúdo de objetos usando o SDK para Java \(p. 256\)](#)
- [Selecionar conteúdo de objetos usando a API REST \(p. 258\)](#)
- [Selecionar conteúdo de objetos usando outros SDKs \(p. 258\)](#)

## Recursos relacionados

- [Usar os AWS SDKs, a CLI e os Explorers \(p. 639\)](#)

## Selecionar conteúdo de objetos usando o SDK para Java

Use o Amazon S3 Select para selecionar conteúdo de um objeto com Java usando o método `selectObjectContent`. Se a ação for bem-sucedida, retornará os resultados da expressão SQL. O bucket e a chave de objeto especificados devem existir; caso contrário, ocorrerá um erro.

## Example Exemplo

O código Java a seguir retorna o valor da primeira coluna para cada registro armazenado em um objeto que contém dados armazenados em formato CSV. Ele também solicita que mensagens de Progress e Stats sejam retornadas. É necessário fornecer um nome de bucket válido e um objeto que contenha dados em formato CSV.

Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

```
package com.amazonaws;

import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.CSVInput;
import com.amazonaws.services.s3.model.CSVOutput;
import com.amazonaws.services.s3.model.CompressionType;
import com.amazonaws.services.s3.model.ExpressionType;
import com.amazonaws.services.s3.model.InputSerialization;
import com.amazonaws.services.s3.model.OutputSerialization;
import com.amazonaws.services.s3.model.SelectObjectContentEvent;
import com.amazonaws.services.s3.model.SelectObjectContentVisitor;
import com.amazonaws.services.s3.model.SelectObjectContentRequest;
import com.amazonaws.services.s3.model.SelectObjectContentResult;

import java.io.File;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.concurrent.atomic.AtomicBoolean;

import static com.amazonaws.util.IOUtils.copy;

/**
 * This example shows how to query data from S3Select and consume the response in the form
 * of an
 * InputStream of records and write it to a file.
 */

public class RecordInputStreamExample {

    private static final String BUCKET_NAME = "${my-s3-bucket}";
    private static final String CSV_OBJECT_KEY = "${my-csv-object-key}";
    private static final String S3_SELECT_RESULTS_PATH = "${my-s3-select-results-path}";
    private static final String QUERY = "select s._1 from S3Object s";

    public static void main(String[] args) throws Exception {
        final AmazonS3 s3Client = AmazonS3ClientBuilder.defaultClient();

        SelectObjectContentRequest request = generateBaseCSVRequest(BUCKET_NAME,
        CSV_OBJECT_KEY, QUERY);
        final AtomicBoolean isResultComplete = new AtomicBoolean(false);

        try (OutputStream fileOutputStream = new FileOutputStream(new File
        (S3_SELECT_RESULTS_PATH));
        SelectObjectContentResult result = s3Client.selectObjectContent(request)) {
            InputStream resultInputStream = result.getPayload().getRecordsInputStream(
            new SelectObjectContentEventVisitor() {
                @Override
                public void visit(SelectObjectContentEvent.StatsEvent event)
                {
                    System.out.println(
                        "Received Stats, Bytes Scanned: " +
                    event.getDetails().getBytesScanned()
                }
            })
        }
    }
}
```

```
+ " Bytes Processed: " +  
event.getDetails().getBytesProcessed());  
}  
  
/*  
 * An End Event informs that the request has finished successfully.  
 */  
@Override  
public void visit(SelectObjectContentEvent.EndEvent event)  
{  
    isResultComplete.set(true);  
    System.out.println("Received End Event. Result is complete.");  
}  
}  
);  
  
copy(resultInputStream, fileOutputStream);  
}  
  
/*  
 * The End Event indicates all matching records have been transmitted.  
 * If the End Event is not received, the results may be incomplete.  
 */  
if (!isResultComplete.get()) {  
    throw new Exception("S3 Select request was incomplete as End Event was not  
received.");  
}  
}  
  
private static SelectObjectContentRequest generateBaseCSVRequest(String bucket, String  
key, String query) {  
    SelectObjectContentRequest request = new SelectObjectContentRequest();  
    request.setBucketName(bucket);  
    request.setKey(key);  
    request.setExpression(query);  
    request.setExpressionType(ExpressionType.SQL);  
  
    InputSerialization inputSerialization = new InputSerialization();  
    inputSerialization.setCsv(new CSVInput());  
    inputSerialization.setCompressionType(CompressionType.NONE);  
    request.setInputSerialization(inputSerialization);  
  
    OutputSerialization outputSerialization = new OutputSerialization();  
    outputSerialization.setCsv(new CSVOutput());  
    request.setOutputSerialization(outputSerialization);  
  
    return request;  
}  
}
```

## Selecionar conteúdo de objetos usando a API REST

Você pode usar o SDK da AWS para selecionar conteúdo de objetos. Contudo, se o seu aplicativo exigir, você pode enviar solicitações REST diretamente. Para obter mais informações sobre o formato de solicitação e de resposta, consulte [Conteúdo de objetos do SELECT](#).

## Selecionar conteúdo de objetos usando outros SDKs

É possível selecionar o conteúdo de um objeto usando o Amazon S3 Select com outros SDKs. Para obter mais informações, consulte:

- Python: [Usar o AWS SDK for Python \(Boto\) \(p. 651\)](#).

## Restaurar objetos arquivados

Os objetos arquivados no Amazon S3 Glacier não estão acessíveis em tempo real. Você deve primeiro iniciar uma solicitação de restauração e, em seguida, esperar até que uma cópia temporária do objeto esteja disponível pela duração (número de dias) que você especificar na solicitação. O tempo necessário para que os trabalhos de restauração sejam concluídos depende da opção de recuperação que você especificar: **Standard**, **Expedited** ou **Bulk**. Seja notificado quando a restauração estiver concluída usando notificações de evento do Amazon S3. Para obter mais informações, consulte [Configurar notificações de evento do Amazon S3 \(p. 522\)](#).

Depois de receber uma cópia temporária do objeto restaurado, a classe de armazenamento do objeto continuará sendo GLACIER (uma solicitação GET ou HEAD retornará GLACIER como a classe de armazenamento). Ao restaurar um arquivo, você paga pelo arquivamento (taxa GLACIER) e pela cópia temporariamente restaurada (taxa de armazenamento de Reduced Redundancy Storage [RRS]). Para obter mais informações sobre definição de preços, consulte [Definição de preço do Amazon S3](#).

Para obter informações sobre como usar transições do ciclo de vida para mover objetos para a classe de armazenamento GLACIER, consulte [Transição para a classe de armazenamento GLACIER \(arquivamento de objeto\) \(p. 127\)](#).

Os tópicos a seguir fornecem mais informações.

### Tópicos

- [Opções de recuperação de arquivos \(p. 259\)](#)
- [Restaurar um objeto arquivado usando o console do Amazon S3 \(p. 260\)](#)
- [Restaurar um objeto arquivado usando o AWS SDK for Java \(p. 261\)](#)
- [Restaurar um objeto arquivado usando o AWS SDK para .NET \(p. 262\)](#)
- [Restaurar um objeto arquivado usando a API REST \(p. 263\)](#)

## Opções de recuperação de arquivos

Você pode especificar uma das seguintes opções ao restaurar um objeto arquivado:

- **Expedited** - As recuperações expressas permitem que você acesse rapidamente os seus dados quando forem necessárias solicitações urgentes ocasionais para um subconjunto de arquivos. Salvo para os maiores objetos arquivados (250 MB+), os dados acessados usando recuperações expressas ficam normalmente disponíveis 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 \(p. 260\)](#).
- **Standard** - As recuperações padrão permitem que você acesse alguns de seus objetos arquivados em algumas horas. As recuperações padrão normalmente são concluídas dentro de 3 a 5 horas. Esta é a opção padrão para solicitações de recuperação que não especificam a opção de recuperação.
- **Bulk** – As recuperações em massa são a opção de recuperação de menor custo do Glacier, permitindo recuperar grandes quantidades de dados, até mesmo petabytes, em um dia e com um custo baixo. As recuperações em massa normalmente são concluídas dentro de 5 a 12 horas.

Para fazer uma recuperação Expedited, Standard ou Bulk, defina o elemento de solicitação `Tier` na solicitação da API REST de [restauração de objeto POST](#) para a opção que você deseja, ou o equivalente na AWS CLI, ou nos AWS SDKs. Se você adquiriu a capacidade provisionada, todas as recuperações expressas serão automaticamente fornecidas por meio de sua capacidade provisionada.

É possível restaurar objeto arquivado de maneira programática ou usando o console do Amazon S3. O Amazon S3 processa apenas uma solicitação de restauração por vez, por objeto. Você pode usar o

console e a API do Amazon S3 para verificar o status de restauração e descobrir quando o Amazon S3 excluirá a cópia restaurada.

## Capacidade provisionada

A capacidade provisionada garante que sua capacidade de recuperação para recuperações expressas esteja disponível quando você precisar dela. Cada unidade de capacidade garante que pelo menos três recuperações expressas possam ser realizadas a cada cinco minutos e fornece até 150 MB/s de taxa de transferência de recuperação.

Você deve comprar a capacidade de recuperação provisionada se sua carga de trabalho exigir acesso altamente confiável e previsível a um subconjunto de seus dados em minutos. As recuperações expressas são aceitas sem capacidade provisionada, exceto para situações raras de demanda incomumente alta. No entanto, ao solicitar acesso a recuperações expressas em todas as circunstâncias, você deve comprar a capacidade de recuperação provisionada. Você pode comprar a capacidade provisionada usando o console do Amazon S3, o console do Amazon S3 Glacier, a API REST [Comprar capacidade provisionada](#), os AWS SDKs ou a AWS CLI. Para obter informações sobre a definição de preço da capacidade provisionada, consulte [Definição de preço do Amazon S3 Glacier](#).

## Atualizar a velocidade de uma restauração em andamento

Usando a atualização rápida de restauração do Amazon S3, você pode alterar a velocidade de restauração para uma mais rápida durante o processo. Uma atualização rápida de restauração substitui uma restauração em andamento com um nível mais rápido. Não é possível reduzir uma restauração em andamento.

Para atualizar a velocidade de uma restauração em andamento, faça outra solicitação de restauração para o mesmo objeto definindo um novo elemento de solicitação `Tier` na API REST [POST Object restore](#) ou o equivalente na AWS CLI ou nos SDKs da AWS. Ao emitir uma solicitação para atualizar o nível de restauração, você deve escolher um nível mais rápido do que o nível de restauração do andamento. Você não deve alterar outros parâmetros, como o elemento de solicitação `Days`.

Seja notificado da conclusão da restauração usando notificações de evento do Amazon S3. Para obter informações sobre a definição de preço da atualização rápida de restauração, consulte [Definição de preço do Glacier](#).

## Restaurar um objeto arquivado usando o console do Amazon S3

Você pode usar o console do Amazon S3 para restaurar uma cópia de um objeto que foi arquivado no Amazon S3 Glacier. Para obter instruções sobre como restaurar um arquivo usando o Console de gerenciamento da AWS, consulte [Como restaurar um objeto do S3 que foi arquivado no Amazon S3 Glacier?](#) no Guia do usuário do console do Amazon Simple Storage Service.

Observe que, ao restaurar um arquivo, você está pagando pelo arquivo e também pela cópia que restaurou temporariamente. Para obter mais informações sobre definição de preços, consulte [Definição de preços do Amazon S3](#).

O Amazon S3 restaura uma cópia temporária do objeto somente pela duração especificada. Depois disso, o Amazon S3 exclui a cópia restaurada do objeto. Você pode modificar o período de expiração de uma cópia restaurada, reeditando uma restauração e, nesse caso, o Amazon S3 atualiza o período de expiração em relação ao tempo atual.

O Amazon S3 calcula o período de expiração da cópia restaurada do objeto adicionando o número de dias especificado na solicitação de restauração ao tempo atual e arredondando o tempo resultante para a meia-noite UTC do próximo dia. Por exemplo, se um objeto foi criado em 15/10/2012 às 10h30 UTC e o período de restauração foi especificado como 3 dias, a cópia restaurada expira em 19/10/2012 à meia-noite UTC, quando o Amazon S3 exclui a cópia do objeto.

Você pode restaurar uma cópia de objeto por qualquer número de dias. Contudo você deve recuperar objetos somente pela duração necessária devido aos custos de armazenamento associados a uma cópia de objeto. Para obter informações sobre preços, consulte [Definição de preço do Amazon S3](#).

## Restaurar um objeto arquivado usando o AWS SDK for Java

### Example

O exemplo a seguir mostra como recuperar um objeto arquivado no Amazon S3 Glacier usando o AWS SDK for Java. O exemplo inicia uma solicitação de restauração para o objeto arquivado especificado e verifica seu status de restauração. Para obter mais informações sobre a restauração de objetos arquivados, consulte [Restaurar objetos arquivados \(p. 259\)](#). Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-
developer-guide/blob/master/LICENSE-SAMPLECODE.)

import java.io.IOException;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ObjectMetadata;
import com.amazonaws.services.s3.model.RestoreObjectRequest;

public class RestoreArchivedObject {

    public static void main(String[] args) throws IOException {
        String clientRegion = "**** Client region ****";
        String bucketName = "**** Bucket name ****";
        String keyName = "**** Object key ****";

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

            // Create and submit a request to restore an object from Glacier for two days.
            RestoreObjectRequest requestRestore = new RestoreObjectRequest(bucketName,
keyName, 2);
            s3Client.restoreObjectV2(requestRestore);

            // Check the restoration status of the object.
            ObjectMetadata response = s3Client.getObjectMetadata(bucketName, keyName);
            Boolean restoreFlag = response.getOngoingRestore();
            System.out.format("Restoration status: %s.\n",
                restoreFlag ? "in progress" : "not in progress (finished or failed)");
        }
        catch(AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        }
        catch(SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

}

## Restaurar um objeto arquivado usando o AWS SDK para .NET

### Example

O seguinte exemplo de C# inicia uma solicitação para restaurar um objeto arquivado por 2 dias. O Amazon S3 mantém o status de restauração nos metadados do objeto. Após iniciar a solicitação, o exemplo recupera os metadados do objeto e verifica o valor da propriedade `RestoreInProgress`. Para obter instruções sobre criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-
developer-guide/blob/master/LICENSE-SAMPLECODE.)

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

namespace Amazon.DocSamples.S3
{
    class RestoreArchivedObjectTest
    {
        private const string bucketName = "**** bucket name ****";
        private const string objectKey = "*** archived object key name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            RestoreObjectAsync(client, bucketName, objectKey).Wait();
        }

        static async Task RestoreObjectAsync(IAmazonS3 client, string bucketName, string
objectKey)
        {
            try
            {
                var restoreRequest = new RestoreObjectRequest
                {
                    BucketName = bucketName,
                    Key = objectKey,
                    Days = 2
                };
                RestoreObjectResponse response = await
client.RestoreObjectAsync(restoreRequest);

                // Check the status of the restoration.
                await CheckRestorationStatusAsync(client, bucketName, objectKey);
            }
            catch (AmazonS3Exception amazonS3Exception)
            {
                Console.WriteLine("An AmazonS3Exception was thrown. Exception: " +
amazonS3Exception.ToString());
            }
            catch (Exception e)
            {
                Console.WriteLine("Exception: " + e.ToString());
            }
        }
    }
}
```

```
        }

        static async Task CheckRestorationStatusAsync(IAmazonS3 client, string bucketName,
string objectKey)
{
    GetObjectMetadataRequest metadataRequest = new GetObjectMetadataRequest
    {
        BucketName = bucketName,
        Key = objectKey
    };
    GetObjectMetadataResponse response = await
client.GetObjectMetadataAsync(metadataRequest);
    Console.WriteLine("restoration status: {0}", response.RestoreInProgress ? "in-
progress" : "finished or failed");
}
}
```

## Restaurar um objeto arquivado usando a API REST

O Amazon S3 fornece uma API para que você inicie uma restauração de arquivos. Para obter mais informações, consulte [Restauração de objeto POST](#) no Amazon Simple Storage Service API Reference.

## Consultar objetos arquivados

Com o tipo de seleção [POST Object restore](#), execute as operações de filtro usando declarações simples em SQL diretamente em seus dados arquivados pelo Amazon S3 no Glacier. Ao inserir uma consulta SQL para um objeto arquivado, a seleção executa a consulta em vigor e grava os resultados de saída em um bucket do S3. Você pode executar consultas e análises personalizadas dos dados armazenados no Glacier sem precisar restaurar o objeto todo para o Amazon S3.

Ao executar consultas de seleção, o Glacier fornece três níveis de acesso a dados: expresso, padrão e em massa. Todos esses níveis fornecem diferentes tempos de acesso aos dados e custos, e você pode escolher qualquer um deles dependendo da rapidez com que deseja que os dados sejam disponibilizados. Para obter mais informações, consulte [Nível de acesso aos dados \(p. 265\)](#).

Você pode usar o tipo de seleção de restauração com os AWS SDKs, a API REST do Glacier e a AWS Command Line Interface (AWS CLI).

### Tópicos

- [Requisitos e limites da seleção \(p. 263\)](#)
- [Como consultar dados usando a seleção? \(p. 264\)](#)
- [Como tratar erros \(p. 265\)](#)
- [Nível de acesso aos dados \(p. 265\)](#)
- [Mais informações \(p. 266\)](#)

## Requisitos e limites da seleção

Estes são os requisitos para uso da seleção:

- Os objetos de arquivo consultados pela seleção devem ser formatados como valores separados por vírgulas (CSV) descompactados.
- Um bucket do S3 para saída. A conta da AWS usada para iniciar um trabalho do Glacier Select deve ter permissões para gravação no bucket do S3. O bucket do Amazon S3 deve estar na mesma região da AWS do bucket que contém o objeto arquivado que está sendo consultado.

- A conta da AWS solicitante deve ter permissões para realizar as ações `s3:RestoreObject` e `s3:GetObject`. Para obter mais informações sobre essas permissões, consulte [Permissões relacionadas a operações de sub-recurso de bucket \(p. 332\)](#).
- O arquivo não deve usar criptografia no lado do servidor com chaves fornecidas pelo cliente (SSE-C) ou criptografia no lado do cliente.

Os seguintes limites são aplicáveis ao usar a seleção:

- Não há limite para o número de registros que a seleção pode processar. Um registro de entrada ou de saída não deve exceder 1 MB. Caso contrário, a consulta reporta uma falha. Existe um limite de 1.048.576 colunas por registro.
- Não há limite para o tamanho do resultado final. No entanto, os resultados são divididos em várias partes.
- Uma expressão SQL limita-se a 128 KB.

## Como consultar dados usando a seleção?

Com a seleção, você pode usar comandos SQL para consultar objetos de arquivo do Glacier que estão criptografados em formato CSV descompactado. Com essa restrição, você pode executar operações de consulta simples nos dados baseados em textos no Glacier. Por exemplo, você pode procurar um ID ou um nome específico em um conjunto de arquivos de texto.

Para consultar os dados do Glacier, crie uma solicitação de seleção usando a operação [POST Object restore](#). Ao iniciar uma solicitação de seleção, insira a expressão SQL, o arquivo a ser consultado e o local em que serão armazenados os resultados.

O exemplo de expressão a seguir retorna todos os registros do objeto arquivado especificado em [POST Object restore](#).

```
SELECT * FROM object
```

O Glacier Select oferece suporte a um subconjunto da linguagem SQL ANSI. Ele é compatível com os filtros comuns de cláusulas SQL, como `SELECT`, `FROM` e `WHERE`. Não oferece suporte a `SUM`, `COUNT`, `GROUP BY`, `JOINS`, `DISTINCT`, `UNION`, `ORDER BY` e `LIMIT`. Para obter mais informações sobre suporte a SQL, consulte [Referência SQL de seleção do Amazon S3 e Glacier Select](#) no Guia do desenvolvedor do Amazon Simple Storage Service.

## Saída de seleção

Ao iniciar uma solicitação de seleção, defina um local de saída para os resultados de consulta da seleção. Esse local deve ser um bucket do Amazon S3 na mesma região da AWS do bucket que contém o objeto arquivado que está sendo consultado. A conta da AWS que inicia o trabalho deve ter permissões para gravação no bucket do S3.

Você pode especificar a classe de armazenamento e a criptografia do Amazon S3 nos objetos de saída armazenados no Amazon S3. A seleção oferece suporte às criptografias SSE-KMS e SSE-S3. Ela não oferece suporte às criptografias SSE-C e no lado do cliente. Para obter mais informações sobre classes de armazenamento e criptografia do Amazon S3, consulte [Classes de armazenamento \(p. 107\)](#) e [Proteção de dados usando criptografia no lado do servidor \(p. 410\)](#).

Os resultados do Glacier Select são armazenados no bucket do S3 por meio do prefixo fornecido no local de saída especificado em [POST Object restore](#). A partir dessas informações, a seleção cria um prefixo exclusivo em referência ao ID do trabalho. (Os prefixos são usados para agrupar os objetos do Amazon S3 iniciando os nomes de objeto com uma string em comum.) Nesse prefixo exclusivo, há dois novos prefixos

criados, `results` para resultados e `errors` para logs e erros. Após a conclusão do trabalho, é gravado um manifesto de resultado que contém a localização de todos os resultados.

Há também um arquivo de espaço reservado chamado `job.txt` gravado no local de saída. Esse arquivo é gravado, mas nunca é atualizado. O arquivo de espaço reservado é usado para:

- A validação da permissão para gravação e a maioria dos erros de sintaxe SQL de maneira síncrona.
- Fornecer uma saída estática sobre a solicitação de seleção que você pode facilmente referenciar sempre que quiser.

Vamos supor, por exemplo, que você inicie uma solicitação de seleção com o local de saída dos resultados especificado como `s3://example-bucket/my-prefix` e a resposta do trabalho retorne o ID do trabalho como `examplekne1209ualkdjh812elkassdu9012e`. Após a conclusão do trabalho de seleção, você pode visualizar os seguintes objetos do Amazon S3 no bucket:

```
s3://example-bucket/my-prefix/examplekne1209ualkdjh812elkassdu9012e/job.txt
s3://example-bucket/my-prefix/examplekne1209ualkdjh812elkassdu9012e/results/abc
s3://example-bucket/my-prefix/examplekne1209ualkdjh812elkassdu9012e/results/def
s3://example-bucket/my-prefix/examplekne1209ualkdjh812elkassdu9012e/results/ghi
s3://example-bucket/my-prefix/examplekne1209ualkdjh812elkassdu9012e/result_manifest.txt
```

Os resultados da consulta de seleção são divididos em várias partes. No exemplo, a seleção usa o prefixo especificado ao definir o local de saída e acrescenta o ID do trabalho e o prefixo `results`. Em seguida, ela grava os resultados em três partes, com `abc`, `def` e `ghi` no final dos nomes de objeto. O manifesto de resultados contém os três arquivos para permitir a recuperação de maneira programática. Se o trabalho reportar falha com qualquer tipo de erro, um arquivo ficará visível com o prefixo de erro e um arquivo `error_manifest.txt` será gerado.

A presença de um arquivo `result_manifest.txt` juntamente com a ausência de `error_manifest.txt` garante que a tarefa foi concluída com êxito. Não há nenhuma garantia quanto à maneira em que os resultados são ordenados.

#### Note

O tamanho do nome de um objeto do Amazon S3, também conhecido como chave, não pode ter mais do que 1.024 bytes. O Glacier Select reserva 128 bytes para prefixos. Além disso, o tamanho do caminho do local do Amazon S3 não pode maior do que 512 bytes. Uma solicitação superior a 512 bytes retornará uma exceção, e a solicitação não será aceita.

## Como tratar erros

A seleção envia notificações de dois tipos de erros. O primeiro conjunto de erros é enviado de maneira síncrona quando você envia a consulta em [POST Object restore](#). Esses erros são enviados como parte da resposta HTTP. Outro conjunto de erros pode ocorrer após a consulta ter sido aceita, mas eles ocorrem durante a execução da consulta. Nesse caso, os erros são gravados no local de saída especificado com o prefixo `errors`.

A seleção interrompe a execução da consulta após a detecção de um erro. Para executar a consulta com êxito, você deve resolver todos os erros. Verifique os logs para identificar quais registros causaram a falha.

Como as consultas são executadas paralelamente em vários nós de computação, os erros obtidos não estão em ordem sequencial. Por exemplo, se a consulta falhar com um erro na linha 6.234, isso não significa que todas as linhas anteriores a ela foram processadas com êxito. A próxima execução da consulta pode mostrar um erro em outra linha.

## Nível de acesso aos dados

Você pode especificar um dos níveis de acesso aos dados a seguir ao consultar um objeto arquivado:

- **Expedited** – Permite que você acesse rapidamente os dados quando um subconjunto de arquivos for solicitado com urgência. Exceto para os arquivos maiores (mais de 250 MB), os dados acessados usando recuperações Expedited costumam ser disponibilizados dentro de 1 a 5 minutos. Existem dois tipos de acesso de dados Expedited: sob demanda e provisionado. As solicitações sob demanda são semelhantes às instâncias sob demanda do EC2 e estão disponíveis na maior parte do tempo. As solicitações provisionadas estarão disponíveis garantidamente quando você precisar delas. Para obter mais informações, consulte [Capacidade provisionada \(p. 266\)](#).
- **Standard** – Permite acessar qualquer um dos objetos arquivados em algumas horas. As recuperações Standard normalmente são concluídas dentro de 3 a 5 horas. Esse é o nível padrão.
- **Bulk** – A opção de acesso aos dados de menor custo do Glacier, permitindo recuperar grandes quantidades de dados, até mesmo petabytes, em um dia. O acesso Bulk em geral termina em 5 a 12 horas.

Para fazer uma solicitação de Expedited, Standard ou Bulk, defina o elemento de solicitação Tier na solicitação da API REST de [POST Object restore](#) para a opção que você deseja, ou o equivalente na AWS CLI ou nos AWS SDKs. Para acesso Expedited, não há necessidade de definir se a recuperação expressa é sob demanda ou provisionada. Se você adquiriu a capacidade provisionada, todas as recuperações Expedited serão automaticamente fornecidas por meio de sua capacidade provisionada. Para obter informações sobre a definição de preço do nível, consulte [Definição de preço do Glacier](#).

## Capacidade provisionada

A capacidade provisionada garante que sua capacidade de recuperação para recuperações expressas estará disponível quando você precisar dela. Cada unidade de capacidade garante que pelo menos três recuperações expressas possam ser realizadas a cada cinco minutos e fornece até 150 MB/s de taxa de transferência de recuperação.

Você deve comprar a capacidade de recuperação provisionada se sua carga de trabalho exigir acesso altamente confiável e previsível a um subconjunto de seus dados em minutos. As recuperações Expedited são aceitas sem capacidade provisionada, exceto em situações raras de demanda inesperadamente alta. No entanto, ao solicitar acesso a recuperações Expedited em todas as circunstâncias, você deve comprar a capacidade de recuperação provisionada. Você pode comprar a capacidade provisionada usando o console do Amazon S3, o console do Glacier, a API REST [Comprar capacidade provisionada](#), os AWS SDKs ou a AWS CLI. Para obter informações sobre a definição de preço da capacidade provisionada, consulte [Definição de preços do Glacier](#).

## Mais informações

- [POST Object restore](#) no Amazon Simple Storage Service API Reference
- Referência SQL para o [Amazon S3 Select](#) e o [Glacier Select](#) no Guia do desenvolvedor do Amazon Simple Storage Service

# Análise do Amazon S3 – análise de classe de armazenamento

Usando a análise de classe de armazenamento do Amazon S3, você pode analisar padrões de acesso de armazenamento para ajudar a decidir quando fazer a transição dos dados certos para a classe de armazenamento certa. Esse novo recurso de análise do Amazon S3 observa padrões de acesso de dados para ajudar você a determinar quando fazer a transição do armazenamento STANDARD acessado menos frequentemente para a classe de armazenamento STANDARD\_IA (IA, para acesso raro). Para obter mais informações sobre classes de armazenamento, consulte [Classes de armazenamento \(p. 107\)](#).

Depois que a análise de classe de armazenamento observa os padrões incomuns de acesso a um conjunto filtrado de dados em um período, você pode usar os resultados da análise para ajudá-lo a melhorar suas políticas de ciclo de vida. Você pode configurar a análise de classe de armazenamento para analisar todos os objetos em um bucket. Se desejar, você pode configurar filtros para agrupar para objetos para análise por prefixo comum (ou seja, objetos que têm nomes que começam com uma string comum), por tags de objeto ou por prefixo e por tags. Você provavelmente achará que filtrar por grupos de objeto é a melhor maneira de aproveitar a análise de classe de armazenamento.

## Important

A análise da classe de armazenamento não fornece recomendações de transições para as classes de armazenamento ONEZONE\_IA ou GLACIER.

Você pode ter vários filtros vários de análise de classe de armazenamento por bucket, até 1.000, e receberá uma análise separada para cada filtro. As várias configurações de filtro permitem analisar grupos específicos de objetos para melhorar suas políticas de ciclo de vida que faz a transição de objetos para STANDARD\_IA.

A análise de classe de armazenamento mostra visualizações de uso de armazenamento no console do Amazon S3 que são atualizadas diariamente. Os dados de uso de armazenamento também podem ser exportados diariamente para um arquivo em um bucket do S3. Você pode abrir o arquivo de relatório de uso exportado em um aplicativo de planilha ou usá-lo com ferramentas de business intelligence de sua preferência, como o Amazon QuickSight.

## Tópicos

- [Como faço para configurar a análise de classe de armazenamento? \(p. 267\)](#)
- [Como faço para usar a análise de classe de armazenamento? \(p. 268\)](#)
- [Como posso exportar dados de análise de classe de armazenamento? \(p. 271\)](#)
- [APIs REST de análise do Amazon S3 \(p. 272\)](#)

## Como faço para configurar a análise de classe de armazenamento?

Para configurar a análise de classe de armazenamento, configure os dados de objeto que você deseja analisar. Você pode configurar a análise de classe de armazenamento para fazer o seguinte:

- Analisar o conteúdo completo de um bucket.  
Você receberá uma análise para todos os objetos no bucket.
- Analisar objetos agrupados por prefixo e por tags.

Você pode configurar filtros que agrupam objetos para análise por prefixo, por tags de objeto ou por uma combinação de prefixo e tags. Você recebe uma análise separada para cada filtro configurado. Você pode ter várias configurações de filtro por bucket, até 1.000.

- Exportar dados de análise.

Quando você configura a análise de classe de armazenamento para um bucket ou filtro, pode optar por exportar os dados de análise para um arquivo todo dia. A análise do dia é adicionada ao arquivo para formar um registro histórico de análise para o filtro configurado. O arquivo é atualizado diariamente no destino escolhido por você. Ao selecionar dados para exportar, especifique um bucket de destino e um prefixo de destino opcional onde o arquivo é gravado.

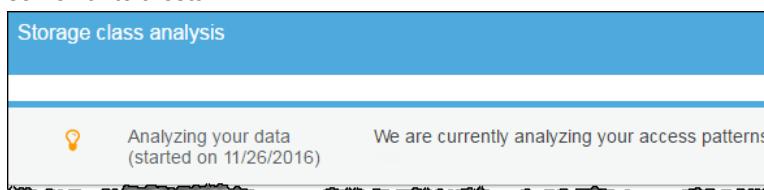
Você pode usar o console do Amazon S3, a API REST, a AWS CLI ou os SDKs da AWS para configurar a análise de classe de armazenamento.

- Para obter informações sobre como configurar a análise de classe de armazenamento no console do Amazon S3, consulte [Como faço para configurar a análise de classe de armazenamento?](#).
- Para usar a API do Amazon S3, use a API REST `PutBucketAnalyticsConfiguration`, ou equivalente, na AWS CLI ou nos SDKs da AWS.

## Como faço para usar a análise de classe de armazenamento?

Você usa a análise de classe de armazenamento para observar os padrões de acesso de dados ao longo do tempo e coletar informações para ajudar a melhorar o gerenciamento de ciclo de vida do armazenamento STANDARD\_IA. Depois de configurar um filtro, você começará a ver a análise de dados baseada no filtro no console do Amazon S3 em 24 a 48 horas. Contudo, a análise de classe de armazenamento observa os padrões de acesso de um conjunto de dados filtrado por 30 dias ou mais para coletar informações para análise antes de oferecer um resultado. A análise continua sendo executada após o resultado inicial e atualiza o resultado à medida que os padrões de acesso mudam.

Quando você configura um filtro pela primeira vez, o console do Amazon S3 mostra uma mensagem semelhante a esta.



A análise de classe de armazenamento observa os padrões de acesso de um conjunto de dados de objeto filtrado por 30 dias ou mais para coletar informações suficientes para a análise. Após a análise de classe de armazenamento coletar informações suficientes, você verá uma mensagem no console do Amazon S3 semelhante a esta.



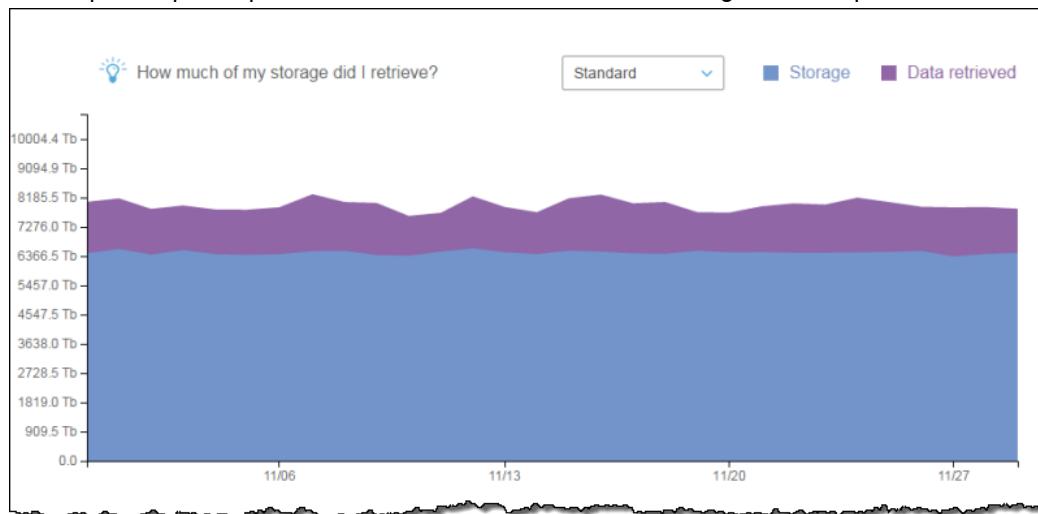
Quando a análise de classe de armazenamento é executada em busca de objetos acessados raramente, o conjunto filtrado de objetos agrupados com base na data de upload no Amazon S3 é observado. A análise

de classe de armazenamento determina se a faixa etária é acessada raramente observando os seguintes fatores do conjunto de dados filtrado:

- Objetos na classe de armazenamento STANDARD que têm mais de 128K.
- Quanto armazenamento total médio você tem por faixa etária.
- Número médio de bytes transferidos para fora (não frequência) por faixa etária.
- Os dados de exportação de análise incluem somente solicitações com dados pertinentes para a análise de classe de armazenamento. Isso pode causar diferenças no número de solicitações e nos bytes totais de upload e solicitação em comparação com o que é mostrado nas métricas de armazenamento ou rastreado por seus próprios sistemas internos.
- As solicitações GET e PUT com falha não são contadas para análise. Contudo, você verá as solicitações com falha nas métricas de armazenamento.

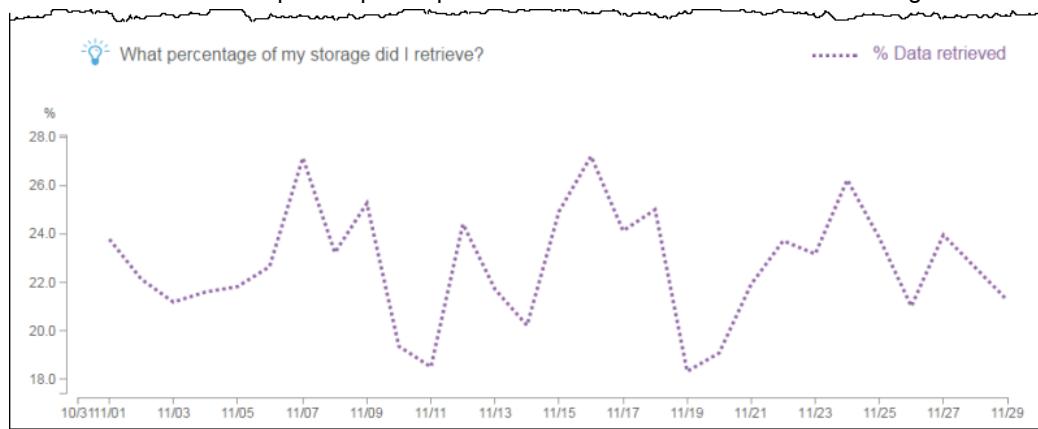
Quanto armazenamento eu recuperrei?

O console do Amazon S3 mostra em um gráfico quanto de armazenamento no conjunto de dados filtrado foi recuperado para o período de observe conforme exibido no seguinte exemplo.



Que porcentagem do armazenamento eu recuperrei?

O console do Amazon S3 também mostra em um gráfico que porcentagem do armazenamento no conjunto de dados filtrado foi recuperado para o período de observe conforme exibido no seguinte exemplo.



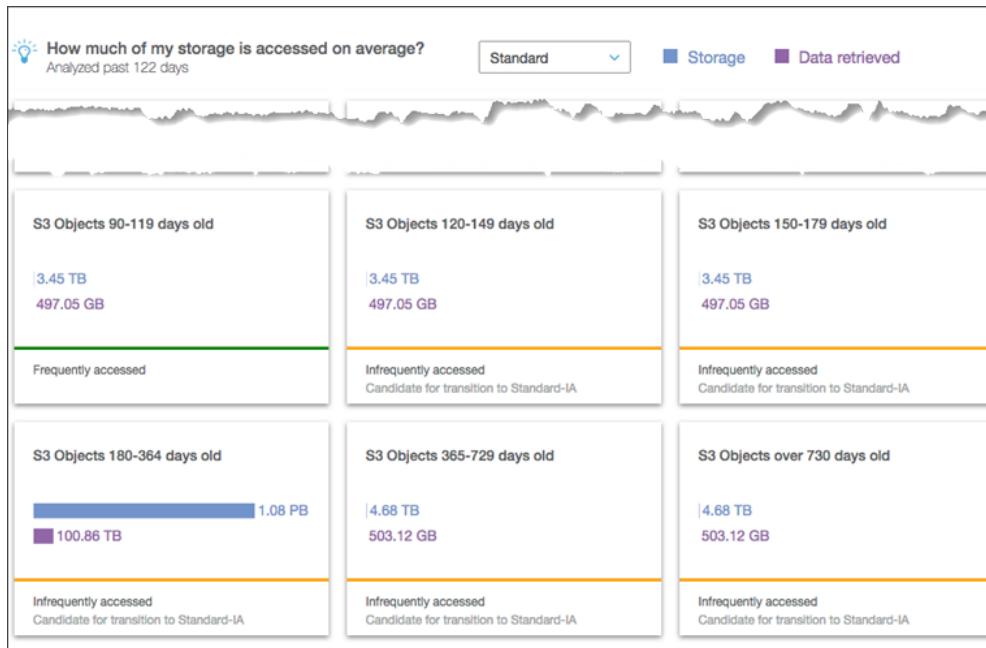
Como mencionado anteriormente neste tópico, quando a análise de classe de armazenamento é executada em busca de objetos acessados raramente, o conjunto filtrado de objetos agrupados com base na data de upload no Amazon S3 é observado. A análise de classe de armazenamento usa as seguintes faixas etárias de objeto predefinidas:

- Objetos do Amazon S3 com menos de 15 dias
- Objetos do Amazon S3 com 15 a 29 dias
- Objetos do Amazon S3 com 30 a 44 dias
- Objetos do Amazon S3 com 45 a 59 dias
- Objetos do Amazon S3 com 60 a 74 dias
- Objetos do Amazon S3 com 75 a 89 dias
- Objetos do Amazon S3 com 90 a 119 dias
- Objetos do Amazon S3 com 120 a 149 dias
- Objetos do Amazon S3 com 150 a 179 dias
- Objetos do Amazon S3 com 180 a 364 dias
- Objetos do Amazon S3 com 365 a 729 dias
- Objetos do Amazon S3 com 730 dias ou mais

Geralmente, leva cerca de 30 dias para observar os padrões de acesso e coletar informações suficientes para gerar um resultado de análise. Pode levar mais de 30 dias, dependendo do padrão de acesso exclusivo dos dados. No entanto, depois de configurar um filtro, você começará a ver a análise de dados baseada no filtro no console do Amazon S3 em 24 a 48 horas. Você pode visualizar a análise de acesso do objeto diariamente dividida por faixa etária no console do Amazon S3.

Quanto do armazenamento é acessado raramente?

O console do Amazon S3 mostra os padrões de acesso agrupados pelas faixas etárias de objeto predefinidas conforme exibido no seguinte exemplo.



O texto Frequently accessed (Acessado com frequência) ou Infrequently accessed (Acessado raramente) mostrado na parte inferior de cada faixa etária baseia-se na mesma lógica da recomendação de política

de ciclo de vida que está sendo preparada. Quando uma idade recomendada para uma política de ciclo de vida estiver pronta (RecommendedObjectAge), todos os níveis de idade mais novos que a idade recomendada serão marcados como acessados raramente, independentemente da proporção de acesso cumulativa atual. Esse texto destina-se a ser um auxílio visual para ajudar no processo de criação de ciclo de vida.

## Como posso exportar dados de análise de classe de armazenamento?

Você pode optar por exportar os relatórios de análise de classe de armazenamento para um arquivo sem formatação de valores separados por vírgula (CSV). Os relatórios são atualizados diariamente e se baseiam nos filtros de faixa etária de objeto que você configura. Ao usar o console do Amazon S3, você pode escolher a opção de exportação de relatório quando cria um filtro. Ao selecionar dados para exportar, especifique um bucket de destino e um prefixo de destino opcional onde o arquivo é gravado. Você pode exportar os dados para um bucket de destino em uma conta diferente. O bucket de destino deve estar na mesma região que o bucket que você configura para ser analisado.

Você deve criar uma política de bucket no bucket de destino para conceder permissões ao Amazon S3 para verificar se a conta da AWS é proprietária do bucket e para gravar objetos no bucket no local definido. Para ver um exemplo de política, consulte [Conceder permissões para o inventário do Amazon S3 e a análise do Amazon S3 \(p. 364\)](#).

Após configurar relatórios de análise de classe de armazenamento, você começará a receber o relatório exportado diariamente após 24 horas. Depois disso, o Amazon S3 continuará monitorando e fornecendo exportações diárias.

Você pode abrir o arquivo CSV em um aplicativo de planilha ou importar o arquivo para outros aplicativos, como o [Amazon QuickSight](#). Para obter informações sobre como usar os arquivos do Amazon S3 com o Amazon QuickSight, consulte [Criar um conjunto de dados usando arquivos do Amazon S3](#) no Guia do usuário do Amazon QuickSight.

Os dados no arquivo exportado são classificados por data na faixa etária de objeto conforme exibido nos exemplos a seguir. Se a classe de armazenamento é STANDARD, a linha também contém dados para as colunas `ObjectAgeForSIATransition` e `RecommendedObjectAgeForSIATransition`.

Date	ConfigId	Filter	StorageClass	ObjectAge	ObjectCount	DataUploaded_MB	Storage_MB	DataRetrieved_MB	GetRequestCount	CumulativeAccessRatio	ObjectAgeForSIATransition	RecommendedObjectAgeForSIATransition
11/26/17	SalesMaterial	SalesMaterial	STANDARD	000-014	39376.428	3610.516	100409232	106131482	17724.24	1.056989285		
11/25/17	SalesMaterial	SalesMaterial	STANDARD	000-014	37904.412	3411.772	90017538.84	100602072	18068.4	1.11758301		
11/24/17	SalesMaterial	SalesMaterial	STANDARD	000-014	39744.432	3478.02	87888239.2	108258204.0	18584.64	1.23226355		
11/23/17	SalesMaterial	SalesMaterial	STANDARD	000-014	40480.44	3544.268	91903507.12	111984477.9	18928.8	1.218500593		
11/22/17	SalesMaterial	SalesMaterial	STANDARD	000-014	40112.436	3544.268	94405093.4	105851751.9	17724.24	1.121250433		
11/21/17	SalesMaterial	SalesMaterial	STANDARD	000-014	40480.44	3345.524	103598371.7	115342424	18412.56	1.113361359		
11/20/17	SalesMaterial	SalesMaterial	STANDARD	000-014	39008.424	3411.772	92668826.16	114416707.1	18756.72	1.234683893		
11/19/17	SalesMaterial	SalesMaterial	STANDARD	000-014	40480.44	3444.896	90254194.28	116396152.9	18068.4	1.28964813		
11/18/17	SalesMaterial	SalesMaterial	STANDARD	000-014	39744.432	3444.896	88724282.88	110218385.6	17724.24	1.242257271		
11/17/17	SalesMaterial	SalesMaterial	STANDARD	000-014	37904.412	3444.896	89542277.32	102845839	18584.64	1.148419061		
11/16/17	SalesMaterial	SalesMaterial	STANDARD	000-014	40112.436	3444.896	91362283.92	116521794.3	18928.8	1.275381802		
11/15/17	SalesMaterial	SalesMaterial	STANDARD	000-014	40112.436	3378.648	87730610.56	113237336.7	18756.72	1.290739184		
11/14/17	SalesMaterial	SalesMaterial	STANDARD	000-014	39744.432	3478.02	96131832.8	110576562.8	17896.32	1.149739473		

11/25/17	SalesMaterial	SalesMaterial	STANDARD	015-029	56856.618	5117.658	135026308.3	150903108.1	27102.6	1.11758301	
11/24/17	SalesMaterial	SalesMaterial	STANDARD	015-029	59616.648	5213.07	131832358.8	162447307	27876.96	1.23226355	
11/23/17	SalesMaterial	SalesMaterial	STANDARD	015-029	60720.66	5316.402	137855260.7	167976716.9	28393.2	1.218500593	
11/22/17	SalesMaterial	SalesMaterial	STANDARD	015-029	60168.654	5316.402	141607640.1	158777627.8	26586.36	1.121250433	
11/21/17	SalesMaterial	SalesMaterial	STANDARD	015-029	60720.66	5018.286	155397557.6	173013635.9	27618.84	1.113361359	
11/20/17	SalesMaterial	SalesMaterial	STANDARD	015-029	58512.636	5117.658	139003239.2	171625060.6	28135.08	1.234683893	
11/19/17	SalesMaterial	SalesMaterial	STANDARD	015-029	60720.66	5167.344	135381291.4	174594229.3	27102.6	1.28964813	
11/18/17	SalesMaterial	SalesMaterial	STANDARD	015-029	59616.648	5167.344	133086424.3	165327578.3	26586.36	1.242257271	
11/17/17	SalesMaterial	SalesMaterial	STANDARD	015-029	56856.618	5167.344	134331416	154268758.5	27876.96	1.148419061	
11/16/17	SalesMaterial	SalesMaterial	STANDARD	015-029	60168.654	5167.344	137043425.9	174782691.5	28393.2	1.275381802	
11/15/17	SalesMaterial	SalesMaterial	STANDARD	015-029	60168.654	5067.972	131595915.8	169856005	28135.08	1.290739184	
11/14/17	SalesMaterial	SalesMaterial	STANDARD	015-029	59616.648	5217.03	144197749.2	165789844.1	26844.48	1.149739473	
11/13/17	SalesMaterial	SalesMaterial	STANDARD	015-029	59446.648	5440.344	132303383.3	156404070.8	27618.84	1.149739473	

No final do relatório, a faixa etária de objeto é ALL. As linhas ALL contêm totais cumulativos para todos os grupos de idade para o dia conforme exibido no exemplo a seguir.

11/25/17 SalesMaterial	SalesMaterial	STANDARD	ALL	777408.45	69729.16	3237567090	2066259214	363519	0.034 090-119	090-119
11/24/17 SalesMaterial	SalesMaterial	STANDARD	ALL	787528.56	69726.02	3222158279	2031465122	362228.4	0.034 090-119	090-119
11/23/17 SalesMaterial	SalesMaterial	STANDARD	ALL	775568.43	69643.21	3224228797	2143607314	365670	0.034 090-119	090-119
11/22/17 SalesMaterial	SalesMaterial	STANDARD	ALL	785688.54	70471.31	3268098456	2093598765	357926.4	0.034 090-119	090-119
11/21/17 SalesMaterial	SalesMaterial	STANDARD	ALL	772808.4	68069.82	3260462574	2152402776	366960.6	0.034 090-119	090-119
11/20/17 SalesMaterial	SalesMaterial	STANDARD	ALL	775568.43	69311.97	3235392872	2135278200	363088.8	0.034 090-119	090-119
11/19/17 SalesMaterial	SalesMaterial	STANDARD	ALL	793048.62	68566.68	3191517795	2156592855	361368	0.034 090-119	090-119
11/18/17 SalesMaterial	SalesMaterial	STANDARD	ALL	784768.53	69808.83	3241761620	2158880455	357926.4	0.034 090-119	090-119
11/17/17 SalesMaterial	SalesMaterial	STANDARD	ALL	782008.5	71050.98	3143115658	2130892653	372983.4	0.034 090-119	090-119
11/16/17 SalesMaterial	SalesMaterial	STANDARD	ALL	773728.41	70636.93	3237654725	2145622446	369111.6	0.034 090-119	090-119
11/15/17 SalesMaterial	SalesMaterial	STANDARD	ALL	773728.41	69477.59	3131717743	2091869264	369541.8	0.034 090-119	090-119
11/14/17 SalesMaterial	SalesMaterial	STANDARD	ALL	789368.58	70388.5	3169870122	2171336445	358355.6	0.034 090-119	090-119
11/13/17 SalesMaterial	SalesMaterial	STANDARD	ALL	767288.34	69643.21	3199668074	2128513319	367390.8	0.034 090-119	090-119
11/12/17 SalesMaterial	SalesMaterial	STANDARD	ALL	767288.34	68649.49	3135649104	2041228760	358355.6	0.034 090-119	090-119
11/11/17 SalesMaterial	SalesMaterial	STANDARD	ALL	774648.42	68897.92	3132710428	2154202987	356635.8	0.034 090-119	090-119
11/10/17 SalesMaterial	SalesMaterial	STANDARD	ALL	774648.42	69311.97	3181716041	2175053920	369541.8	0.034 090-119	090-119
11/9/17 SalesMaterial	SalesMaterial	STANDARD	ALL	771888.39	69643.21	3118609765	2197544222	364809.6	0.034 090-119	090-119
11/8/17 SalesMaterial	SalesMaterial	STANDARD	ALL	781088.49	70885.36	3136378996	2162511633	360507.6	0.034 090-119	090-119
11/7/17 SalesMaterial	SalesMaterial	STANDARD	ALL	783848.52	68649.49	3166312643	2019414755	355345.2	0.034 090-119	090-119
11/6/17 SalesMaterial	SalesMaterial	STANDARD	ALL	771888.39	69146.39	3087018554	2210161640	363519	0.034 090-119	090-119
11/5/17 SalesMaterial	SalesMaterial	STANDARD	ALL	770968.38	68235.44	3098188075	2177492429	360077.4	0.034 090-119	090-119
11/4/17 SalesMaterial	SalesMaterial	STANDARD	ALL	783848.52	69974.45	3115637244	2208645234	367390.8	0.034 090-119	090-119
11/3/17 SalesMaterial	SalesMaterial	STANDARD	ALL	770048.37	69311.97	3125766832	2195447440	354054.6	0.034 090-119	090-119
				772808	721808	3136641556	2131328083	365057.2	0.034 090-119	090-119

A próxima seção descreve as colunas usadas no relatório.

## Layout de arquivos exportados

A tabela a seguir descreve o layout do arquivo exportado.

# APIs REST de análise do Amazon S3

Veja a seguir as operações REST usadas para o inventário de armazenamento.

- [DELETE configuração de análise de bucket](#)
- [GET configuração de análise de bucket](#)
- [Listar configuração de análise de bucket](#)
- [PUT configuração de análise de bucket](#)

# Inventário do Amazon S3

O inventário do Amazon S3 é uma das ferramentas que o Amazon S3 fornece para ajudar a gerenciar seu armazenamento. Você pode usá-lo para auditar e gerar relatórios sobre o status da replicação e criptografia de seus objetos para os negócios, a conformidade e as necessidades normativas. Você também pode simplificar e acelerar os fluxos de trabalho de negócios e as tarefas de big data usando o inventário do Amazon S3, que fornece uma alternativa programada para a operação síncrona da API `List` do Amazon S3.

O inventário do Amazon S3 fornece arquivos de saída nos formatos CSV, [ORC](#) ou [Parquet](#) que listam seus objetos e os metadados correspondentes, diária ou semanalmente, para um bucket ou prefixo compartilhado do S3 (ou seja, objetos que tenham nomes que comecem com uma string comum). Para obter informações sobre a definição de preço de inventário do Amazon S3, consulte [Definição de preço do Amazon S3](#).

Você pode configurar várias listas de inventário para um bucket. Você pode configurar quais metadados de objeto serão incluídos no inventário, se deseja relacionar todas as versões do objeto ou apenas as versões atuais, onde armazenar o arquivo resultante da lista de inventários e se o inventário será gerado em uma frequência diária ou semanal. Você também pode especificar se o arquivo de lista de inventários será criptografado.

Você já pode consultar o inventário do Amazon S3 usando o SQL padrão com o [Amazon Athena](#), o Amazon Redshift Spectrum e outras ferramentas, como [Presto](#), [Apache Hive](#) e [Apache Spark](#). É fácil usar o Athena para executar consultas em seus arquivos de inventário. Use o Athena para consultas de inventário do Amazon S3 em todas as regiões onde o Athena está disponível.

## Tópicos

- [Como configurar o inventário do Amazon S3? \(p. 273\)](#)
- [O que está incluído no inventário do Amazon S3? \(p. 276\)](#)
- [Onde ficam as listas de inventários? \(p. 277\)](#)
- [Como saber quando um inventário está completo? \(p. 280\)](#)
- [Consultar o inventário com o Amazon Athena \(p. 280\)](#)
- [APIs REST de inventário do Amazon S3 \(p. 281\)](#)

## Como configurar o inventário do Amazon S3?

Esta seção descreve como configurar um inventário, incluindo detalhes sobre seus buckets de origem e destino.

### Buckets de origem e destino do inventário do Amazon S3

O bucket para o qual o inventário lista objetos é chamado de bucket de origem. O bucket em que o arquivo de lista de inventários é armazenado é chamado de bucket de destino.

#### Bucket de origem

O inventário lista os objetos armazenados no bucket de origem. Você pode obter as listas de inventário para um bucket inteiro ou filtradas pelo prefixo (nome de chave de objeto).

O bucket de origem:

- Contém os objetos que estão listados no inventário.
- Contém a configuração para o inventário.

Bucket de destino

Os arquivos de lista de inventários do Amazon S3 são gravados no bucket de destino. Você pode especificar um prefixo de destino (nome de chave de objeto) na configuração do inventário para agrupar todos os arquivos de lista de inventários em um local comum no bucket de destino.

O bucket de destino:

- Contém as listas de arquivos de inventário.
- Contém os arquivos manifestos que relacionam todas as listas de inventários de arquivo armazenadas no bucket de destino. Para obter mais informações, consulte [O que é um manifesto de inventário? \(p. 278\)](#)
- Deve haver uma política do bucket para dar permissão ao Amazon S3 para verificar a propriedade do bucket e permissão para gravar arquivos no bucket.
- Deve estar na mesma região da AWS que o bucket de origem.
- Pode ser igual ao bucket de origem.
- Pode pertencer a uma conta da AWS diferente da conta que tem a propriedade do bucket de origem.

## Configurar o inventário do Amazon S3

O inventário do Amazon S3 ajuda você a gerenciar seu armazenamento, criando listas dos objetos em um bucket do S3 com um agendamento definido. Você pode configurar várias listas de inventário para um bucket. As listas de inventários são publicadas em arquivos CSV, ORC ou Parquet em um bucket de destino.

A maneira mais fácil de configurar um inventário é usando o Console de gerenciamento da AWS, mas você também pode usar a API REST, a AWS CLI ou os AWS SDKs. O console realiza a primeira etapa do procedimento a seguir para você: adicionar uma política de bucket ao bucket de destino.

Configurar um inventário do Amazon S3 para um bucket do S3

1. Adicionar uma política do bucket para o bucket de destino.

Você deve criar uma política do bucket no bucket de destino para conceder permissões ao Amazon S3 para gravar objetos no bucket no local definido. Para ver um exemplo de política, consulte [Conceder permissões para o inventário do Amazon S3 e a análise do Amazon S3 \(p. 364\)](#).

2. Configurar um inventário para relacionar objetos em um bucket de origem e publicar a lista em um bucket de destino.

Ao configurar uma lista de inventários para um bucket de origem, você especifica o bucket de destino no qual deseja que a lista seja armazenada e se quer gerar a lista diária ou semanalmente. Você também pode configurar quais metadados de objeto serão incluídos e se serão relacionadas todas as versões dos objetos ou apenas as versões atuais.

Você pode especificar se o arquivo de lista de inventários deve ser criptografado usando chaves gerenciadas pelo Amazon S3 (SSE-S3) ou chaves gerenciadas pelo AWS KMS (SSE-KMS). Para obter

mais informações sobre SSE-S3 e SSE-KMS, consulte [Proteção de dados usando criptografia no lado do servidor \(p. 410\)](#). Se você pretende usar a criptografia SSE-KMS, consulte a Etapa 3.

- Para obter informações sobre como usar o console para configurar uma lista de inventários, consulte [Como configuro o inventário do Amazon S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.
  - Para usar a API do Amazon S3 para configurar uma lista de inventários, use a API REST [configuração de inventário de PUT Bucket](#) ou o equivalente da AWS CLI ou dos AWS SDKs.
3. Para criptografar o arquivo de lista de inventários com SSE-KMS, dê permissão para que o Amazon S3 use a chave do AWS KMS.

Você pode configurar a criptografia do arquivo de lista de inventários usando o Console de gerenciamento da AWS, a API REST, a AWS CLI ou os AWS SDKs. Independentemente do que você escolher, dê permissão para que o Amazon S3 use a chave mestra de cliente do AWS KMS (CMK) para criptografar o arquivo de inventário. Para conceder permissão ao Amazon S3, modifique a política de chaves da CMK do AWS KMS que está sendo usada para criptografar o arquivo de inventário. Para obter mais informações, consulte a próxima seção, [Conceder permissão para o Amazon S3 criptografar usando a chave do AWS KMS \(p. 275\)](#).

## Conceder permissão para o Amazon S3 criptografar usando a chave do AWS KMS

Conceda permissão para que o Amazon S3 criptografe usando sua chave do AWS KMS com uma política de chaves. O procedimento a seguir descreve como usar o console do AWS Identity and Access Management (IAM) para modificar a política de chaves da CMK do AWS KMS usada para criptografar o arquivo de inventário.

Para conceder permissões de criptografia usando a chave do AWS KMS

1. Faça login no Console de gerenciamento da AWS usando a conta da AWS que tem a CMK do AWS KMS e abra o console do IAM pelo <https://console.aws.amazon.com/iam/>.
2. No painel de navegação à esquerda, escolha Encryption keys (Chaves de criptografia).
3. Em Region (Região), escolha a região da AWS apropriada. Não use o seletor de regiões na barra de navegação (canto superior direito).
4. Escolha o alias da CMK com o qual você quer criptografar o inventário.
5. Na seção Key Policy (Política de chaves) da página, escolha Switch to policy view (Alternar para exibição de política).
6. Use o editor de Key Policy (Política de chaves), insira a política de chaves a seguir na política existente e, em seguida, escolha Save Changes (Salvar alterações). Talvez você queira copiar a política para o final da política existente.

```
{  
    "Sid": "Allow Amazon S3 use of the key",  
    "Effect": "Allow",  
    "Principal": {  
        "Service": "s3.amazonaws.com"  
    },  
    "Action": [  
        "kms:GenerateDataKey*"  
    ],  
    "Resource": "*"  
}
```

Você também pode usar a API de política da chave PUT do AWS KMS [PutKeyPolicy](#) para copiar a política de chaves para a CMK que está sendo usada para criptografar o arquivo de inventário. Para obter mais

informações sobre como criar e editar as CMKs do AWS KMS, consulte [Conceitos básicos](#) no AWS Key Management Service Developer Guide.

## O que está incluído no inventário do Amazon S3?

Um arquivo de lista de inventários contém uma lista dos objetos no bucket de origem e os metadados de cada objeto. As listas de inventários são armazenadas no bucket de destino como um arquivo CSV compactado com GZIP, como um arquivo colunar de linhas otimizado (ORC) do Apache compactado com ZLIB ou como um arquivo do Apache Parquet (Parquet) compactado com Snappy.

Uma lista de inventários contém uma lista dos objetos em um bucket do S3 e os seguintes metadados para cada objeto relacionado:

- Nome do bucket – o nome do bucket a que o inventário se refere.
- Nome da chave – nome da chave de objeto (ou chave) que identifica cada objeto no bucket. Ao usar o formato de arquivo CSV, o nome da chave é codificado em URL e deve ser decodificado para que possa ser usado.
- ID de versão – ID de versão de objetos. Quando você habilita o versionamento em um bucket, o Amazon S3 atribui um número de versão aos objetos adicionados ao bucket. Para obter mais informações, consulte [Versionamento de objeto \(p. 111\)](#). (O campo não está incluído se a lista for somente para a versão atual dos objetos.)
- IsLatest – definido como `True` se o objeto for a versão atual do objeto. (O campo não está incluído se a lista for somente para a versão atual dos objetos.)
- Tamanho – tamanho do objeto em bytes.
- Data da última modificação – data de criação do objeto ou data da última modificação, a mais atual entre as duas.
- ETag – a tag de entidade é um hash do objeto. O ETag reflete as alterações apenas no conteúdo de um objeto, não em seus metadados. A ETag pode ou não ser um resumo MD5 dos dados do objeto. Tudo depende de como o objeto foi criado e de como está criptografado.
- Classe de armazenamento – classe de armazenamento usada para armazenar o objeto. Para obter mais informações, consulte [Classes de armazenamento \(p. 107\)](#).
- Sinalizador de multipart upload – definido como `True` se o objeto foi carregado como um multipart upload. Para obter mais informações, consulte [Visão geral do multipart upload \(p. 181\)](#).
- Marcador de exclusão – definido como `True`, se o objeto for um marcador de exclusão. Para obter mais informações, consulte [Versionamento de objeto \(p. 111\)](#). (O campo não está incluído se a lista for somente para a versão atual dos objetos.)
- Status de replicação – definido para `PENDING`, `COMPLETED`, `FAILED` ou `REPLICA`. Para obter mais informações, consulte [Informação sobre o status da replicação entre regiões \(p. 585\)](#).
- Status da criptografia – definido como `SSE-S3`, `SSE-C`, `SSE-KMS` ou `NOT-SSE`. O status da criptografia no lado do servidor para SSE-S3, SSE-KMS e SSE com chaves fornecidas pelo cliente (SSE-C). Um status `NOT-SSE` significa que o objeto não foi criptografado com a criptografia no lado do servidor. Para obter mais informações, consulte [Proteção de dados usando criptografia \(p. 409\)](#).
- Manutenção do bloqueio do objeto até a data – A data até a qual o objeto bloqueado não pode ser excluído. Para obter mais informações, consulte [Introdução ao Amazon S3 Object Lock \(p. 470\)](#).
- Modo de bloqueio do objeto – Definido como `Governance` ou `Compliance` para objetos que sejam bloqueados. Para obter mais informações, consulte [Introdução ao Amazon S3 Object Lock \(p. 470\)](#).
- Status de retenção legal de bloqueio do objeto – Definido como `on` caso uma retenção legal tenha sido aplicada a um objeto; do contrário, ele é definido como `off`. Para obter mais informações, consulte [Introdução ao Amazon S3 Object Lock \(p. 470\)](#).

Veja a seguir um exemplo de lista de inventários em formato CSV aberta em um aplicativo de planilha. A linha de título aparece somente para ajudar a esclarecer o exemplo; ela não aparece na lista real.

Bucket	Key	VersionId	IsLatest	IsDeleteMarker	Size	LastModifiedDate	Etag	StorageClass	MultipartUploaded	ReplicationStatus
example-bucket	object1			FALSE	2.4E+08	2016-08-11T01:19	e80d8eda4	STANDARD	TRUE	
example-bucket	object2			FALSE	0	2016-08-10T22:23	d41d8cd98	STANDARD	FALSE	
example-bucket	object3			FALSE	9	2016-08-10T20:18	9090441e4	STANDARD_IA	FALSE	
example-bucket	object4			FALSE	9	2016-08-10T20:36	9090441e4	STANDARD_IA	FALSE	
example-bucket	object1			FALSE	22	2016-08-10T20:35	9090441e4	STANDARD	FALSE	
example-bucket	object1			FALSE	2016-08-10T20:34	9090441e4	REDUCED_RED	FALSE		
example-bucket	object1			FALSE	2016-08-10T21:13	9090441e4	GLACIER	FALSE		

Recomendamos que você crie uma política de ciclo de vida que exclua listas de inventário antigas. Para obter mais informações, consulte [Gerenciamento do ciclo de vida de objetos \(p. 122\)](#).

## Consistência de inventário

Todos os seus objetos podem não aparecer em cada lista de inventários. A lista de inventários fornece uma consistência eventual para PUTs dos objetos novos e também dos substituídos e excluídos. As listas de inventário são um snapshot de rolamento de itens do bucket, que são eventualmente consistentes (ou seja, a lista pode não incluir os objetos adicionados ou excluídos recentemente).

Para validar o estado do objeto antes de você realizar uma ação no objeto, recomendamos que faça uma solicitação `HEAD Object` da API REST para recuperar metadados do objeto ou verifique suas propriedades no console do Amazon S3. Você também pode verificar metadados do objeto com a AWS CLI ou os SDKs da AWS. Para obter mais informações, consulte o [objeto HEAD](#) no Amazon Simple Storage Service API Reference.

## Onde ficam as listas de inventários?

Quando uma lista de inventários é publicada, os arquivos manifestos são publicados no seguinte local no bucket de destino.

```
destination-prefix/source-bucket/config-ID/YYYY-MM-DDTHH-MMZ/manifest.json
destination-prefix/source-bucket/config-ID/YYYY-MM-DDTHH-MMZ/manifest.checksum
destination-prefix/source-bucket/config-ID/hive/dt=YYYY-MM-DD-HH-MM/symlink.txt
```

- **prefixo de destino** é o prefixo (nome de chave de objeto) definido na configuração de inventário, que pode ser usado para agrupar todos os arquivos de lista de inventários em um local comum no bucket de destino.
- **bucket de origem** é o bucket de origem ao qual a lista de inventários se refere. Ele é adicionado para evitar colisões quando vários relatórios do inventário de diferentes buckets de origem são enviados ao mesmo bucket de destino.
- O **ID de config** é adicionado para evitar colisões com vários relatórios do inventário do mesmo bucket de origem que são enviados ao mesmo bucket de destino.
- **YYYY-MM-DDTHH-MMZ** é o time stamp que consiste na hora de início e na data em que o relatório de inventário começa a fazer a varredura no bucket. Por exemplo, 2016-11-06T21-32Z. O armazenamento adicionado após o time stamp não está no relatório.
- **manifest.json** é o arquivo manifesto.
- **manifest.checksum** é o MD5 do conteúdo do arquivo **manifest.json**.
- **symlink.txt** é o arquivo manifesto compatível com o Apache Hive.

As listas de inventários são publicadas diária ou semanalmente no seguinte local do bucket de destino.

```
destination-prefix/source-bucket/config-ID/example-file-name.csv.gz
...
destination-prefix/source-bucket/config-ID/example-file-name-1.csv.gz
```

- **destination-prefix** é o prefixo (nome da chave de objeto) definido na configuração de inventário. Ele pode ser usado para agrupar todos os arquivos da lista de inventários em um local comum no bucket de destino.
- **bucket de origem** é o bucket de origem ao qual a lista de inventários se refere. Ele é adicionado para evitar colisões quando vários relatórios do inventário de diferentes buckets de origem são enviados ao mesmo bucket de destino.
- **example-file-name.csv.gz** é um dos arquivos de inventário em formato CSV. Os nomes de inventário ORC terminam com a extensão do nome do arquivo **.orc**, e os nomes de inventário do Parquet terminam com a extensão de nome de arquivo **.parquet**.

## O que é um manifesto de inventário?

Os arquivos manifestos **manifest.json** e **symlink.txt** descrevem onde os arquivos de inventário estão localizados. Sempre que uma nova lista de inventários é entregue, um novo conjunto de arquivos manifestos a acompanha.

Cada manifesto contido no arquivo **manifest.json** fornece metadados e outras informações básicas sobre um inventário. Essas informações incluem:

- Nome do bucket de origem
- Nome do bucket de destino
- Versão do inventário
- Time stamp de criação no formato de data de referência (epoch) que consiste na hora de início e na data em que o relatório de inventário começa a fazer a varredura no bucket
- Formato e esquema de arquivos de inventário
- Lista dos arquivos de inventário que estão no bucket de destino

Sempre que um arquivo **manifest.json** é gravado, ele é acompanhado por um arquivo **manifest.checksum**, que representa o MD5 do conteúdo do arquivo **manifest.json**.

Veja a seguir um exemplo de manifesto em um arquivo **manifest.json** para um inventário em formato CSV.

```
{
    "sourceBucket": "example-source-bucket",
    "destinationBucket": "arn:aws:s3:::example-inventory-destination-bucket",
    "version": "2016-11-30",
    "creationTimestamp" : "1514944800000",
    "fileFormat": "CSV",
    "fileSchema": "Bucket, Key, VersionId, IsLatest, IsDeleteMarker, Size,
LastModifiedDate, ETag, StorageClass, IsMultipartUploaded, ReplicationStatus,
EncryptionStatus, ObjectLockRetainUntilDate, ObjectLockMode, ObjectLockLegalHoldStatus",
    "files": [
        {
            "key": "Inventory/example-source-bucket/2016-11-06T21-32Z/
files/939c6d46-85a9-4ba8-87bd-9db705a579ce.csv.gz",
            "size": 2147483647,
            "MD5checksum": "f11166069f1990abeb9c97ace9cdfabc"
        }
    ]
}
```

```
}
```

Veja a seguir um exemplo de manifesto em um arquivo `manifest.json` para um inventário em formato ORC.

```
{
    "sourceBucket": "example-source-bucket",
    "destinationBucket": "arn:aws:s3:::example-destination-bucket",
    "version": "2016-11-30",
    "creationTimestamp" : "1514944800000",
    "fileFormat": "ORC",
    "fileSchema":
        "struct<bucket:string,key:string,version_id:string,is_latest:boolean,is_delete_marker:boolean,size:bigint> {
            "files": [
                {
                    "key": "inventory/example-source-bucket/data/d794c570-95bb-4271-9128-26023c8b4900.orc",
                    "size": 56291,
                    "MD5checksum": "5925f4e78e1695c2d020b9f6eexample"
                }
            ]
        }
}
```

Veja a seguir um exemplo de manifesto em um arquivo `manifest.json` para um inventário em formato Parquet.

```
{
    "sourceBucket": "example-source-bucket",
    "destinationBucket": "arn:aws:s3:::example-destination-bucket",
    "version": "2016-11-30",
    "creationTimestamp" : "1514944800000",
    "fileFormat": "Parquet",
    "fileSchema": "message s3.inventory { required binary bucket (UTF8); required binary key (UTF8); optional binary version_id (UTF8); optional boolean is_latest; optional boolean is_delete_marker; optional int64 size; optional int64 last_modified_date (TIMESTAMP_MILLIS); optional binary e_tag (UTF8); optional binary storage_class (UTF8); optional boolean is.multipart_uploaded; optional binary replication_status (UTF8); optional binary encryption_status (UTF8); }"
    "files": [
        {
            "key": "inventory/example-source-bucket/data/d754c470-85bb-4255-9218-47023c8b4910.parquet",
            "size": 56291,
            "MD5checksum": "5825f2e18e1695c2d030b9f6eexample"
        }
    ]
}
```

O `symlink.txt` é um arquivo manifesto compatível com o Apache Hive pelo qual o Hive detecta automaticamente arquivos de inventário e seus arquivos de dados associados. O manifesto compatível com o Hive funciona com os serviços compatíveis com o Hive, como Athena e Amazon Redshift Spectrum. Ele também funciona com aplicativos compatíveis com o Hive, como [Presto](#), [Apache Hive](#), [Apache Spark](#) e muitos outros.

#### Important

O arquivo manifesto compatível com Apache Hive `symlink.txt` atualmente não funciona com o AWS Glue.

A leitura do `symlink.txt` com o [Apache Hive](#) e o [Apache Spark](#) não é compatível com arquivos de inventário nos formatos ORC e Parquet.

## Como saber quando um inventário está completo?

Você pode configurar uma notificação de evento do Amazon S3 para receber aviso quando o arquivo de soma de verificação do manifesto for criado, o que indica que uma lista de inventários foi adicionada ao bucket de destino. O manifesto é uma lista atualizada de todas as listas de inventário no local de destino.

O Amazon S3 pode publicar eventos em um tópico do Amazon Simple Notification Service (Amazon SNS), uma fila do Amazon Simple Queue Service (Amazon SQS) ou uma função do AWS Lambda. Para obter mais informações, consulte [Configurar notificações de evento do Amazon S3 \(p. 522\)](#).

A configuração de notificação a seguir define que todos os arquivos `manifest.checksum` recém-adicionados ao bucket de destino são processados pela `cloud-function-list-write` do AWS Lambda.

```
<NotificationConfiguration>
  <QueueConfiguration>
    <Id>1</Id>
    <Filter>
      <S3Key>
        <FilterRule>
          <Name>prefix</Name>
          <Value>destination-prefix/source-bucket</Value>
        </FilterRule>
        <FilterRule>
          <Name>suffix</Name>
          <Value>checksum</Value>
        </FilterRule>
      </S3Key>
    </Filter>
    <Cloudcode>arn:aws:lambda:us-west-2:22223334444:cloud-function-list-write</Cloudcode>
    <Event>s3:ObjectCreated:*</Event>
  </QueueConfiguration>
</NotificationConfiguration>
```

Para obter mais informações, consulte [Usar políticas do AWS Lambda com o Amazon S3](#) no AWS Lambda Developer Guide.

## Consultar o inventário com o Amazon Athena

Você pode consultar um inventário do Amazon S3 usando o SQL padrão com o Amazon Athena em todas as regiões em que o Athena está disponível. Para verificar a disponibilidade de região da AWS, consulte a [Tabela de regiões da AWS](#).

O Athena consegue consultar arquivos de inventário do Amazon S3 nos formatos ORC, Parquet ou CSV. Quando você usar o Athena para consultar inventários, use arquivos de inventário no formato ORC ou Parquet. Os formatos ORC e Parquet têm um desempenho de consulta mais rápido e custos mais baixos de consulta. ORC e Parquet são formatos de arquivo colunares do tipo autodescritivo projetados para o [Apache Hadoop](#). O formato colunar permite que o leitor leia, descompacte e processe apenas as colunas necessárias para a consulta atual. Os formatos ORC e Parquet para inventário do Amazon S3 estão disponíveis em todas as regiões da AWS.

Para começar a usar Athena para consultar inventários do Amazon S3

1. Crie uma tabela do Athena. Para obter mais informações sobre a criação de tabelas, consulte [Criar tabelas no Amazon Athena](#) no Guia do usuário do Amazon Athena.

O exemplo de consulta a seguir inclui todos os campos opcionais no relatório de inventário no formato ORC. Descarte todos os campos opcionais que você não selecionou no inventário para que a consulta

corresponda aos campos escolhidos. Além disso, use o nome do seu bucket e a localização. A localização indica o caminho de destino do inventário; por exemplo, s3://destination-prefix/source-bucket/config-ID/hive/.

```
CREATE EXTERNAL TABLE your-table-name(  
    'bucket' string,  
    key string,  
    version_id string,  
    is_latest boolean,  
    is_delete_marker boolean,  
    size bigint,  
    last_modified_date timestamp,  
    e_tag string,  
    storage_class string,  
    is_multipart_uploaded boolean,  
    replication_status string,  
    encryption_status string,  
    object_lock_retain_until_date timestamp,  
    object_lock_mode string,  
    object_lock_legal_hold_status string  
)  
PARTITIONED BY (dt string)  
ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.orc.OrcSerde'  
STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.SymlinkTextInputFormat'  
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.IgnoreKeyTextOutputFormat'  
LOCATION 's3://destination-prefix/source-bucket/config-ID/hive/';
```

Ao usar Athena para consultar um relatório de inventário no formato Parquet, use a instrução Parquet SerDe a seguir em vez de ORC SerDe na instrução ROW FORMAT SERDE.

```
ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
```

2. Para adicionar novas listas de inventários à sua tabela, use o comando MSCK REPAIR TABLE a seguir.

```
MSCK REPAIR TABLE your-table-name;
```

3. Depois de executar as duas primeiras etapas, você pode executar consultas ad-hoc no inventário, como mostrado no exemplo a seguir.

```
SELECT encryption_status, count(*) FROM your-table-name GROUP BY encryption_status;
```

Para obter mais informações sobre o uso de Athena, consulte [Guia do usuário do Amazon Athena](#).

## APIs REST de inventário do Amazon S3

Veja a seguir as operações REST usadas para o inventário do Amazon S3.

- [Inventário do bucket DELETE](#)
- [Inventário do bucket GET](#)
- [Inventário do bucket de lista](#)
- [Inventário do bucket PUT](#)

# Gerenciamento de permissões de acesso aos recursos do Amazon S3

Por padrão, todos os recursos do Amazon S3 — como buckets, objetos e sub-recursos relacionados (por exemplo, configuração de lifecycle e configuração de website) — são privados: somente o proprietário do recurso (uma conta da AWS que o criou) pode acessá-lo. O proprietário do recurso pode conceder permissões de acesso a outros, criando uma política de acesso.

O Amazon S3 oferece opções de política de acesso classificadas amplamente como políticas com base em recursos e políticas de usuário. As políticas de acesso que você anexa aos recursos (buckets e objetos) são chamadas de políticas com base em recursos. Por exemplo, as políticas de bucket e as listas de controle de acesso (ACLs) são políticas com base em recursos. Você também pode anexar políticas de acesso a usuários em sua conta. Elas são chamadas de políticas de usuário. Você pode optar por usar políticas com base em recursos, políticas de usuário ou qualquer combinação delas para gerenciar permissões para seus recursos do Amazon S3. Os tópicos introdutórios fornecem diretrizes gerais para gerenciamento de permissões.

Recomendamos que você analise, primeiramente, os tópicos de visão geral de controle de acesso. Para obter mais informações, consulte [Introdução ao gerenciamento de permissões de acesso aos recursos do Amazon S3 \(p. 282\)](#). Depois, para obter mais informações sobre as opções específicas da política de acesso, consulte estes tópicos:

- [Uso de políticas de bucket e políticas de usuário \(p. 326\)](#)
- [Gerenciar o acesso com ACLs \(p. 390\)](#)
- [Usar o Amazon S3 Block Public Access \(p. 402\)](#)

## Introdução ao gerenciamento de permissões de acesso aos recursos do Amazon S3

### Tópicos

- [Visão geral do gerenciamento de acesso \(p. 283\)](#)
- [Como o Amazon S3 autoriza uma solicitação \(p. 288\)](#)
- [Diretrizes para usar as opções disponíveis de política de acesso \(p. 293\)](#)
- [Demonstrações com exemplo: gerenciar o acesso aos recursos do Amazon S3 \(p. 297\)](#)

Os tópicos nesta seção fornecem uma visão geral do gerenciamento de permissões de acesso aos recursos do Amazon S3 e fornecem as diretrizes sobre quando usar qual método de controle de acesso. O tópico também fornece demonstrações introdutórias de exemplo. Recomendamos que revise esses tópicos na ordem.

## Visão geral do gerenciamento de acesso

### Tópicos

- [Recursos da Amazon S3 \(p. 283\)](#)
- [Operações de recurso \(p. 284\)](#)
- [Gerenciamento de acesso a recursos \(opções de política de acesso\) \(p. 284\)](#)
- [Qual método de controle de acesso devo usar? \(p. 287\)](#)
- [Tópicos relacionados \(p. 287\)](#)

Ao conceder permissões, você decide quem as recebe, a quais recursos do Amazon S3 as permissões se referem e as ações específicas que deseja permitir nesses recursos.

## Recursos da Amazon S3

Os buckets e objetos são os principais recursos do Amazon S3 e ambos têm sub-recursos associados. Por exemplo, os sub-recursos de bucket incluem o seguinte:

- `lifecycle` – Armazena informações de configuração do ciclo de vida (consulte [Gerenciamento do ciclo de vida de objetos \(p. 122\)](#)).
- `website` – Armazena informações de configuração do site se você configurar seu bucket para hospedagem de sites (consulte [Hospedagem de um site estático no Amazon S3 \(p. 494\)](#)).
- `versioning` – Armazena a configuração de versionamento (consulte [Versionamento do bucket PUT](#)).
- `policy` e `acl` (lista de controle de acesso) – Armazena informações de permissão de acesso do bucket.
- `cors` (compartilhamento de recursos de origem cruzada) – Oferece suporte a configuração do bucket para permitir solicitações de origem cruzada (consulte [Cross-Origin Resource Sharing \(CORS, Compartilhamento de recursos de origem cruzada\) \(p. 156\)](#)).
- `logging` – Permite solicitar ao Amazon S3 para salvar logs de acesso de bucket.

Os sub-recursos de objeto incluem o seguinte:

- `acl` – Armazena uma lista de permissões de acesso no objeto. Este tópico discute como usar esse sub-recurso para gerenciar permissões de objeto (consulte [Gerenciar o acesso com ACLs \(p. 390\)](#)).
- `restore` – Oferece suporte para restaurar, temporariamente, um objeto arquivado (consulte [Restauração do objeto POST](#)). Um objeto na classe de armazenamento Glacier é um objeto arquivado. Para acessar o objeto, você deve, primeiro, iniciar uma solicitação de restauração, que restaura uma cópia do objeto arquivado. Na solicitação, especifique o número de dias que você deseja que a cópia restaurada exista. Para obter mais informações sobre arquivamento de objetos, consulte [Gerenciamento do ciclo de vida de objetos \(p. 122\)](#).

## Sobre o proprietário de recursos

Por padrão, todos os recursos do Amazon S3 são privados. Apenas um proprietário de recursos pode acessar o recurso. O proprietário do recurso refere-se à conta da AWS que criou o recurso. Por exemplo:

- A conta da AWS que você usa para criar buckets e objetos detém esses recursos.
- Se você criar um usuário do AWS Identity and Access Management (IAM) em sua conta da AWS, essa conta será o proprietário pai. Se o usuário do IAM fizer upload de um objeto, a conta pai, à qual o usuário pertence, deterá o objeto.
- Um proprietário do bucket pode conceder permissões entre contas à outra conta da AWS (ou aos usuários em outra conta) para fazer upload de objetos. Nesse caso, a conta da AWS que faz upload

dos objetos detém esses objetos. O proprietário do bucket não tem permissões nos objetos que outras contas possuem, com as seguintes exceções:

- O proprietário do bucket paga as faturas. O proprietário do bucket pode negar acesso a todos os objetos ou excluir objetos no bucket, independentemente de quem o possui.
- O proprietário do bucket pode arquivar todos os objetos ou restaurar os objetos arquivados, independentemente de quem os possui. Arquivo refere-se à classe de armazenamento usada para armazenar os objetos. Para obter mais informações, consulte [Gerenciamento do ciclo de vida de objetos \(p. 122\)](#).

#### Important

A AWS recomenda não usar credenciais raiz da conta da AWS para fazer solicitações. Em vez disso, crie um usuário do IAM e conceda acesso total a esse usuário. Esses usuários são conhecidos como usuários administradores. As credenciais do usuário administrador podem ser usadas em vez das credenciais raiz da conta, para interagir com a AWS e executar tarefas, tais como criar um bucket, criar usuários e conceder permissões a eles. Para obter mais informações, consulte [Credenciais de conta raiz vs. Credenciais de usuário do IAM](#) no AWS General Reference e [Melhores práticas do IAM](#) no Guia do usuário do IAM.

O diagrama a seguir mostra uma conta da AWS que possui recursos, os usuários do IAM, buckets e objetos.



## Operações de recurso

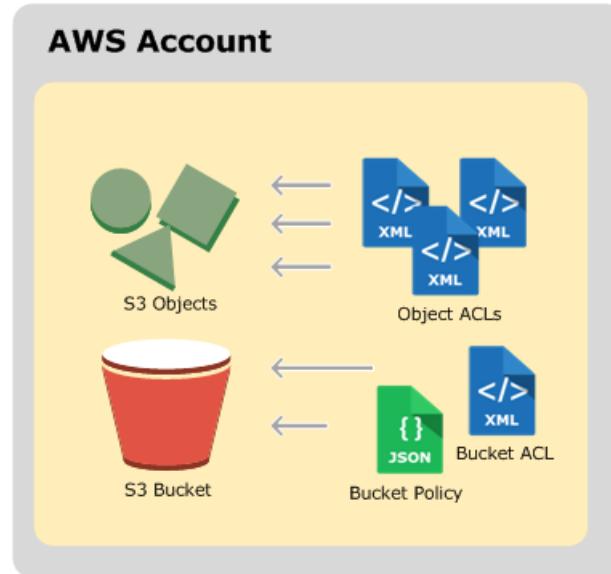
O Amazon S3 fornece um conjunto de operações para funcionar com recursos do Amazon S3. Para obter uma lista de operações disponíveis, acesse [Operações em buckets](#) e [Operações em objetos](#) no Amazon Simple Storage Service API Reference.

## Gerenciamento de acesso a recursos (opções de política de acesso)

Gerenciamento refere-se à concessão de permissões a outros (usuários e contas da AWS) para realizar operações de recurso, criando uma política de acesso. Por exemplo, você pode conceder a permissão `PUT Object` a um usuário em uma conta da AWS para que o usuário possa fazer upload de objetos em seu bucket. Além da concessão de permissões para usuários e contas individuais, você pode conceder permissões a todos (também chamado de acesso anônimo) ou a todos os usuários autenticados (usuários com credenciais da AWS). Por exemplo, se você configurar seu bucket como um site, talvez queira tornar os objetos públicos, concedendo a permissão `GET Object` a todos.

A política de acesso descreve quem tem acesso a quê. Você pode associar uma política de acesso a um recurso (bucket e objeto) ou um usuário. Conforme necessário, você pode classificar as políticas de acesso do Amazon S3 disponíveis da seguinte maneira:

- Políticas com base em recursos – As políticas de bucket e listas de controle de acesso (ACLs) usam como base recursos, porque você as anexa aos recursos do Amazon S3.



- ACL – Cada bucket e cada objeto têm uma ACL associada. Uma ACL é uma lista de concessões que identifica o concessionário e a permissão concedida. Você usa ACLs para conceder permissões de leitura/gravação básicas a outras contas da AWS. As ACLs usam um esquema XML específico do Amazon S3.

Esta é uma ACL de bucket de exemplo. A concessão na ACL mostra um proprietário do bucket com permissão de controle total.

```
<?xml version="1.0" encoding="UTF-8"?>
<AccessControlPolicy xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Owner>
    <ID>*** Owner-Canonical-User-ID ***</ID>
    <DisplayName>owner-display-name</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:type="Canonical User">
        <ID>*** Owner-Canonical-User-ID ***</ID>
        <DisplayName>display-name</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

As ACLs de bucket e objeto usam o mesmo esquema XML.

- Política de bucket – Para seu bucket, você pode adicionar uma política de bucket para conceder a outras contas da AWS ou usuários do IAM permissões ao bucket e aos objetos contidos nele. As permissões de objeto aplicam-se somente aos objetos criados pela proprietário do bucket. As políticas do bucket complementam e, em muitos casos, substituem as políticas de acesso com base em ACL.

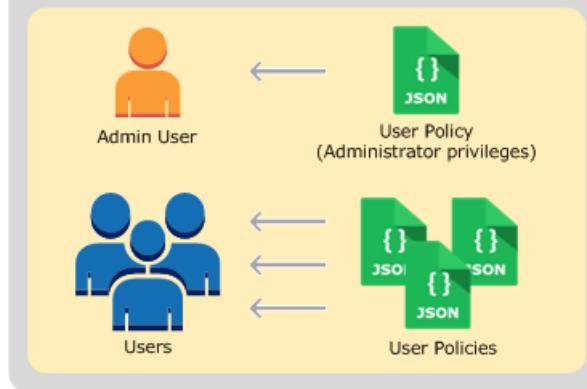
A seguir há um exemplo de política de bucket. Você expressa a política de bucket (e a política de usuário) usando um arquivo JSON. A política concede permissão de leitura anônima a todos os objetos em um bucket. A política de bucket tem uma instrução, que permite a ação s3:GetObject

(permissão de leitura) em objetos em um bucket chamado `examplebucket`. Especificando `principal` com um caractere curinga (\*), a política concede acesso anônimo.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": "*",  
            "Action": ["s3:GetObject"],  
            "Resource": ["arn:aws:s3:::examplebucket/*"]  
        }  
    ]  
}
```

- Políticas de usuário – Você pode usar o IAM para gerenciar o acesso a recursos do Amazon S3. Você pode criar usuários, grupos e funções do IAM em sua conta e anexar políticas de acesso que concedem acesso a recursos da AWS, incluindo o Amazon S3.

### AWS Account



Para obter mais informações sobre o IAM, acesse a página de detalhes do produto do [AWS Identity and Access Management \(IAM\)](#).

Veja a seguir um exemplo de política de usuário. Você não pode conceder permissões anônimas em uma política de usuário do IAM, pois a política é anexada a um usuário. A política de exemplo permite ao usuário associado que está anexado executar seis ações diferentes do Amazon S3 em um bucket nos objetos contidos nele. É possível anexar essa política a um usuário, grupo ou função específico do IAM.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "ExampleStatement1",  
            "Effect": "Allow",  
            "Action": [  
                "s3:PutObject",  
                "s3:GetObject",  
                "s3>ListBucket",  
                "s3>DeleteObject",  
                "s3:GetBucketLocation"  
            ],  
            "Resource": [  
                "arn:aws:s3:::examplebucket/*",  
                "arn:aws:s3:::examplebucket"  
            ]  
        },  
    ]  
}
```

```
{  
    "Sid": "ExampleStatement2",  
    "Effect": "Allow",  
    "Action": "s3>ListAllMyBuckets",  
    "Resource": "*"  
}  
]  
}
```

Quando o Amazon S3 recebe uma solicitação, deve avaliar todas as políticas de acesso para determinar se deve autorizar ou negar a solicitação. Para obter mais informações sobre como o Amazon S3 avalia essas políticas, consulte [Como o Amazon S3 autoriza uma solicitação \(p. 288\)](#).

## Qual método de controle de acesso devo usar?

Com as opções disponíveis para gravar uma política de acesso, surgem as seguintes perguntas:

- Quando devo usar qual método de controle de acesso? Por exemplo, para conceder permissões de bucket, devo usar uma política de bucket ou uma ACL de bucket? Possuo um bucket e os objetos no bucket. Devo usar uma política de acesso baseada em recursos ou uma política de usuário do IAM? Se eu usar uma política de acesso baseada em recursos, devo usar uma política de bucket ou uma ACL de objeto para gerenciar permissões de objeto?
- Possuo um bucket, mas não possuo todos os seus objetos. Como as permissões de acesso são gerenciadas para objetos que alguém possui?
- Se eu conceder acesso usando uma combinação dessas opções de política de acesso, como o Amazon S3 determinará se um usuário tem permissão para executar uma operação solicitada?

As seções a seguir explicam essas alternativas de controle de acesso, como o Amazon S3 avalia mecanismos de controle de acesso e quando usar cada método de controle de acesso. Também fornecem demonstrações com exemplos.

[Como o Amazon S3 autoriza uma solicitação \(p. 288\)](#)

[Diretrizes para usar as opções disponíveis de política de acesso \(p. 293\)](#)

[Demonstrações com exemplo: gerenciar o acesso aos recursos do Amazon S3 \(p. 297\)](#)

## Tópicos relacionados

Recomendamos que você analise, primeiramente, os tópicos introdutórios que explicam as opções disponíveis para gerenciar o acesso aos recursos do Amazon S3. Para obter mais informações, consulte [Introdução ao gerenciamento de permissões de acesso aos recursos do Amazon S3 \(p. 282\)](#). Você pode usar os seguintes tópicos para obter mais informações sobre as opções específicas da política de acesso.

- [Uso de políticas de bucket e políticas de usuário \(p. 326\)](#)
- [Gerenciar o acesso com ACLs \(p. 390\)](#)

## Como o Amazon S3 autoriza uma solicitação

### Tópicos

- [Tópicos relacionados \(p. 289\)](#)
- [Como o Amazon S3 autoriza uma solicitação para uma operação de bucket \(p. 289\)](#)
- [Como o Amazon S3 autoriza uma solicitação para uma operação de objeto \(p. 292\)](#)

Quando o Amazon S3 recebe uma solicitação — por exemplo, uma operação de bucket ou de objeto — ele primeiro verifica se o solicitante tem as permissões necessárias. O Amazon S3 avalia todas as políticas de acesso relevantes, as políticas de usuário e as políticas baseadas em recursos (política de bucket, ACL de bucket, ACL de objeto) para decidir se autoriza a solicitação. Os seguintes são alguns dos cenários de exemplo:

- Se o solicitante for um usuário do IAM, o Amazon S3 deve determinar se a conta da pai AWS à qual o usuário pertence concedeu ao usuário a permissão para executar a operação. Além disso, se a solicitação for para uma operação de bucket, como uma solicitação para listar o conteúdo do bucket, o Amazon S3 deverá verificar se o proprietário do bucket concedeu permissão para o solicitante executar a operação.

### Note

Para executar uma operação específica em um recurso, um usuário do IAM precisa de permissão da conta pai da AWS à qual pertence e da conta da AWS que possui o recurso.

- Se a solicitação for para uma operação em um objeto que o proprietário do bucket não possui, além de verificar se o solicitante tem permissões do proprietário do objeto, o Amazon S3 também deverá verificar a política do bucket para garantir que o proprietário do bucket não definiu negação explícita no objeto.

### Note

Um proprietário do bucket (que paga a fatura) pode negar explicitamente o acesso aos objetos do bucket, independentemente de quem os possui. O proprietário do bucket também pode excluir qualquer objeto do bucket.

Para determinar se o solicitante tem permissão para executar a operação específica, o Amazon S3 faz o seguinte, na ordem, quando recebe uma solicitação:

1. Converte todas as políticas de acesso relevantes (política de usuário, política de bucket, ACLs) em tempo de execução em um conjunto de políticas para avaliação.
2. Avalia o conjunto de políticas resultante nas seguintes etapas. Em cada etapa, o Amazon S3 avalia um subconjunto de políticas em um contexto específico, com base na autoridade contextual.
  - a. Contexto de usuário – no contexto de usuário, a conta pai à qual o usuário pertence é a autoridade contextual.

O Amazon S3 avalia um subconjunto de políticas de propriedade da conta pai. Esse subconjunto inclui a política de usuário que o pai anexa ao usuário. Se o pai também possuir o recurso na solicitação (bucket, objeto), o Amazon S3 também avaliará as políticas de recursos correspondentes (política de bucket, ACL de bucket e ACL de objeto) ao mesmo tempo.

Um usuário deve ter permissão da conta pai para executar uma operação.

A etapa se aplicará apenas se a solicitação for feita por um usuário em uma conta da AWS. Se a solicitação for feita usando credenciais raiz de uma conta da AWS, o Amazon S3 ignorará esta etapa.

- b. Contexto de bucket – no contexto de bucket, o Amazon S3 avalia as políticas de propriedade da conta da AWS que possui o bucket.

Se a solicitação for para uma operação de bucket, o solicitante deverá ter permissão do proprietário do bucket. Se a solicitação for para um objeto, o Amazon S3 avaliará todas as políticas de propriedade do proprietário do bucket para verificar se o proprietário do bucket não negou explicitamente o acesso ao objeto. Se houver uma negação explícita definida, o Amazon S3 não autorizará a solicitação.

- c. Contexto de objeto – se a solicitação for para um objeto, o Amazon S3 avaliará o subconjunto de políticas de propriedade do proprietário do objeto.

As seções a seguir descrevem em detalhes e fornecem exemplos:

- [Como o Amazon S3 autoriza uma solicitação para uma operação de bucket \(p. 289\)](#)
- [Como o Amazon S3 autoriza uma solicitação para uma operação de objeto \(p. 292\)](#)

## Tópicos relacionados

Recomendamos que primeiro você analise os tópicos introdutórios que explicam as opções para gerenciar o acesso aos recursos do Amazon S3. Para obter mais informações, consulte [Introdução ao gerenciamento de permissões de acesso aos recursos do Amazon S3 \(p. 282\)](#). Você pode usar os seguintes tópicos para obter mais informações sobre as opções específicas da política de acesso.

- [Uso de políticas de bucket e políticas de usuário \(p. 326\)](#)
- [Gerenciar o acesso com ACLs \(p. 390\)](#)

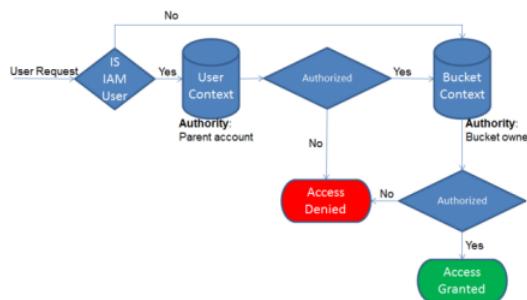
## Como o Amazon S3 autoriza uma solicitação para uma operação de bucket

Quando o Amazon S3 recebe uma solicitação para uma operação de bucket, o Amazon S3 converte todas as permissões relevantes — permissões de baseadas em recursos (política de bucket, lista de controle de acesso (ACL) de bucket) e política de usuário do IAM se a solicitação for de um usuário — em um conjunto de políticas a serem avaliadas no tempo de execução. Em seguida, ele avalia o conjunto de políticas resultante em uma série de etapas, de acordo com o contexto específico — contexto de usuário ou contexto de bucket.

1. Contexto de usuário – se o solicitante for um usuário do IAM, o usuário deverá ter permissão da conta pai da AWS à qual pertence. Nesta etapa, o Amazon S3 avalia um subconjunto de políticas de propriedade da conta pai (também conhecida como a autoridade de contexto). Esse subconjunto de políticas inclui a política de usuário que a conta pai anexa ao usuário. Se o pai também possuir o recurso na solicitação (neste caso, o bucket), o Amazon S3 também avaliará as políticas dos recursos correspondentes (a política do bucket e a ACL do bucket) ao mesmo tempo. Sempre que uma solicitação para uma operação de bucket é feita, os logs de acesso ao servidor registram o ID canônico do usuário do solicitante. Para obter mais informações, consulte [Registro em log de acesso ao servidor Amazon S3 \(p. 625\)](#).
2. Contexto de bucket – o solicitante deve ter permissões do proprietário do bucket para executar uma operação específica de bucket. Nesta etapa, o Amazon S3 avalia um subconjunto de políticas de propriedade da conta da AWS que possui o bucket.

O proprietário do bucket pode conceder permissão usando uma política do bucket ou a ACL do bucket. Observe que, se a conta da AWS que possui o bucket também for a conta pai de um usuário do IAM, ela poderá configurar permissões de bucket em uma política de usuário.

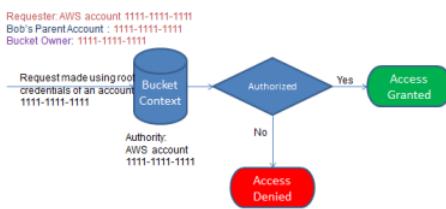
O seguinte é uma ilustração gráfica da avaliação baseada em contexto para a operação de bucket.



Os exemplos a seguir ilustram a lógica da avaliação.

### Exemplo 1: operação de bucket solicitada pelo proprietário do bucket

Neste exemplo, o proprietário do bucket envia uma solicitação para uma operação de bucket usando as credenciais raiz da conta da AWS.

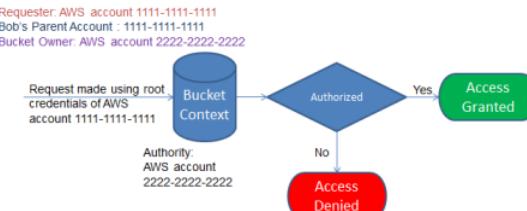


O Amazon S3 executa a avaliação de contexto da seguinte forma:

1. Como a solicitação é feita usando credenciais raiz de uma conta da AWS, o contexto de usuário não é avaliado.
2. No contexto de bucket, o Amazon S3 analisa a política de bucket para determinar se o solicitante tem permissão para executar a operação. O Amazon S3 autoriza a solicitação.

### Exemplo 2: operação de bucket solicitada por uma conta da AWS que não é a proprietária do bucket

Neste exemplo, uma solicitação é feita usando as credenciais raiz da conta da AWS 1111-1111-1111 para uma operação de bucket de propriedade da conta da AWS 2222-2222-2222. Nenhum usuário do IAM está envolvido nessa solicitação.



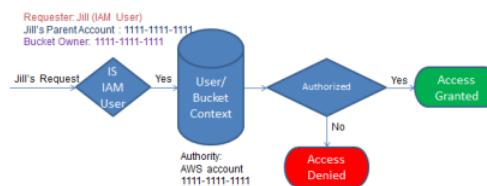
Neste caso, o Amazon S3 avalia o contexto da seguinte forma:

1. Como a solicitação é feita usando as credenciais raiz de uma conta da AWS, o contexto de usuário não é avaliado.
2. No contexto de bucket, o Amazon S3 examina a política de bucket. Se o proprietário do bucket (conta da AWS 2222-2222-2222) não autorizou a conta da AWS 1111-1111-1111 a executar a operação

solicitada, o Amazon S3 negará a solicitação. Caso contrário, o Amazon S3 concederá a solicitação e executará a operação.

### Exemplo 3: operação de bucket solicitada por um usuário do IAM cuja conta pai da AWS também é a proprietária do bucket

No exemplo, a solicitação é enviada por Jill, uma usuária do IAM na conta da AWS 1111-1111-1111, que também possui o bucket.



O Amazon S3 executa a seguinte avaliação de contexto:

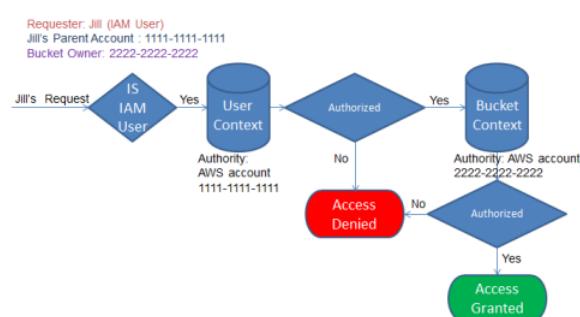
1. Como a solicitação é de uma usuária do IAM, no contexto de usuário, o Amazon S3 avalia todas as políticas que pertencem à conta pai da AWS para determinar se Jill tem permissão para executar a operação.

Neste exemplo, a conta pai da AWS 1111-1111-1111, à qual a usuária pertence, também é a proprietária do bucket. Como resultado, além da política do usuário, o Amazon S3 também avalia a política do bucket e a ACL do bucket no mesmo contexto, porque pertencem à mesma conta.

2. Como o Amazon S3 avaliou a política do bucket e a ACL do bucket como parte do contexto de usuário, ele não avalia o contexto de bucket.

### Exemplo 4: operação de bucket solicitada por uma usuária do IAM cuja conta pai da AWS não é a proprietária do bucket

Neste exemplo, a solicitação é enviada por Jill, uma usuária do IAM cuja conta pai da AWS é 1111-1111-1111, mas o bucket é de propriedade da outra conta da AWS, 2222-2222-2222.



Jill precisará de permissões da conta pai da AWS e do proprietário do bucket. O Amazon S3 avalia o contexto da seguinte forma:

1. Como a solicitação é de uma usuária do IAM, o Amazon S3 avalia o contexto de usuário analisando as políticas criadas pela conta para verificar se Jill tem as permissões necessárias. Se Jill tiver permissão, o Amazon S3 avaliará o contexto de bucket. Caso contrário, ele negará a solicitação.
2. No contexto de bucket, o Amazon S3 verifica se o proprietário do bucket 2222-2222-2222 concedeu a Jill (ou à conta pai da AWS) permissão para executar a operação solicitada. Se ela tiver permissão, o

Amazon S3 concederá a solicitação e executará a operação. Caso contrário, o Amazon S3 negará a solicitação.

## Como o Amazon S3 autoriza uma solicitação para uma operação de objeto

Quando o Amazon S3 recebe uma solicitação para uma operação de objeto, ele converte todas as permissões relevantes — permissões de baseadas em recursos (lista de controle de acesso (ACL) de objeto, política de bucket, ACL do bucket) e políticas de usuário do IAM se a solicitação for de um usuário — em um conjunto de políticas a serem avaliadas no tempo de execução. Em seguida, ele avalia o conjunto de políticas resultante em uma série de etapas. Em cada etapa, avalia um subconjunto de políticas em três contextos específicos — contexto de usuário, contexto de bucket e contexto de objeto.

1. Contexto de usuário – se o solicitante for um usuário do IAM, o usuário deverá ter permissão da conta pai da AWS à qual pertence. Nesta etapa, o Amazon S3 avalia um subconjunto de políticas de propriedade da conta pai (também conhecida como a autoridade de contexto). Esse subconjunto de políticas inclui a política de usuário que o pai anexa ao usuário. Se o pai também possuir o recurso na solicitação (bucket, objeto), o Amazon S3 avaliará as políticas de recursos correspondentes (política de bucket, ACL de bucket e ACL de objeto) ao mesmo tempo.

### Note

Se a conta pai da AWS possuir o recurso (bucket ou objeto), ela poderá conceder permissões de recursos a seu usuário do IAM usando a política de usuário ou a política de recurso.

2. Contexto de bucket – neste contexto, o Amazon S3 avalia as políticas de propriedade da conta da AWS que possui o bucket.

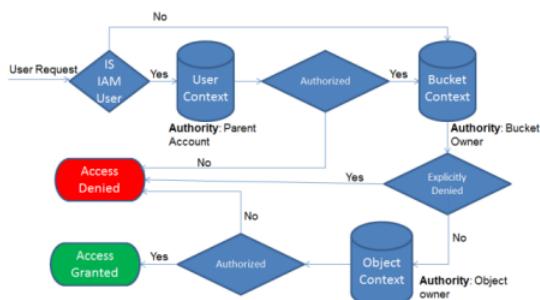
Se a conta da AWS que possui o objeto na solicitação não for a mesma que a do proprietário do bucket, no contexto do bucket, o Amazon S3 verificará as políticas se o proprietário do bucket tiver negado explicitamente o acesso ao objeto. Se houver uma negação explícita definida no objeto, o Amazon S3 não autorizará a solicitação.

3. Contexto de objeto – o solicitante deve ter permissões do proprietário do objeto para executar uma operação específica de objeto. Nesta etapa, o Amazon S3 avalia a ACL do objeto.

### Note

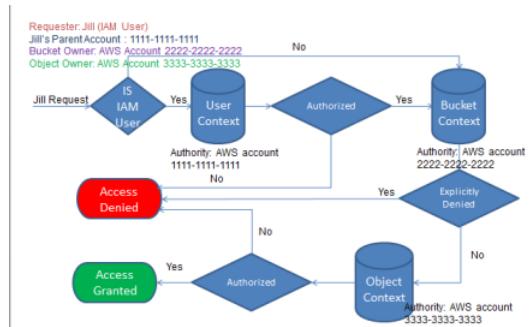
Se os proprietários do bucket e do objeto forem os mesmos, o acesso ao objeto poderá ser concedido na política de bucket, que é avaliada no contexto de bucket. Se os proprietários forem diferentes, os proprietários do objeto deverão usar uma ACL do objeto para conceder permissões. Se a conta da AWS que possui o objeto também for a conta pai à qual o usuário do IAM pertence, ela poderá configurar permissões de objeto nas políticas de usuário, que é avaliada no contexto de usuário. Para obter mais informações sobre como usar essas alternativas de política de acesso, consulte [Diretrizes para usar as opções disponíveis de política de acesso \(p. 293\)](#).

O seguinte é uma ilustração gráfica da avaliação baseada em contexto para uma operação de objeto.



## Exemplo 1: solicitação de operação de objeto

Neste exemplo, a usuária do IAM, Jill, cuja conta pai da AWS é 1111-1111-1111, envia uma solicitação de operação de objeto (por exemplo, Get object) para um objeto de propriedade da conta da AWS 3333-3333-3333 em um bucket de propriedade da conta da AWS 2222-2222-2222.



Jill precisará de permissão da conta pai da AWS, do proprietário do bucket e do proprietário do objeto. O Amazon S3 avalia o contexto da seguinte forma:

1. Como a solicitação é de uma usuária do IAM, o Amazon S3 avalia o contexto de usuário para verificar se a conta pai da AWS 1111-1111-1111 forneceu a Jill permissão para executar a operação solicitada. Se ela tiver essa permissão, o Amazon S3 avaliará o contexto de bucket. Caso contrário, o Amazon S3 negará a solicitação.
2. No contexto de bucket, o proprietário do bucket, conta da AWS 2222-2222-2222, é a autoridade de contexto. O Amazon S3 avalia a política de bucket para determinar se o proprietário do bucket negou explicitamente o acesso de Jill ao objeto.
3. No contexto de objeto, a autoridade de contexto é a conta da AWS 3333-3333-3333, a proprietária do objeto. O Amazon S3 avalia a ACL do objeto para determinar se Jill tem permissão para acessar o objeto. Se tiver, o Amazon S3 autorizará a solicitação.

## Diretrizes para usar as opções disponíveis de política de acesso

O Amazon S3 oferece suporte para políticas com base em recursos e políticas de usuário para gerenciar o acesso aos recursos do Amazon S3 (consulte [Gerenciamento de acesso a recursos \(opções de política de acesso\) \(p. 284\)](#)). As políticas com base em recursos incluem políticas de bucket, ACLs de bucket e ACLs de objeto. Esta seção descreve cenários específicos para usar políticas de acesso com base em recursos para gerenciar o acesso aos recursos do Amazon S3.

## Quando usar uma política de acesso com base em ACL (ACLs de bucket e objeto)

Os buckets e objetos têm ACLs associadas que você pode usar para conceder permissões. As seguintes seções descrevem cenários para usar ACLs de objeto e ACLs de bucket.

### Quando usar uma ACL de objeto

Além de uma ACL de objeto, o proprietário de objeto pode gerenciar permissões de objeto de outras formas. Por exemplo:

- Se a conta da AWS que possui o objeto também possuir o bucket, poderá gravar uma política de bucket para gerenciar permissões de objeto.
- Se a conta da AWS que possui o objeto desejar conceder permissão a um usuário em sua conta, poderá usar uma política de usuário.

Então quando usar ACLs de objeto para gerenciar permissões de objeto? Os cenários a seguir mostram quando você usa ACLs de objeto para gerenciar permissões de objeto.

- Uma ACL de objeto é a única maneira de gerenciar o acesso aos objetos que não são do proprietário do bucket – A conta da AWS que é proprietária do bucket pode conceder outra permissão de conta da AWS para fazer upload de objetos. O proprietário do bucket não detém esses objetos. A conta da AWS que criou o objeto deve conceder permissões usando ACLs de objeto.

#### Note

Um proprietário do bucket não pode conceder permissões em objetos que não possui. Por exemplo, uma política de bucket que concede permissões de objeto aplica-se somente a objetos do proprietário do bucket. Contudo, o proprietário do bucket, que paga as faturas, pode gravar uma política de bucket para negar acesso a todos os objetos no bucket, independentemente de quem o possui. O proprietário do bucket também pode excluir todos os objetos no bucket.

- As permissões variam por objeto e você precisa gerenciar permissões no nível do objeto – Você pode gravar uma única instrução de política que concede a uma conta da AWS permissão de leitura em milhões de objetos com um prefixo específico de nome de chave. Por exemplo, conceda permissão de leitura em objetos que começam com “logs” de prefixo de nome de chave. Contudo, se as permissões de acesso variarem por objeto, conceder permissões para objetos individuais usando uma política de bucket talvez não seja prático. Além disso, as políticas de bucket são limitadas a 20 KB.

Nesse caso, você pode achar que o uso de ACLs de objeto é uma alternativa apropriada. No entanto, mesmo uma ACL de objeto é limitada a, no máximo, 100 concessões (consulte [Visão geral da Lista de controle de acesso \(ACL\) \(p. 390\)](#)).

- As ACLs de objeto controlam somente permissões no nível do objeto – Há uma única política de bucket para o bucket todo, mas as ACLs de objeto são especificadas por objeto.

Uma conta da AWS que possui um bucket pode conceder outra permissão de conta da AWS para gerenciar a política de acesso. Ela permite que a conta altere algo na política. Para gerenciar permissões melhor, você pode optar por não conceder uma permissão tão ampla e conceder somente as permissões READ-ACP e WRITE-ACP em um subconjunto de objetos. Isso limita a conta para gerenciar permissões somente em objetos específicos atualizando ACLs de objeto individuais.

### Quando usar uma ACL de bucket

O único caso de uso recomendado para a ACL de bucket é conceder a permissão de gravação para o grupo de entrega de logs do Amazon S3 para gravar objetos de log de acesso no bucket (consulte [Registro](#)

em log de acesso ao servidor Amazon S3 (p. 625)). Se você quiser que o Amazon S3 forneça logs de acesso ao seu bucket, precisará conceder permissão de gravação no bucket ao grupo de entrega de log. A única maneira de conceder as permissões necessárias para o grupo de entrega de log é usar uma ACL de bucket, conforme exibido no seguinte fragmento de ACL de bucket.

```
<?xml version="1.0" encoding="UTF-8"?>
<AccessControlPolicy xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Owner>
    ...
  </Owner>
  <AccessControlList>
    <Grant>
      ...
    </Grant>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
        <URI>http://acs.amazonaws.com/groups/s3/LogDelivery</URI>
      </Grantee>
      <Permission>WRITE</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

## Quando usar uma política de bucket

Se uma conta da AWS que possui um bucket desejar conceder permissão aos usuários em sua conta, poderá usar uma política de bucket ou de usuário. Mas nos seguintes cenários, você precisará usar uma política de bucket.

- Você quer gerenciar permissões entre contas para todas as permissões do Amazon S3 – Você pode usar ACLs para conceder permissões entre contas a outras contas, mas as ACLs oferecem suporte somente para um conjunto finito de permissões ([Quais permissões posso conceder? \(p. 393\)](#)), que não inclui todas as permissões do Amazon S3. Por exemplo, você não pode conceder permissões em sub-recursos de bucket (consulte [Gerenciamento de permissões de acesso aos recursos do Amazon S3 \(p. 282\)](#)) usando uma ACL.

Embora as políticas de bucket e de usuário permitam conceder permissão para todas as operações do Amazon S3 (consulte [Especificação de permissões em uma política \(p. 330\)](#)), as políticas de usuário servem para gerenciar permissões para usuários em sua conta. Para permissões entre contas para outras contas da AWS ou usuários em outra conta, você deve usar uma política de bucket.

## Quando usar uma política de usuário

Geralmente, você pode usar uma política de usuário ou uma política de bucket para gerenciar permissões. Você pode optar por gerenciar permissões criando usuários e gerenciando permissões, individualmente, anexando políticas a usuários (ou grupos de usuário), ou pode achar que as políticas com base em recursos, como uma política de bucket, funcionam melhor para seu cenário.

Observe que o AWS Identity and Access Management (IAM) permite criar vários usuários em sua conta da AWS e gerenciar suas permissões por meio de políticas de usuário. Um usuário do IAM deve ter permissões na conta pai a qual pertence e na conta da AWS que possui o recurso que o usuário deseja acessar. As permissões podem ser concedidas do seguinte modo:

- Permissão na conta pai – A conta pai pode conceder permissões para seu usuário anexando uma política de usuário.
- Permissão do proprietário do recurso – O proprietário do recurso pode conceder permissão para o usuário do IAM (usando uma política de bucket) ou para a conta pai (usando uma política de bucket, uma ACL de bucket ou uma ACL de objeto).

É semelhante a uma criança que deseja brincar com um brinquedo que pertença a alguém. Nesse caso, a criança deve obter permissão de um responsável para brincar com o brinquedo e permissão do proprietário do brinquedo.

## Delegação de permissão

Se uma conta da AWS possuir um recurso, poderá conceder essas permissões para outra conta da AWS. Essa conta pode então delegar essas permissões, ou um subconjunto delas, para usuários da conta. Isso é chamado de delegação de permissão. Mas uma conta que recebe permissões de outra conta não pode delegar permissão entre contas para outra conta da AWS.

## Tópicos relacionados

Recomendamos que você analise, primeiramente, todos os tópicos introdutórios que explicam como gerenciar o acesso aos recursos do Amazon S3 e as diretrizes relacionadas. Para obter mais informações, consulte [Introdução ao gerenciamento de permissões de acesso aos recursos do Amazon S3 \(p. 282\)](#). Você pode usar os seguintes tópicos para obter mais informações sobre as opções específicas da política de acesso.

- [Uso de políticas de bucket e políticas de usuário \(p. 326\)](#)
- [Gerenciar o acesso com ACLs \(p. 390\)](#)

## Demonstrações com exemplo: gerenciar o acesso aos recursos do Amazon S3

Este tópico fornece exemplos de demonstrações introdutórias para conceder acesso aos recursos do Amazon S3. Esses exemplos usam o Console de gerenciamento da AWS para criar recursos (buckets, objetos, usuários) e conceder permissões a eles. Em seguida, os exemplos mostram como verificar as permissões usando as ferramentas da linha de comando, para que nenhum código precise ser escrito. Fornecemos comandos usando a Interface de linha de comando (CLI) da AWS e as ferramentas da AWS para Windows PowerShell.

- [Exemplo 1: Proprietário do bucket que concede permissões do bucket aos usuários \(p. 301\)](#)

Por padrão, os usuários do IAM criados na conta não têm permissões. Neste exercício, você concederá uma permissão de usuário para realizar operações no bucket e no objeto.

- [Exemplo 2: proprietário do bucket concedendo permissões de bucket entre contas. \(p. 306\)](#)

Neste exercício, um proprietário do bucket, conta A, concede permissões de conta cruzada para outra conta da AWS, conta B. Em seguida, a conta B delega essas permissões para os usuários em sua conta.

- Gerenciar permissões de objeto quando os proprietários do bucket e do objeto são diferentes

Nesse caso, os cenários do exemplo são um proprietário de bucket que concede permissões de objeto para outros, mas nem todos os objetos do bucket pertencem ao proprietário do bucket. De quais permissões o proprietário do bucket precisa e como ele pode delegar essas permissões?

A conta da AWS que cria um bucket é chamada de proprietário do bucket. O proprietário pode conceder permissões a outras contas da AWS para carregar objetos, e os proprietários desses objetos são as contas da AWS que os criaram. O proprietário do bucket não tem permissões sobre esses objetos criados por outras contas da AWS. Se o proprietário do bucket escreve uma política de bucket concedendo acesso aos objetos, a política não se aplica aos objetos pertencentes a outras contas.

Nesse caso, o proprietário do objeto deve, primeiro, conceder permissões ao proprietário do bucket usando uma ACL do objeto. Em seguida, o proprietário do bucket pode autorizar essas permissões de objeto para outros, para usuários em sua própria conta ou para outra conta da AWS, conforme ilustrado pelos exemplos a seguir.

- [Exemplo 3: o proprietário do bucket concede permissões aos usuários para objetos que não possui \(p. 312\)](#)

Neste exercício, primeiro o proprietário do bucket obtém permissões do proprietário do objeto. Em seguida, o proprietário do bucket delega tais permissões para usuários em sua própria conta.

- [Exemplo 4: Proprietário do bucket concede permissões entre contas a objetos que não possui \(p. 316\)](#)

Depois de receber as permissões do proprietário do objeto, o proprietário do bucket não pode delegar permissões para outras contas da AWS já que não há suporte para a delegação de conta cruzada ([consulte Delegação de permissão \(p. 296\)](#)). Em vez disso, o proprietário do bucket pode criar uma função do IAM com permissões para realizar determinadas operações (como obter objeto) e permitir que outra conta da AWS assuma essa função. Qualquer um que assumir a função pode, então, acessar os objetos. Este exemplo mostra como um proprietário do bucket pode usar uma função do IAM para habilitar essa delegação de conta cruzada.

## Antes de tentar as demonstrações de exemplo

Esses exemplos usam o Console de gerenciamento da AWS para criar recursos e conceder permissões. Para testar permissões, os exemplos usam ferramentas da linha de comando, AWS Command Line

Interface (CLI) e ferramentas da AWS para Windows PowerShell, para que nenhum código precise ser escrito. Para testar permissões você precisará configurar uma dessas ferramentas. Para obter mais informações, consulte [Configurar as ferramentas para as demonstrações com exemplos \(p. 298\)](#).

Além disso, ao criar recursos esses exemplos não usam credenciais raiz de uma conta da AWS. Em vez disso, crie um usuário administrador nessas contas para executar essas tarefas.

## Sobre o uso de um usuário administrador para criar recursos e conceder permissões

O AWS Identity and Access Management (IAM) recomenda não usar credenciais raiz da conta da AWS para fazer solicitações. Em vez disso, crie um usuário do IAM, conceda acesso total a esse usuário e, em seguida, use as credenciais desse usuário para interagir com a AWS. Esse usuário é conhecido como usuário administrador. Para obter mais informações, consulte [Credenciais de conta raiz vs. Credenciais de usuário do IAM](#) no AWS General Reference e [Melhores práticas do IAM](#) no Guia do usuário do IAM.

Todos as demonstrações com exemplos nesta seção usam as credenciais do usuário administrador. Caso você não tenha criado um usuário administrador para sua conta da AWS, os tópicos mostram como fazê-lo.

Observe que para fazer login no Console de gerenciamento da AWS usando as credenciais de usuário, você deverá usar o URL de login do usuário do IAM. O console do IAM fornece esse URL para a conta da AWS. Os tópicos mostram como obter o URL.

## Configurar as ferramentas para as demonstrações com exemplos

Os exemplos introdutórios (consulte [Demonstrações com exemplo: gerenciar o acesso aos recursos do Amazon S3 \(p. 297\)](#)) usam o Console de gerenciamento da AWS para criar recursos e conceder permissões. Para testar permissões, os exemplos usam ferramentas da linha de comando, AWS Command Line Interface (CLI) e ferramentas da AWS para Windows PowerShell, para que nenhum código precise ser escrito. Para testar permissões, você deve configurar uma dessas ferramentas.

### Para configurar a AWS CLI

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

[Configurar com a interface de linha de comando da AWS](#)

[Instalar a interface de linha de comando da AWS](#)

[Configurar a interface de linha de comando da AWS](#)

2. Defina o perfil padrão.

Você armazenará as credenciais de usuário no arquivo de configuração da CLI da AWS. Crie um perfil padrão no arquivo de configuração usando as credenciais da conta da AWS. Consulte [Arquivos de configuração e credenciais](#) para obter instruções sobre como localizar e editar o arquivo de configuração do AWS CLI.

```
[default]
aws_access_key_id = access key ID
aws_secret_access_key = secret access key
region = us-west-2
```

3. Verifique a configuração inserindo o comando a seguir no prompt de comando. Ambos os comandos não fornecem as credenciais explicitamente, de modo que as credenciais do perfil padrão são usadas.

- Experimente o comando de ajuda

```
aws help
```

- Use o aws s3 ls para obter uma lista dos buckets na conta configurada.

```
aws s3 ls
```

À medida que você avançar nas demonstrações, criará usuários e salvará credenciais de usuários nos arquivos de configuração ao criar perfis, conforme mostra o exemplo a seguir. Observe que esses perfis têm nomes (AccountAdmin e AccountBadmin):

```
[profile AccountAdmin]
aws_access_key_id = User AccountAdmin access key ID
aws_secret_access_key = User AccountAdmin secret access key
region = us-west-2

[profile AccountBadmin]
aws_access_key_id = Account B access key ID
aws_secret_access_key = Account B secret access key
region = us-east-1
```

Para executar um comando usando essas credenciais de usuário, adicione o parâmetro --profile especificando o nome do perfil. O comando da CLI da AWS a seguir recupera uma lista de objetos em examplebucket e especifica o perfil AccountBadmin.

```
aws s3 ls s3://examplebucket --profile AccountBadmin
```

Como alternativa, configure um conjunto de credenciais de usuário como perfil padrão alterando a variável de ambiente AWS\_DEFAULT\_PROFILE no prompt de comando. Depois que fizer isso, sempre que você executar comandos da CLI da AWS sem o parâmetro --profile, a CLI da AWS usará o perfil definido na variável de ambiente como perfil padrão.

```
$ export AWS_DEFAULT_PROFILE=AccountAdmin
```

Para configurar as ferramentas da AWS para Windows PowerShell

1. Faça o download e configure as ferramentas da AWS para Windows PowerShell. Para obter instruções, acesse [Baixar e instalar as ferramentas da AWS para Windows PowerShell](#) no Guia do usuário do AWS Tools para Windows PowerShell.

#### Note

Para carregar as ferramentas da AWS para o módulo do Windows PowerShell, você precisará habilitar a execução do script PowerShell. Para obter mais informações, acesse [Habilitar a execução de script](#) no Guia do usuário do AWS Tools para Windows PowerShell.

2. Para esses exercícios, especifique credenciais da AWS por sessão usando o comando Set-AWSCredentials. O comando salva as credenciais em um armazenamento persistente (-StoreAs parâmetro).

```
Set-AWSCredentials -AccessKey AccessKeyID -SecretKey SecretAccessKey -storeas string
```

3. Verifique a configuração.

- Execute Get-Command para recuperar uma lista de comandos disponíveis que podem ser usados para operações do Amazon S3.

```
Get-Command -module awspowershell -noun s3* -StoredCredentials string
```

- Execute o comando Get-S3Object para recuperar uma lista de objetos em um bucket.

```
Get-S3Object -BucketName bucketname -StoredCredentials string
```

Para obter uma lista de comandos, acesse [Cmdlets do Amazon Simple Storage Service](#).

Agora você está pronto testar os exercícios. Siga os links fornecidos no início da seção.

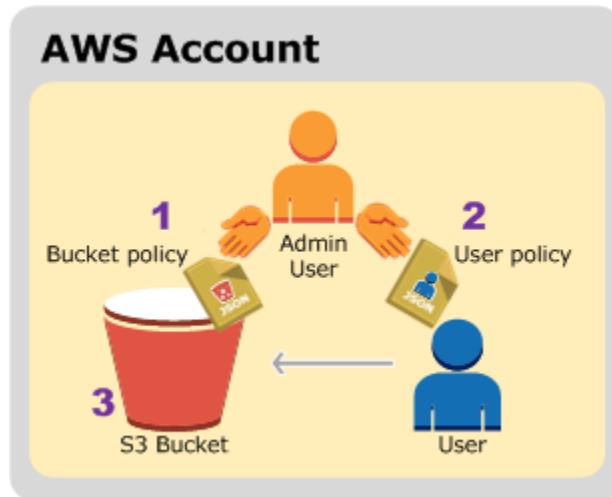
## Exemplo 1: Proprietário do bucket que concede permissões do bucket aos usuários

### Tópicos

- [Etapa 0: Preparação para a demonstração \(p. 302\)](#)
- [Etapa 1: Criar recursos \(um bucket e um usuário do IAM\) na Conta A e conceder permissões \(p. 302\)](#)
- [Etapa 2: Testar permissões \(p. 304\)](#)

Neste exercício, uma conta da AWS tem um bucket e há um usuário do IAM na conta. O usuário, por padrão, não tem nenhuma permissão. A conta pai deve conceder permissões ao usuário para executar qualquer tarefa. O proprietário do bucket e a conta pai à qual o usuário pertence são o mesmo. Portanto, a conta da AWS pode usar uma política do bucket, uma política de usuário, ou ambas para conceder suas permissões de usuário no bucket. Você concederá algumas permissões usando uma política do bucket e concederá outras permissões usando uma política de usuário.

Os passos a seguir resumem as etapas de demonstração:



1. O administrador da conta cria uma política do bucket concedendo um conjunto de permissões ao usuário.
2. O administrador da conta anexa uma política de usuário ao usuário concedendo permissões adicionais.
3. O usuário então testa as permissões concedidas por política do bucket e por política de usuário.

Para este exemplo, você precisará de uma conta da AWS. Em vez de usar as credenciais raiz da conta, você criará um usuário administrador (consulte [Sobre o uso de um usuário administrador para criar recursos e conceder permissões \(p. 298\)](#)). Nós nos referimos à conta da AWS e ao usuário administrador como se segue:

ID da conta	Conta referida como	Usuário administrador na conta
1111-1111-1111	Conta A	AccountAdmin

Todas as tarefas de criar usuários e conceder permissões são feitas no Console de gerenciamento da AWS. Para verificar permissões, a demonstração usa as ferramentas de linha de comando, a interface de linha de comando (CLI) da AWS e as ferramentas da AWS para Windows PowerShell, portanto, você não precisa gravar nenhum código para verificar as permissões.

## Etapa 0: Preparação para a demonstração

1. Certifique-se de que você tem uma conta da AWS e de que ela tem um usuário com privilégios de administrador.
  - a. Cadastre-se para obter uma conta, se necessário. Nós nos referimos a essa conta como conta A.
    - i. Acesse <https://aws.amazon.com/s3> e clique em Sign Up (Cadastrar-se).
    - ii. Siga as instruções da tela.

A AWS o notificará por e-mail quando sua conta estiver ativa e disponível para uso.
  - b. Na conta A, crie um usuário administrador AccountAdmin. Usando as credenciais da conta A, faça login no [console do IAM](#) e faça o seguinte:
    - i. Crie um usuário AccountAdmin e anote as credenciais de segurança do usuário.  
Para obter instruções, consulte [Criar um usuário do IAM em sua conta da AWS](#) no Guia do usuário do IAM.
    - ii. Conceda privilégios de administrador da AccountAdmin anexando uma política de usuário com pleno acesso.  
Para instruções, consulte [Trabalhar com políticas](#) no Guia do usuário do IAM.
    - iii. Anote o IAM User Sign-In URL (URL de login de usuário do IAM) para AccountAdmin. Você precisará usar esse URL para fazer login no Console de gerenciamento da AWS. Para obter mais informações sobre onde encontrá-lo, consulte [Como usuários fazem login na sua conta](#) no Guia do usuário do IAM. Anote o URL para cada uma das contas.
2. Configure a interface da linha de comando (CLI) da AWS ou as ferramentas da AWS para o Windows PowerShell. Certifique-se de ter salvado as credenciais de usuário do administrador deste modo:
  - Se estiver usando a CLI da AWS, crie um perfil, AccountAdmin, no arquivo config.
  - Se estiver usando as ferramentas da AWS para o Windows PowerShell, certifique-se de armazenar as credenciais para a sessão como AccountAdmin.

Para obter instruções, consulte [Configurar as ferramentas para as demonstrações com exemplos \(p. 298\)](#).

## Etapa 1: Criar recursos (um bucket e um usuário do IAM) na Conta A e conceder permissões

Com as credenciais do usuário AccountAdmin na Conta A e o URL especial de login do usuário do IAM, faça login no Console de gerenciamento da AWS e faça o seguinte:

1. Crie recursos (um bucket e um usuário do IAM)
  - a. No console do Amazon S3, crie um bucket. Anote a região da AWS; na qual você o criou. Para obter instruções, consulte [Como criar um bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.
  - b. No console do IAM, faça o seguinte:
    - i. Crie um usuário Dave.  
Para obter instruções, consulte [Criação de usuários do IAM \(Console de Gerenciamento da AWS\)](#) no Guia do usuário do IAM.
    - ii. Anote as credenciais de UserDave.

- iii. Anote o Nome de recurso da Amazon (ARN) para o usuário Dave. No console do IAM, selecione o usuário e a guia Summary (Resumo) fornecerá o ARN do usuário
2. Conceda permissões.

Como o proprietário do bucket e a conta pai a que o usuário pertence são um só, a conta da AWS pode conceder permissões de usuário usando uma política do bucket, uma política de usuário, ou ambas. Neste exemplo, você faz ambos. Se o objeto também for de propriedade da mesma conta, o proprietário do bucket pode conceder permissões de objeto na política do bucket (ou em uma política do IAM).

- a. No console do Amazon S3, anexe a seguinte política do bucket a *examplebucket*.

A política tem duas instruções.

- A primeira instrução concede a Dave as permissões de operação dos buckets `s3:GetBucketLocation` e `s3>ListBucket`.
- A segunda instrução concede a permissão `s3:GetObject`. Como a Conta A também possui o objeto, o administrador da conta pode conceder a permissão `s3:GetObject`.

Na instrução Principal, Dave é identificado por seu ARN de usuário. Para obter mais informações sobre elementos de política, consulte [Visão geral da linguagem da política de acesso \(p. 326\)](#).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "statement1",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::AccountA-ID:user/Dave"  
            },  
            "Action": [  
                "s3:GetBucketLocation",  
                "s3>ListBucket"  
            ],  
            "Resource": [  
                "arn:aws:s3:::examplebucket"  
            ]  
        },  
        {  
            "Sid": "statement2",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::AccountA-ID:user/Dave"  
            },  
            "Action": [  
                "s3:GetObject"  
            ],  
            "Resource": [  
                "arn:aws:s3:::examplebucket/*"  
            ]  
        }  
    ]  
}
```

- b. Crie uma política inline para o usuário Dave usando as seguintes políticas. A política concede a Dave a permissão `s3:PutObject`. Você precisa atualizar a política, fornecendo o nome de seu bucket.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "PermissionForObjectOperations",  
            "Effect": "Allow",  
            "Action": [  
                "s3:PutObject"  
            ],  
            "Resource": [  
                "arn:aws:s3:::examplebucket/*"  
            ]  
        }  
    ]  
}
```

Para instruções, consulte [Trabalhar com políticas em linha](#) no Guia do usuário do IAM. Observe que você precisa fazer login no console usando as credenciais da Conta A.

## Etapa 2: Testar permissões

Com as credenciais de Dave, verifique se as permissões funcionam. Você pode usar um dos dois procedimentos a seguir.

### Testar usando a CLI da AWS

1. Atualizar o arquivo de configuração da CLI da AWS, adicionando o perfil de UserDaveAccountA a seguir. Para obter mais informações, consulte [Configurar as ferramentas para as demonstrações com exemplos \(p. 298\)](#).

```
[profile UserDaveAccountA]  
aws_access_key_id = access-key  
aws_secret_access_key = secret-access-key  
region = us-east-1
```

2. Verifique se Dave pode executar as operações conforme concedido na política de usuário. Faça upload de um objeto de exemplo usando o seguinte comando put-object da CLI da AWS.

O parâmetro --body no comando identifica o arquivo de origem para fazer upload. Por exemplo, se o arquivo está na raiz da unidade C: de um computador Windows, você especifica c:\HappyFace.jpg. O parâmetro --key fornece o nome de chave para o objeto.

```
aws s3api put-object --bucket examplebucket --key HappyFace.jpg --body HappyFace.jpg --  
profile UserDaveAccountA
```

Execute o seguinte comando da CLI da AWS para obter o objeto.

```
aws s3api get-object --bucket examplebucket --key HappyFace.jpg OutputFile.jpg --  
profile UserDaveAccountA
```

### Teste usando as ferramentas da AWS para o Windows PowerShell

1. Armazene as credenciais de Dave como AccountADave. Depois, você usa essas credenciais para PUT e GET um objeto.

```
set-awscredentials -AccessKey AccessKeyId -SecretKey SecretAccessKey -storeas  
AccountADave
```

2. Faça upload de um objeto de exemplo usando as ferramentas da AWS para o comando Write-S3Object do Windows PowerShell usando as credenciais armazenadas do usuário Dave.

```
Write-S3Object -bucketname examplebucket -key HappyFace.jpg -file HappyFace.jpg -  
StoredCredentials AccountADave
```

Faça download do objeto anteriormente carregado.

```
Read-S3Object -bucketname examplebucket -key HappyFace.jpg -file Output.jpg -  
StoredCredentials AccountADave
```

## Exemplo 2: proprietário do bucket concedendo permissões de bucket entre contas.

### Tópicos

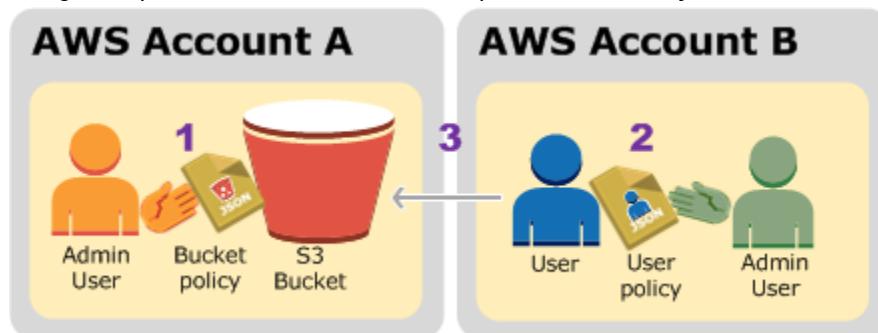
- [Etapa 0: Preparação para a demonstração \(p. 307\)](#)
- [Etapa 1: Faça as tarefas da Conta A \(p. 308\)](#)
- [Etapa 2: Faça as tarefas da Conta B \(p. 309\)](#)
- [Etapa 3: Crédito extra: tente negação explícita \(p. 310\)](#)
- [Etapa 4: Limpeza \(p. 311\)](#)

Uma conta da AWS — por exemplo, a Conta A — pode conceder a outra conta da AWS, a Conta B, permissão para acessar seus recursos, como buckets e objetos. A Conta B pode então delegar essas permissões para usuários em sua conta. Neste cenário de exemplo, o proprietário do bucket concede a permissão entre contas a outra conta para executar operações específicas no bucket.

#### Note

A Conta A também pode conceder diretamente a um usuário na Conta B permissões usando uma política de bucket. Mas o usuário ainda precisará ter permissão da conta pai, a Conta B, à qual o usuário pertence, mesmo que a Conta B não tenha permissões da Conta A. Desde que o usuário tenha permissão do proprietário do recurso e da conta pai, o usuário poderá acessar o recurso.

A seguir é apresentado um resumo das etapas de demonstração:



1. O usuário administrador da Conta A anexa uma política de bucket concedendo permissões entre contas à Conta B para executar operações específicas no bucket.

Observe que o usuário administrador da Conta B herdará automaticamente as permissões.

2. O usuário administrador da Conta B anexa uma política de usuário ao usuário delegando as permissões que recebeu da Conta A.
3. Em seguida, o usuário na Conta B verifica as permissões acessando um objeto no bucket de propriedade da Conta A.

Para este exemplo, você precisará de duas contas. A tabela a seguir mostra como nos referimos a essas contas e aos usuários administradores nessas contas. Conforme as diretrizes do IAM (veja [Sobre o uso de um usuário administrador para criar recursos e conceder permissões \(p. 298\)](#)), não usamos as credenciais raiz de conta nesta apresentação. Em vez disso, você cria um usuário administrador em cada conta e usa essas credenciais para criar recursos e conceder permissões a eles.

ID da conta da AWS	Conta referida como	Usuário administrador na conta
<a href="#">1111-1111-1111</a>	Conta A	AccountAdmin

ID da conta da AWS	Conta referida como	Usuário administrador na conta
2222-2222-2222	Conta B	AccountBadmin

Todas as tarefas de criar usuários e conceder permissões são feitas no Console de gerenciamento da AWS. Para verificar as permissões, o passo a passo usa as ferramentas da linha de comando, a interface de linha de comando (CLI) da AWS e as ferramentas da AWS para Windows PowerShell, portanto, você não precisa escrever nenhum código.

## Etapa 0: Preparação para a demonstração

1. Verifique se você tem duas contas da AWS e se cada conta tem um usuário administrador, conforme mostrado na tabela na seção anterior.
  - a. Cadastre-se em uma conta da AWS, se necessário.
    - i. Acesse <https://aws.amazon.com/s3/> e clique em Create an AWS Account (Criar uma conta da AWS).
    - ii. Siga as instruções da tela.

A AWS o notificará por e-mail quando sua conta estiver ativa e disponível para uso.
  - b. Usando as credenciais da Conta A, faça login no [Console do IAM](#) para criar o usuário administrador:
    - i. Crie um usuário AccountAdmin e anote as credenciais de segurança. Para obter instruções, consulte [Criar um usuário do IAM em sua conta da AWS](#) no Guia do usuário do IAM.
    - ii. Conceda privilégios de administrador da AccountAdmin anexando uma política de usuário com pleno acesso. Para instruções, consulte [Trabalhar com políticas](#) no Guia do usuário do IAM.
  - c. Enquanto estiver no console do IAM, anote a IAM User Sign-In URL (URL de login de usuário do IAM) no Dashboard (Painel). Todos os usuários nessa conta devem usar essa URL para fazer login no Console de gerenciamento da AWS.
2. Configure a interface da linha de comando (CLI) da AWS ou as ferramentas da AWS para o Windows PowerShell. Certifique-se de ter salvado as credenciais de usuário do administrador deste modo:
  - Se estiver usando a CLI da AWS, crie dois perfis, AccountAdmin e AccountBadmin, no arquivo de configuração.
  - Se estiver usando as ferramentas da AWS para o Windows PowerShell, certifique-se de armazenar as credenciais para a sessão como AccountAdmin e AccountBadmin.

Para obter instruções, consulte [Configurar as ferramentas para as demonstrações com exemplos \(p. 298\)](#).

3. Salve as credenciais do usuário administrador, também conhecidas como perfis. Você pode usar o nome do perfil em vez de especificar as credenciais para cada comando digitado. Para obter mais informações, consulte [Configurar as ferramentas para as demonstrações com exemplos \(p. 298\)](#).
  - a. Adicione perfis no arquivo de credenciais da AWS CLI para cada um dos usuários administradores nas duas contas.

```
[AccountAdmin]
aws_access_key_id = access-key-ID
aws_secret_access_key = secret-access-key
region = us-east-1

[AccountBadmin]
aws_access_key_id = access-key-ID
aws_secret_access_key = secret-access-key
region = us-east-1
```

- b. Se estiver usando as ferramentas da AWS para Windows PowerShell

```
set-awscredentials -AccessKey AcctA-access-key-ID -SecretKey AcctA-secret-access-
key -storeas AccountAdmin
set-awscredentials -AccessKey AcctB-access-key-ID -SecretKey AcctB-secret-access-
key -storeas AccountBadmin
```

## Etapa 1: Faça as tarefas da Conta A

### Etapa 1.1: Faça login no Console de Gerenciamento da AWS.

Usando a URL de login do usuário do IAM para a Conta A, primeiro faça login no Console de gerenciamento da AWS como o usuário AccountAdmin. Esse usuário criará um bucket e anexará uma política a ele.

### Etapa 1.2: Criar um bucket

1. No console do Amazon S3, crie um bucket. Este exercício supõe que o bucket é criado na região Leste dos EUA (Norte da Virgínia) e que o nome é examplebucket.

Para obter instruções, consulte [Como criar um bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

2. Faça upload de um objeto de exemplo no bucket.

Para obter instruções, vá para [Adicionar um objeto a um bucket](#) no Guia de conceitos básicos do Amazon Simple Storage Service.

### Etapa 1.3: Anexar uma política de bucket para conceder permissões entre contas para a Conta B

A política de bucket concede as permissões s3:GetBucketLocation e s3>ListBucket para a Conta B. Supõe-se que você ainda esteja conectado no console usando as credenciais do usuário AccountAdmin.

1. Anexe a política de bucket a seguir ao examplebucket. A política concede à Conta B permissão para as ações s3:GetBucketLocation e s3>ListBucket.

Para obter instruções, consulte [Como adicionar uma política de bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Example permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::AccountB-ID:root"
            }
        }
    ]
}
```

```
        },
        "Action": [
            "s3:GetBucketLocation",
            "s3>ListBucket"
        ],
        "Resource": [
            "arn:aws:s3:::examplebucket"
        ]
    }
}
```

2. Verifique se a Conta B (e, portanto, do usuário administrador) pode executar as operações.

- Usar a CLI da AWS

```
aws s3 ls s3://examplebucket --profile AccountBadmin
aws s3api get-bucket-location --bucket examplebucket --profile AccountBadmin
```

- Usar as ferramentas da AWS para Windows PowerShell

```
get-s3object -BucketName example2bucket -StoredCredentials AccountBadmin
get-s3bucketlocation -BucketName example2bucket -StoredCredentials AccountBadmin
```

## Etapa 2: Faça as tarefas da Conta B

Agora o administrador da Conta B cria um usuário, Dave, e delega as permissões recebidas da Conta A.

### Etapa 2.1: Fazer login no Console de Gerenciamento da AWS

Usando a URL de login do usuário do IAM da Conta B, primeiro faça login no Console de gerenciamento da AWS como o usuário AccountBadmin.

### Etapa 2.2: Criar o usuário Dave na Conta B

No console do IAM, crie um usuário, Dave.

Para obter instruções, consulte [Criação de usuários do IAM \(Console de Gerenciamento da AWS\)](#) no Guia do usuário do IAM.

### Etapa 2.3: Delegar permissões para o usuário Dave

Crie uma política inline para o usuário Dave usando as seguintes políticas. Você precisará atualizar a política fornecendo o nome do bucket.

Supõe-se que você está conectado no console usando as credenciais do usuário AccountBadmin.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Example",
            "Effect": "Allow",
            "Action": [
                "s3>ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::examplebucket"
            ]
        }
    ]
}
```

```
        ]  
    }
```

Para instruções, consulte [Trabalhar com políticas em linha](#) no Guia do usuário do IAM.

#### Etapa 2.4: Testar permissões

Agora Dave, na conta B, pode listar o conteúdo do `examplebucket` de propriedade da Conta A. Você pode verificar as permissões usando um dos seguintes procedimentos.

##### Testar usando a CLI da AWS

1. Adicione o perfil UserDave ao arquivo de configuração da AWS CLI. Para obter mais informações sobre o arquivo de configuração, consulte [Configurar as ferramentas para as demonstrações com exemplos \(p. 298\)](#)

```
[profile UserDave]  
aws_access_key_id = access-key  
aws_secret_access_key = secret-access-key  
region = us-east-1
```

2. No prompt de comando, digite o seguinte comando da AWS CLI para verificar se Dave agora pode obter uma lista de objetos do `examplebucket` de propriedade da Conta A. Observe que o comando especifica o perfil UserDave.

```
aws s3 ls s3://examplebucket --profile UserDave
```

Dave não tem nenhuma outra permissão. Portanto, se ele tentar qualquer outra operação — por exemplo, o seguinte get bucket location — o Amazon S3 retornará permissão negada.

```
aws s3api get-bucket-location --bucket examplebucket --profile UserDave
```

##### Testar usando as ferramentas da AWS para Windows PowerShell

1. Armazene as credenciais de Dave como AccountBDave.

```
set-awscredentials -AccessKey AccessKeyID -SecretKey SecretAccessKey -storeas  
AccountBDave
```

2. Teste o comando List Bucket.

```
get-s3object -BucketName example2bucket -StoredCredentials AccountBDave
```

Dave não tem nenhuma outra permissão. Portanto, se ele tentar qualquer outra operação — por exemplo, o seguinte get bucket location — o Amazon S3 retornará permissão negada.

```
get-s3bucketlocation -BucketName example2bucket -StoredCredentials AccountBDave
```

#### Etapa 3: Crédito extra: tente negação explícita

Você pode ter permissões concedidas por uma ACL, por uma política de bucket e por uma política de usuário. Mas se houver uma negação explícita definida por uma política de bucket ou por uma política

de usuário, a negação explícita terá precedência sobre qualquer outra permissão. Para testar, vamos atualizar a política de bucket e negar explicitamente a permissão s3>ListBucket para a conta B. A política também concede a permissão s3>ListBucket, mas a negação explícita tem precedência, e a Conta B ou os usuários da Conta B não poderão listar objetos no examplebucket.

1. Usando as credenciais do usuário AccountAdmin na Conta A, substitua a política de bucket pela seguinte.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "Example permissions",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::AccountB-ID:root"  
            },  
            "Action": [  
                "s3:GetBucketLocation",  
                "s3>ListBucket"  
            ],  
            "Resource": [  
                "arn:aws:s3:::examplebucket"  
            ]  
        },  
        {  
            "Sid": "Deny permission",  
            "Effect": "Deny",  
            "Principal": {  
                "AWS": "arn:aws:iam::AccountB-ID:root"  
            },  
            "Action": [  
                "s3>ListBucket"  
            ],  
            "Resource": [  
                "arn:aws:s3:::examplebucket"  
            ]  
        }  
    ]  
}
```

2. Agora, se você tentar obter uma lista de bucket usando as credenciais de AccountBadmin, você terá o acesso negado.

- Usar a CLI da AWS:

```
aws s3 ls s3://examplebucket --profile AccountBadmin
```

- Usar as ferramentas da AWS para Windows PowerShell:

```
get-s3object -BucketName example2bucket -StoredCredentials AccountBDave
```

## Etapa 4: Limpeza

1. Depois de terminar os testes, você pode seguir uma das etapas a seguir para fazer a limpeza.
  - Faça login no Console de gerenciamento da AWS ([Console de gerenciamento da AWS](#)) usando as credenciais da Conta A e faça o seguinte:
    - No console do Amazon S3, remova a política de bucket anexada a `examplebucket`. Nas Propriedades do bucket, exclua a política na seção Permissions (Permissões).

- Se o bucket foi criado para este exercício, no console do Amazon S3, exclua os objetos e, em seguida, exclua o bucket.
  - No console do IAM, remova o usuário AccountAdmin.
2. Faça login no Console de gerenciamento da AWS ([Console de gerenciamento da AWS](#)) usando as credenciais da Conta B. No console do IAM, exclua o usuário AccountBadmin.

## Exemplo 3: o proprietário do bucket concede permissões aos usuários para objetos que não possui

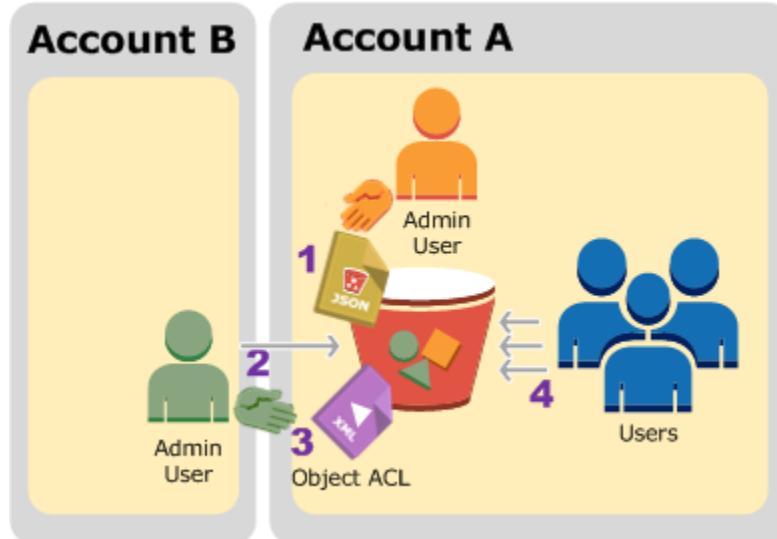
### Tópicos

- [Etapa 0: Preparação para a demonstração \(p. 313\)](#)
- [Etapa 1: Faça as tarefas da Conta A \(p. 314\)](#)
- [Etapa 2: Faça as tarefas da Conta B \(p. 315\)](#)
- [Etapa 3: Testar permissões \(p. 315\)](#)
- [Etapa 4: Limpeza \(p. 316\)](#)

O cenário deste exemplo é que o proprietário do bucket deseja conceder permissão para acessar objetos, mas nem todos os objetos no bucket são de propriedade do proprietário do bucket. Como um proprietário de bucket pode conceder permissão para objetos que não possui? Para este exemplo, o proprietário do bucket está tentando conceder permissão aos usuários em sua própria conta.

Um proprietário de bucket pode habilitar outras contas da AWS para fazer upload de objetos. Esses objetos são de propriedade das contas que os criou. O proprietário do bucket não possui objetos próprios que não foram criados pelo proprietário do bucket. Portanto, para o proprietário do bucket conceder acesso a esses objetos, o proprietário do objeto deve primeiro conceder permissão ao proprietário do bucket usando um objeto da ACL. O proprietário do bucket pode delegar essas permissões por meio da política de bucket. Neste exemplo, o proprietário do bucket delega permissão aos usuários em sua própria conta.

A seguir é apresentado um resumo das etapas de demonstração:



1. O usuário administrador da Conta A anexa uma política de bucket com duas declarações.

- Habilitar permissão entre contas para a Conta B fazer upload de objetos.
- Permitir que um usuário em sua própria conta acesse objetos no bucket.

2. O usuário administrador da Conta B faz upload de objetos no bucket de propriedade da Conta A.
3. O administrador da Conta B atualiza a ACL do objeto adicionando uma concessão que dá ao proprietário do bucket permissão de controle total sobre o objeto.
4. O usuário na Conta A verifica acessando objetos no bucket, independentemente de quem os possui.

Para este exemplo, você precisará de duas contas. A tabela a seguir mostra como nos referimos a essas contas e aos usuários administradores nessas contas. Conforme as diretrizes do IAM (veja [Sobre o uso de um usuário administrador para criar recursos e conceder permissões \(p. 298\)](#)), não usamos as credenciais raiz de conta nesta apresentação. Em vez disso, você cria um usuário administrador em cada conta e usa essas credenciais para criar recursos e conceder permissões a eles.

ID da conta da AWS	Conta referida como	Usuário administrador na conta
<a href="#"><b>1111-1111-1111</b></a>	Conta A	AccountAdmin
<a href="#"><b>2222-2222-2222</b></a>	Conta B	AccountBadmin

Todas as tarefas de criar usuários e conceder permissões são feitas no Console de gerenciamento da AWS. Para verificar as permissões, o passo a passo usa as ferramentas da linha de comando, a interface de linha de comando (CLI) da AWS e as ferramentas da AWS para Windows PowerShell, portanto, você não precisa escrever nenhum código.

## Etapa 0: Preparação para a demonstração

1. Verifique se você tem duas contas da AWS e se cada conta tem um usuário administrador, conforme mostrado na tabela na seção anterior.
  - a. Cadastre-se em uma conta da AWS, se necessário.
    - i. Acesse <https://aws.amazon.com/s3/> e clique em Create an AWS Account (Criar uma conta da AWS).
    - ii. Siga as instruções da tela. A AWS o notificará por e-mail quando sua conta estiver ativa e disponível para uso.
  - b. Usando as credenciais da Conta A, faça login no [console do IAM](#) e faça o seguinte para criar um usuário administrador:
    - Crie um usuário AccountAdmin e anote as credenciais de segurança. Para obter mais informações sobre como adicionar usuários, consulte [Criar um usuário do IAM na sua conta da AWS](#) no Guia do usuário do IAM.
    - Conceda privilégios de administrador da AccountAdmin anexando uma política de usuário com pleno acesso. Para instruções, consulte [Trabalhar com políticas](#) no Guia do usuário do IAM.
    - No Dashboard (Painel) do console do IAM, anote a IAM User Sign-In URL (URL de login de usuário do IAM). Os usuários nessa conta devem usar esse URL para fazer login no Console de gerenciamento da AWS. Para obter mais informações, consulte [Como os usuários fazem login em sua conta](#) no Guia do usuário do IAM.
  - c. Repita a etapa anterior usando as credenciais da Conta B e crie um usuário administrador AccountBadmin.
2. Configure a interface da linha de comando (CLI) da AWS ou as ferramentas da AWS para o Windows PowerShell. Certifique-se de ter salvado as credenciais de usuário do administrador deste modo:
  - Se estiver usando a CLI da AWS, crie dois perfis, AccountAdmin e AccountBadmin, no arquivo de configuração.
  - Se estiver usando as ferramentas da AWS para o Windows PowerShell, certifique-se de armazenar as credenciais para a sessão como AccountAdmin e AccountBadmin.

Para obter instruções, consulte [Configurar as ferramentas para as demonstrações com exemplos \(p. 298\)](#).

## Etapa 1: Faça as tarefas da Conta A

### Etapa 1.1: Faça login no Console de Gerenciamento da AWS.

Usando a URL de login do usuário do IAM para a Conta A, primeiro faça login no Console de gerenciamento da AWS como o usuário AccountAdmin. Esse usuário criará um bucket e anexará uma política a ele.

### Etapa 1.2: Criar um bucket, um usuário e adicionar uma política de bucket que concede permissões ao usuário

1. No console do Amazon S3, crie um bucket. Este exercício supõe que o bucket é criado na região Leste dos EUA (Norte da Virgínia) e que o nome é examplebucket.

Para obter instruções, consulte [Como criar um bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

2. No console do IAM, crie um usuário Dave.

Para obter instruções, consulte [Criação de usuários do IAM \(Console de Gerenciamento da AWS\)](#) no Guia do usuário do IAM.

3. Anote as credenciais de Dave.
4. No console do Amazon S3, anexe a seguinte política de bucket ao bucket examplebucket. Para obter instruções, consulte [Como adicionar uma política de bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service. Siga as etapas para adicionar uma política de bucket. Para obter informações sobre como encontrar IDs de conta, consulte [Encontrar seu ID da conta da AWS](#).

A política concede à Conta B as permissões s3:PutObject e s3>ListBucket. A política também concede ao usuário Dave a permissão s3:GetObject.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "Statement1",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::AccountB-ID:root"  
            },  
            "Action": [  
                "s3:PutObject"  
            ],  
            "Resource": [  
                "arn:aws:s3:::examplebucket/*"  
            ]  
        },  
        {  
            "Sid": "Statement3",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::AccountA-ID:user/Dave"  
            },  
            "Action": [  
                "s3:GetObject"  
            ]  
        }  
    ]  
}
```

```
        ],
        "Resource": [
            "arn:aws:s3:::examplebucket/*"
        ]
    }
}
```

## Etapa 2: Faça as tarefas da Conta B

Agora que a Conta B tem permissões para executar operações no bucket da Conta A, o administrador da Conta B fará o seguinte:

- Fazer upload de um objeto no bucket da Conta A.
- Adicionar uma concessão à ACL do objeto para permitir que a Conta A, a proprietária do bucket, tenha controle total.

### Usar a CLI da AWS

1. Usando o comando da AWS CLI `put-object`, fazer upload de um objeto. O parâmetro `--body` no comando identifica o arquivo de origem para fazer upload. Por exemplo, se o arquivo estiver na unidade C: de um computador Windows, você especificará `c:\HappyFace.jpg`. O parâmetro `--key` fornece o nome de chave para o objeto.

```
aws s3api put-object --bucket examplebucket --key HappyFace.jpg --body HappyFace.jpg --profile AccountBAdmin
```

2. Adicionar uma concessão à ACL do objeto para permitir controle total do objeto ao proprietário do bucket. Para obter informações sobre como encontrar um ID de usuário canônico, consulte [Encontrar seu ID de usuário canônico da conta](#).

```
aws s3api put-object-acl --bucket examplebucket --key HappyFace.jpg --grant-full-control id="AccountA-CanonicalUserID" --profile AccountBAdmin
```

### Usar as ferramentas da AWS para Windows PowerShell

1. Usar as ferramentas da AWS `Write-S3Object` para comando e upload de um objeto pelo Windows PowerShell.

```
Write-S3Object -BucketName examplebucket -key HappyFace.jpg -file HappyFace.jpg -StoredCredentials AccountBAdmin
```

2. Adicionar uma concessão à ACL do objeto para permitir controle total do objeto ao proprietário do bucket.

```
Set-S3ACL -BucketName examplebucket -Key HappyFace.jpg -CannedACLName "bucket-owner-full-control" -StoredCreden
```

## Etapa 3: Testar permissões

Verifique agora se o usuário Dave na Conta A pode acessar o objeto de propriedade da Conta B.

## Usar a CLI da AWS

1. Adicione as credenciais do usuário Dave ao arquivo config da AWS CLI e crie um novo perfil, UserDaveAccountA. Para obter mais informações, consulte [Configurar as ferramentas para as demonstrações com exemplos \(p. 298\)](#).

```
[profile UserDaveAccountA]
aws_access_key_id = access-key
aws_secret_access_key = secret-access-key
region = us-east-1
```

2. Execute o comando get-object da AWS CLI para baixar o HappyFace.jpg e salve-o localmente. Você fornece credenciais ao usuário Dave adicionando o parâmetro --profile.

```
aws s3api get-object --bucket examplebucket --key HappyFace.jpg Outputfile.jpg --profile UserDaveAccountA
```

## Usar as ferramentas da AWS para Windows PowerShell

1. Armazene as credenciais da AWS do usuário Dave, como UserDaveAccountA, para armazenamento persistente.

```
Set-AWSCredentials -AccessKey UserDave-AccessKey -SecretKey UserDave-SecretAccessKey -storeas UserDaveAccountA
```

2. Execute o comando Read-S3Object para baixar o objeto HappyFace.jpg e salvá-lo localmente. Você fornece credenciais ao usuário Dave adicionando o parâmetro -StoredCredentials.

```
Read-S3Object -BucketName examplebucket -Key HappyFace.jpg -file HappyFace.jpg -StoredCredentials UserDaveAccountA
```

## Etapa 4: Limpeza

1. Depois de terminar os testes, você pode seguir uma das etapas a seguir para fazer a limpeza.
  - Faça login no Console de gerenciamento da AWS ([Console de gerenciamento da AWS](#)) usando as credenciais da Conta A e faça o seguinte:
    - No console do Amazon S3, remova a política de bucket anexada a `examplebucket`. Nas Propriedades do bucket, exclua a política na seção Permissions (Permissões).
    - Se o bucket foi criado para este exercício, no console do Amazon S3, exclua os objetos e, em seguida, exclua o bucket.
    - No console do IAM, remova o usuário AccountAdmin.
2. Faça login no Console de gerenciamento da AWS ([Console de gerenciamento da AWS](#)) usando as credenciais da Conta B. No console do IAM, exclua o usuário AccountBadmin.

## Exemplo 4: Proprietário do bucket concede permissões entre contas a objetos que não possuem

### Tópicos

- [Histórico: Permissões entre contas e uso de funções do IAM \(p. 317\)](#)
- [Etapa 0: Preparação para a demonstração \(p. 318\)](#)

- [Etapa 1: Faça as tarefas da Conta A \(p. 320\)](#)
- [Etapa 2: Faça as tarefas da Conta B \(p. 322\)](#)
- [Etapa 3: Faça as tarefas da Conta C \(p. 323\)](#)
- [Etapa 4: Limpeza \(p. 324\)](#)
- [Recursos relacionados \(p. 325\)](#)

Neste cenário de exemplo, você possui um bucket e habilitou outras contas da AWS para fazer upload de objetos. Ou seja, seu bucket pode ter objetos de propriedade de outras contas da AWS.

Agora, suponha que, como proprietário do bucket, você precise conceder permissão entre contas aos objetos, independentemente de quem seja o proprietário, a um usuário em outra conta. Por exemplo, esse usuário pode ser um aplicativo de faturamento que precise acessar metadados de objeto. Há dois problemas principais:

- O proprietário do bucket não tem permissões sobre esses objetos criados por outras contas da AWS. Então para que o proprietário do bucket possa conceder permissões sobre objetos que não possui, o proprietário do objeto, a conta da AWS que criou os objetos, deve primeiro conceder permissão ao proprietário do bucket. Depois, o proprietário do bucket pode delegar essas permissões.
- A conta do proprietário do bucket pode delegar permissões a usuários em sua própria conta (veja [Exemplo 3: o proprietário do bucket concede permissões aos usuários para objetos que não possui \(p. 312\)](#)), mas não pode delegar permissões para outras contas da AWS, porque não há suporte para a delegação entre contas.

Neste cenário, o proprietário do bucket pode criar uma função do AWS Identity and Access Management com permissão para acessar objetos e conceder permissão a outra conta da AWS para assumir a função temporariamente, permitindo que ela acesse objetos no bucket.

## Histórico: Permissões entre contas e uso de funções do IAM

As funções do IAM permitem vários cenários para delegar acesso a seus recursos, e o acesso entre contas é um dos cenários principais. Neste exemplo, o proprietário do bucket, a Conta A, usa uma função do IAM para delegar temporariamente acessos a objetos entre contas a usuários em outra conta da AWS, Conta C. Cada função do IAM que você criar tem duas políticas anexadas a ela:

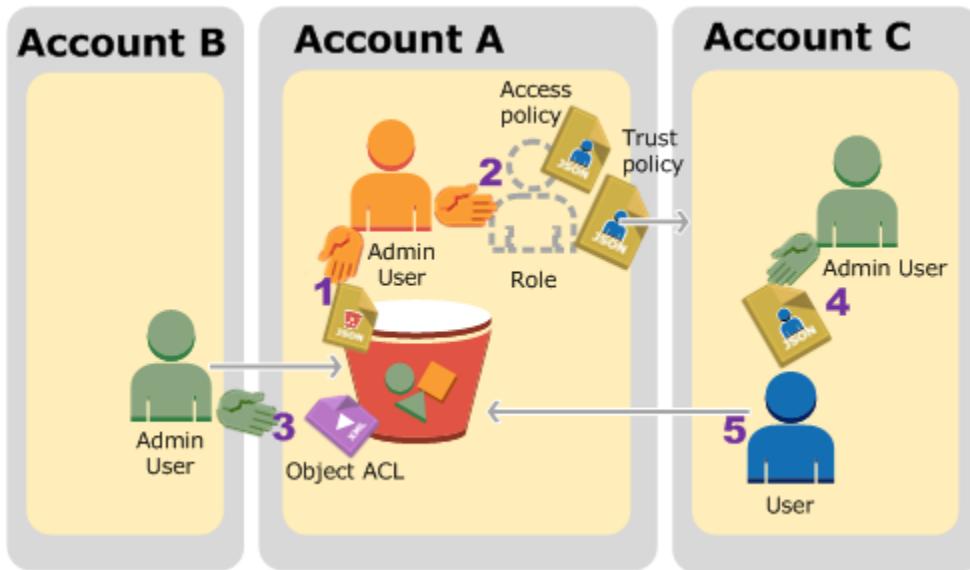
- Uma política de confiança que identifica outra conta da AWS que pode assumir a função.
- Uma política de acesso que define quais permissões — por exemplo, `s3:GetObject` — são permitidas quando alguém assume a função. Para obter uma lista de permissões que você pode especificar em uma política, consulte [Especificação de permissões em uma política \(p. 330\)](#).

A conta da AWS identificada na política de confiança então concede sua permissão de usuário para assumir a função. O usuário pode então fazer o seguinte para acessar os objetos:

- Assumir a função e, em resposta, obter credenciais de segurança temporárias.
- Usando as credenciais de segurança temporárias, acessar os objetos no bucket.

Para obter mais informações sobre as funções do IAM, acesse [Funções do IAM](#) no Guia do usuário do IAM.

A seguir é apresentado um resumo das etapas de demonstração:



1. O usuário administrador da Conta A anexa a política do bucket que concede à Conta B uma permissão condicional para fazer upload de objetos.
2. O administrador da Conta A cria uma função do IAM, estabelecendo a confiança com a Conta C, e assim os usuários dessa conta podem acessar a Conta A. A política de acesso anexada à função limita o que o usuário na Conta C pode fazer quando acessa a Conta A.
3. O administrador da Conta B faz upload de um objeto no bucket de propriedade da Conta A, concedendo permissão de controle total ao proprietário do bucket.
4. O administrador da Conta C cria um usuário e anexa uma política de usuário que permite que o usuário assuma a função.
5. O usuário na Conta C primeiro assume a função, que retorna as credenciais de segurança temporárias ao usuário. Usando essas credenciais de segurança temporárias, o usuário então acessa os objetos no bucket.

Para este exemplo, você precisará de três contas. A tabela a seguir mostra como nos referimos a essas contas e aos usuários administradores nessas contas. Conforme as diretrizes do IAM (veja [Sobre o uso de um usuário administrador para criar recursos e conceder permissões \(p. 298\)](#)), não usamos as credenciais raiz de conta nesta apresentação. Em vez disso, você cria um usuário administrador em cada conta e usa essas credenciais para criar recursos e conceder permissões a eles.

ID da conta da AWS	Conta referida como	Usuário administrador na conta
<b>1111-1111-1111</b>	Conta A	AccountAadmin
<b>2222-2222-2222</b>	Conta B	AccountBadmin
<b>3333-3333-3333</b>	Conta C	AccountCadmin

## Etapa 0: Preparação para a demonstração

### Note

É aconselhável abrir um editor de texto e escrever algumas informações enquanto você passa pelas etapas. Especificamente, você vai precisar dos IDs de conta, IDs de usuários canônicos,

URLs de login de usuário do IAM de cada conta para se conectar ao console, e Nomes de recurso da Amazon (ARN) dos usuários do IAM e funções.

1. Certifique-se de ter três contas da AWS e de que cada conta tenha um usuário administrador conforme exibido na tabela na seção anterior.
  - a. Cadastre-se para contas da AWS, se necessário. Nós nos referimos a essas contas como Conta A, Conta B e Conta C.
    - i. Acesse <https://aws.amazon.com/s3/> e clique em Create an AWS Account (Criar uma conta da AWS).
    - ii. Siga as instruções da tela.
- A AWS o notificará por e-mail quando sua conta estiver ativa e disponível para uso.
- b. Usando as credenciais da Conta A, faça login no [console do IAM](#) e faça o seguinte para criar um usuário administrador:
  - Crie um usuário AccountAdmin e anote as credenciais de segurança. Para obter mais informações sobre como adicionar usuários, consulte [Criar um usuário do IAM na sua conta da AWS](#) no Guia do usuário do IAM.
  - Conceda privilégios de administrador da AccountAdmin anexando uma política de usuário com pleno acesso. Para instruções, consulte [Trabalhar com políticas](#) no Guia do usuário do IAM.
  - No Dashboard (Painel) do console do IAM, anote o IAM User Sign-In URL (URL de login de usuário do IAM). Os usuários nessa conta devem usar esse URL para fazer login no Console de gerenciamento da AWS. Para obter mais informações, consulte [Como os usuários fazem login na conta](#) no Guia do usuário do IAM.
- c. Repita a etapa anterior para criar usuários administradores na Conta B e na Conta C.

2. Para a Conta C, anote o ID da conta.

Quando criar uma função do IAM na Conta A, a política de confiança concederá à Conta C a permissão para assumir a função especificando o ID da conta. Você pode localizar as informações da conta da seguinte forma:

- a. Acesse <https://aws.amazon.com/> e, no menu suspenso My Account/Console (Minha conta/Console), selecione Security Credentials (Credenciais de segurança).
  - b. Faça login usando as credenciais da conta apropriadas.
  - c. Clique em Account Identifiers (Identificadores de conta) e anote o AWS Account ID (ID da conta da AWS) e o Canonical User ID (ID de usuário canônico).
3. Ao criar uma política do bucket, você precisará das seguintes informações. Anote esses valores:
    - ID de usuário canônico da Conta A – Quando o administrador da Conta A conceder permissão condicional para fazer upload de objeto ao administrador da Conta B, a condição especifica o ID de usuário canônico do usuário da Conta A que deverá ter pleno controle dos objetos.

#### Note

O ID de usuário canônico é o único conceito do Amazon S3. Ele é a versão oculta de 64 caracteres do ID da conta.

- Nome de recurso da Amazon (ARN) de usuário para o administrador da conta B – Você pode encontrar o Nome de recurso da Amazon (ARN) do usuário no console do IAM. Você precisará selecionar o usuário e encontrar o Nome de recurso da Amazon (ARN) de usuário na guia Summary (Resumo).

Na política do bucket, você concede ao AccountBadmin permissão para fazer upload de objetos e especifica o usuário usando o Nome de recurso da Amazon (ARN). Aqui está um exemplo de valor do Nome de recurso da Amazon (ARN):

```
arn:aws:iam::AccountB-ID:user/AccountBadmin
```

4. Configure a interface da linha de comando (CLI) da AWS ou as ferramentas da AWS para o Windows PowerShell. Certifique-se de ter salvado as credenciais de usuário do administrador deste modo:
  - Se estiver usando a CLI da AWS, crie perfis, AccountAdmin e AccountBadmin, no arquivo config.
  - Se estiver usando as ferramentas da AWS para o Windows PowerShell, certifique-se de armazenar as credenciais para a sessão como AccountAdmin e AccountBadmin.

Para obter instruções, consulte [Configurar as ferramentas para as demonstrações com exemplos \(p. 298\)](#).

## Etapa 1: Faça as tarefas da Conta A

Neste exemplo, a Conta A é o proprietário do bucket. Então o usuário AccountAdmin na Conta A vai criar um bucket, anexar uma política do bucket concedendo ao administrador da Conta B permissão para fazer upload de objetos, criar uma função do IAM que concede permissão à Conta C para assumir a função de maneira que ela possa acessar objetos no bucket.

### Etapa 1.1: Faça login no Console de Gerenciamento da AWS.

Usando o URL de login de usuário do IAM para a Conta A, primeiro faça login no Console de gerenciamento da AWS como usuário AccountAdmin. Esse usuário criará um bucket e anexará uma política a ele.

### Etapa 1.2: Crie um bucket e anexe uma política do bucket

No console do Amazon S3, faça o seguinte:

1. Crie um bucket. Este exercício supõe que o nome do bucket é `examplebucket`.

Para obter instruções, consulte [Como criar um bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

2. Anexe a seguinte política do bucket, concedendo ao administrador da Conta B uma permissão condicional para fazer upload de objetos.

Você precisa atualizar a política fornecendo seus próprios valores para `examplebucket`, `AccountB-ID`, e `CanonicalUserId-of-AWSaccountA-BucketOwner`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "111",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::AccountB-ID:user/AccountBadmin"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::examplebucket/*"
        },
        {
            "Sid": "112",
            "Effect": "Deny",
            "Principal": {
                "AWS": "arn:aws:iam::AccountB-ID:user/AccountBadmin"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::examplebucket/*"
        }
    ]
}
```

```
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::examplebucket/*",
    "Condition": {
        "StringNotEquals": {
            "s3:x-amz-grant-full-control": "id=CanonicalUserId-of-AWSaccountA-BucketOwner"
        }
    }
}
```

### Etapa 1.3: Crie uma função do IAM para permitir à Conta C acesso entre contas à Conta A

No console do IAM, crie uma função do IAM (“exampleroles”) que conceda à Conta C permissão para assumir a função. Certifique-se que você ainda está conectado como administrador da Conta A porque a função deve ser criada na Conta A.

1. Antes de criar a função, prepare as políticas gerenciadas que definem as permissões necessárias à função. Em uma etapa posterior, você anexará essa política à função.
  - a. No painel de navegação à esquerda, clique em Policies (Políticas) e, em seguida, clique em Create Policy (Criar política).
  - b. Ao lado de Create Your Own Policy (Criar sua própria política), clique em Select (Selecionar).
  - c. Insira access-accountA-bucket no campo Policy Name (Nome da política).
  - d. Copie a política de acesso a seguir e cole-a no campo Policy Document (Documento de políticas). A política de acesso concede permissão da função s3:GetObject, e assim, quando o usuário da Conta C assumir a função, ele só poderá executar a operação s3:GetObject.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::examplebucket/*"
        }
    ]
}
```

- e. Clique em Create Policy (Criar política).

As novas políticas aparecem na lista de políticas gerenciadas.

2. No painel de navegação à esquerda, clique em Roles (Funções) e, em seguida, clique em Create New Role (Criar nova função).
3. Insira exampleroles para o nome da função e, em seguida, clique em Next Step (Próxima etapa).
4. Em Select Role Type (Selecionar tipo de função), selecione Role for Cross-Account Access (Função para acesso entre contas) e, em seguida, clique no botão Select (Selecionar) ao lado de Provide access between AWS accounts you own (Fornecer acesso entre suas contas da AWS).
5. Insira o ID de conta da Conta C.

Para esta demonstração, não é necessário exigir que os usuários tenham autenticação multifator (MFA) para assumirem a função, portanto, deixe essa opção desmarcada.

6. Clique em Next Step (Próxima etapa) para definir as permissões que serão associadas à função.
7. Selecione a caixa ao lado da política access-accountA-bucket que você criou e, em seguida, clique em Next Step (Próxima etapa).

A página Revisar será exibida para que você possa confirmar as configurações para a função antes de criá-la. Um item muito importante a observar nesta página é o link que você pode enviar aos usuários que precisem usar essa função. Os usuários que clicarem no link irão diretamente para a página Mudança de função com os campos ID da conta e Nome da função já preenchidos. Você também pode ver esse link mais tarde na página Resumo da função de qualquer função entre contas.

8. Depois de revisar a função, clique em Create Role (Criar função).

A função `examplerole` é exibida na lista de funções.

9. Clique no nome da função `examplerole`.
10. Selecione a guia Trust Relationships (Relacionamentos de confiança).
11. Clique em Show policy document (Mostrar documento de política) e verifique se a política de confiança mostrada corresponde à política a seguir.

A política de confiança a seguir estabelece a confiança com a Conta C, permitindo-lhe a ação `sts:AssumeRole`. Para obter mais informações, acesse [AssumeRole](#) no AWS Security Token Service API Reference.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::AccountC-ID:root"  
            },  
            "Action": "sts:AssumeRole"  
        }  
    ]  
}
```

12. Anote o Nome de recurso da Amazon (ARN) da função `examplerole` que você criou.

Depois, nas etapas a seguir, você anexará um política de usuário para permitir que um usuário do IAM assuma essa função e identificará a função com o valor de Nome de recurso da Amazon (ARN).

## Etapa 2: Faça as tarefas da Conta B

O `examplebucket` de propriedade da Conta A precisa de objetos de propriedade de outras contas. Nessa etapa, o administrador da Conta B faz upload de um objeto usando as ferramentas da linha de comando.

- Usando o comando de CLI da AWS put-object, faça upload de um objeto para `examplebucket`.

```
aws s3api put-object --bucket examplebucket --key HappyFace.jpg --body HappyFace.jpg --grant-full-control id="canonicalUserId-ofTheBucketOwner" --profile AccountBadmin
```

Observe o seguinte:

- O parâmetro `--Profile` especifica o perfil `AccountBadmin`, e assim o objeto é de propriedade da Conta B.
- O parâmetro `grant-full-control` concede ao proprietário do bucket permissão de pleno controle sobre o objeto conforme exigido pela política do bucket.
- O parâmetro `--body` identifica o arquivo de origem para fazer upload. Por exemplo, se o arquivo está na unidade C: de um computador Windows, você especifica `c:\HappyFace.jpg`.

## Etapa 3: Faça as tarefas da Conta C

Nas etapas anteriores, a Conta A já criou uma função, `examplerole`, estabelecendo a confiança com a Conta C. Isso permite que os usuários na Conta C acessem a Conta A. Nessa etapa, o administrador da Conta C cria um usuário (Dave) e lhe delega a permissão `sts:AssumeRole` recebida da Conta A. Isso permitirá que Dave assuma a `examplerole` e tenha acesso temporário à Conta A. A política de acesso que a Conta A anexou à função vai limitar o que Dave pode fazer ao acessar a Conta A — especificamente, obter objetos em `examplebucket`.

### Etapa 3.1: Crie um usuário na Conta C e delegue a permissão para assumir `examplerole`

1. Usando o URL de login de usuário do IAM para a Conta C, primeiro faça login no Console de gerenciamento da AWS como usuário AccountCadmin.
2. No console do IAM, crie um usuário Dave.

Para obter instruções, consulte [Criação de usuários do IAM \(Console de Gerenciamento da AWS\)](#) no Guia do usuário do IAM.

3. Anote as credenciais de Dave. Dave precisará dessas credenciais para assumir a função `examplerole`.
4. Crie uma política inline para o usuário Dave do IAM para delegar a Dave a permissão `sts:AssumeRole` na função `examplerole` da conta A.
  - a. No painel de navegação à esquerda, clique em Users (Usuários).
  - b. Clique no nome de usuário Dave.
  - c. Na página detalhes do usuário, selecione a guia Permissions (Permissões) e, em seguida, expanda a seção Inline Policies (Políticas em linha).
  - d. Escolha clique aqui (ou Create User Policy [Criar política de usuário]).
  - e. Clique em Custom Policy (Política personalizada) e, em seguida, clique em Select (Selecionar).
  - f. Insira um nome para a política no campo Policy Name (Nome da política).
  - g. Cole a seguinte política no campo Policy Document (Documento da política).

Você terá de atualizar a políticas fornecendo o ID da Conta A.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": ["sts:AssumeRole"],  
            "Resource": "arn:aws:iam::AccountA-ID:role/examplerole"  
        }  
    ]  
}
```

- h. Clique em Apply Policy (Aplicar política)
5. Salve as credenciais de Dave no arquivo de config da CLI da AWS, adicionando outro perfil, AccountCDave.

```
[profile AccountCDave]  
aws_access_key_id = UserDaveAccessKeyID  
aws_secret_access_key = UserDaveSecretAccessKey  
region = us-west-2
```

### Etapa 3.2: Assumir a função (`examplerole`) e acessar objetos

Agora Dave pode acessar objetos no bucket de propriedade da Conta A, desta forma:

- Dave primeiro assume a `examplerole` usando suas próprias credenciais. Isso retornará credenciais temporárias.
  - Usando as credenciais temporárias, Dave então acessará os objetos no bucket da Conta A.
1. No prompt de comando, execute o seguinte comando `assume-role` da CLI da AWS, usando o perfil `AccountCDave`.

Você terá de atualizar o valor de Nome de recurso da Amazon (ARN) no comando, fornecendo o ID da Conta A onde `examplerole` foi definido.

```
aws sts assume-role --role-arn arn:aws:iam::accountA-ID:role/examplerole --profile AccountCDave --role-session-name test
```

Em resposta, o AWS Security Token Service (STS) retorna credenciais de segurança temporárias (ID da chave de acesso, chave de acesso secreta e um token de sessão).
  2. Salve as credenciais de segurança temporárias no arquivo config da CLI da AWS no perfil `TempCred`.

```
[profile TempCred]
aws_access_key_id = temp-access-key-ID
aws_secret_access_key = temp-secret-access-key
aws_session_token = session-token
region = us-west-2
```

3. No prompt de comando, execute o comando a seguir da CLI da AWS para acessar objetos usando as credenciais temporárias. Por exemplo, o comando especifica a API do objeto `head` para recuperar os metadados do objeto para o objeto `HappyFace.jpg`.

```
aws s3api get-object --bucket examplebucket --key HappyFace.jpg SaveFileAs.jpg --profile TempCred
```

Como a política de acesso anexada a `examplerole` permite as ações, o Amazon S3 processa a solicitação. Você pode tentar qualquer outra ação em qualquer outro objeto no bucket.

Se tentar qualquer outra ação — por exemplo, `get-object-acl` —, você terá a permissão negada, porque a função não tem permissão para essa ação.

```
aws s3api get-object-acl --bucket examplebucket --key HappyFace.jpg --profile TempCred
```

Usamos o usuário Dave para assumir a função e acessar o objeto usando credenciais temporárias. Também poderia ser um aplicativo na Conta C que acessa objetos em `examplebucket`. O aplicativo pode obter credenciais de segurança temporárias, e a Conta C pode delegar permissão de aplicativo para assumir `examplerole`.

## Etapa 4: Limpeza

1. Depois de terminar os testes, você pode seguir uma das etapas a seguir para fazer a limpeza.
  - Faça login no Console de gerenciamento da AWS ([Console de gerenciamento da AWS](#)) usando as credenciais da Conta A e faça o seguinte:
    - No console do Amazon S3, remova a política de bucket anexada a `examplebucket`. Nas Propriedades do bucket, exclua a política na seção Permissions (Permissões).
    - Se o bucket foi criado para este exercício, no console do Amazon S3, exclua os objetos e, em seguida, exclua o bucket.

- No console do IAM, remova o `examplerole` que você criou na Conta A.
  - No console do IAM, remova o usuário AccountAadmin.
2. Faça login no Console de gerenciamento da AWS ([Console de gerenciamento da AWS](#)) usando as credenciais da Conta B. No console do IAM, exclua o usuário AccountBadmin.
  3. Faça login no Console de gerenciamento da AWS ([Console de gerenciamento da AWS](#)) usando as credenciais da Conta C. No console do IAM, exclua o usuário AccountCadmin e o usuário Dave.

## Recursos relacionados

- [Criar uma função para delegar permissões a um usuário do IAM](#) no Guia do usuário do IAM.
- [Tutorial: Delegar acesso em contas da AWS usando funções do IAM](#) no Guia do usuário do IAM.
- [Trabalhar com políticas](#) no Guia do usuário do IAM.

# Uso de políticas de bucket e políticas de usuário

## Tópicos

- [Visão geral da linguagem da política de acesso \(p. 326\)](#)
- [Exemplos de políticas de bucket \(p. 358\)](#)
- [Exemplos de política de usuário \(p. 367\)](#)

A política de bucket e a política de usuário são duas opções de política de acesso disponíveis para conceder permissão aos seus recursos do Amazon S3. As duas usam linguagem de política de acesso baseada em JSON. Os tópicos nesta seção descrevem os elementos da linguagem de política de chave, com ênfase em detalhes específicos do Amazon S3 e oferecem exemplos de política de bucket e usuário.

### Important

Recomendamos analisar primeiro os tópicos introdutórios que explicam os conceitos básicos e as opções disponíveis para gerenciar o acesso aos recursos do Amazon S3. Para obter mais informações, consulte [Introdução ao gerenciamento de permissões de acesso aos recursos do Amazon S3 \(p. 282\)](#).

## Visão geral da linguagem da política de acesso

Os tópicos nesta seção descrevem os elementos básicos usados em políticas de bucket e usuário conforme utilizados no Amazon S3. Para obter informações completas sobre linguagem de política, consulte os tópicos [Visão geral de políticas do IAM](#) e [Referência de política do IAM da AWS](#) no Guia do usuário do IAM.

### Note

As políticas de bucket são limitadas a 20 KB.

## Elementos comuns em uma política de acesso

No sentido mais básico, uma política contém os seguintes elementos:

- Recursos – buckets e objetos são os recursos do Amazon S3 para os quais você pode permitir ou negar permissões. Em uma política, você usa o nome de recurso da Amazon (ARN) para identificar o recurso.
- Ações – para cada recurso, o Amazon S3 oferece suporte a um conjunto de operações. Você identifica as operações de recurso que permitirá (ou negará) usando palavras-chave de ação (consulte [Especificação de permissões em uma política \(p. 330\)](#)).

Por exemplo, a permissão s3 : ListBucket concede permissão de usuário à operação [GET Bucket \(Indicar objetos\)](#) do Amazon S3.

- Efeito – qual será o efeito quando o usuário solicitar a ação específica — pode ser permitir ou negar.

Se você não conceder (permitir) explicitamente acesso a um recurso, o acesso estará implicitamente negado. Você também pode negar explicitamente o acesso a um recurso, o que pode fazer para ter a certeza de que um usuário não consiga acessá-lo, mesmo que uma política diferente conceda acesso.

- Principal – a conta ou usuário que tem permissão de acesso a ações e recursos na declaração. Em uma política de bucket, a entidade principal é o usuário, a conta, o serviço ou outro destinatário que receba essa permissão.

O exemplo de política de bucket a seguir mostra os elementos comuns de política anteriores. A política permite que Dave, um usuário na conta [ID de conta](#), tenha as permissões s3 : GetObject, s3 : GetBucketLocation e s3 : ListBucket do Amazon S3 no bucket examplebucket.

```
{  
    "Version": "2012-10-17",  
    "Id": "ExamplePolicy01",  
    "Statement": [  
        {  
            "Sid": "ExampleStatement01",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::Account-ID:user/Dave"  
            },  
            "Action": [  
                "s3:GetObject",  
                "s3:GetBucketLocation",  
                "s3>ListBucket"  
            ],  
            "Resource": [  
                "arn:aws:s3::::examplebucket/*",  
                "arn:aws:s3::::examplebucket"  
            ]  
        }  
    ]  
}
```

Para obter mais informações sobre os elementos de política de acesso, consulte os seguintes tópicos:

- [Especificação de recursos em uma política \(p. 327\)](#)
- [Especificação de um principal em uma política \(p. 329\)](#)
- [Especificação de permissões em uma política \(p. 330\)](#)
- [Especificação de condições em uma política \(p. 335\)](#)

Os tópicos a seguir oferecem exemplos adicionais de política:

- [Exemplos de políticas de bucket \(p. 358\)](#)
- [Exemplos de política de usuário \(p. 367\)](#)

## Especificação de recursos em uma política

Veja a seguir o formato do nome de recurso da Amazon (ARN) comum para identificar quaisquer recursos na AWS.

```
arn:partition:service:region:namespace:relative-id
```

Para seus recursos do Amazon S3:

- aws é um nome de partição comum. Se os recursos estiverem na região China (Pequim), aws-cn será o nome da partição.
- s3 é o serviço.
- A região e o namespace não são especificados.
- Para o Amazon S3, pode ser um bucket-name ou um bucket-name/object-key. Você pode usar um curinga.

Em seguida, o formato do ARN para recursos do Amazon S3 se reduz a:

```
arn:aws:s3::::bucket_name
```

```
arn:aws:s3:::bucket_name/key_name
```

Veja a seguir exemplos de ARNs de recurso do Amazon S3.

- Este ARN identifica o objeto `/developers/design_info.doc` no bucket `examplebucket`.

```
arn:aws:s3:::examplebucket/developers/design_info.doc
```

- Você pode usar curingas como parte do ARN do recurso. É possível usar caracteres curinga (\*) e (?) em qualquer segmento de ARN (partes separadas por dois pontos). Um asterisco (\*) representa qualquer combinação de zero ou mais caracteres, e um ponto de interrogação (?) representa qualquer caractere único. É possível usar vários caracteres \* ou ? em cada segmento, mas um curinga não pode abranger segmentos.

- Este ARN usa o curinga \* na parte do ARN relativa ao ID para identificar todos os objetos no bucket `examplebucket`.

```
arn:aws:s3:::examplebucket/*
```

Este ARN usa \* para indicar todos os recursos do Amazon S3 (todos os buckets e objetos do S3 em sua conta).

```
arn:aws:s3:::*
```

- Este ARN usa os curingas \* e ? na parte relativa ao ID. Ele identifica todos os objetos em buckets, como `example1bucket`, `example2bucket`, `example3bucket` e assim por diante.

```
arn:aws:s3:::example?bucket/*
```

- Você pode usar variáveis de política em ARNs do Amazon S3. No momento da avaliação da política, essas variáveis predefinidas são substituídas pelos valores correspondentes. Vamos supor que você organize seu bucket como um conjunto de pastas, sendo uma pasta para cada um dos seus usuários. O nome da pasta é igual ao nome do usuário. Para conceder aos usuários permissão às pastas, você pode especificar uma variável de política no ARN do recurso:

```
arn:aws:s3:::bucket_name/developers/${aws:username}/
```

Em tempo de execução, quando a política é avaliada, a variável `${aws:username}` no ARN do recurso é substituída pelo nome do usuário que faz a solicitação.

Para encontrar o ARN de um bucket do S3, consulte as páginas de permissões Bucket Policy (Política de bucket) ou CORS configuration (Configuração CORS) do console do Amazon S3. Para obter mais informações, consulte [Como eu faço para adicionar uma política de bucket do S3?](#) ou [Como ativo o compartilhamento de recursos entre domínios com CORS?](#) no Guia do usuário do console do Amazon Simple Storage Service.

Para obter mais informações sobre ARNs, consulte:

- [Recurso](#) no Guia do usuário do IAM
- [Visão geral de variáveis de política do IAM](#) no Guia do usuário do IAM
- [ARNs](#) na AWS General Reference

Para obter mais informações sobre outros elementos de linguagem de políticas de acesso, consulte [Visão geral da linguagem da política de acesso \(p. 326\)](#).

## Especificação de um principal em uma política

O elemento `Principal` especifica o usuário, a conta, o serviço ou outra entidade que tem o acesso permitido ou negado a um recurso. Veja a seguir exemplos de especificação de `Principal`. Para obter mais informações, consulte [Principal](#) no Guia do usuário do IAM.

- Para conceder permissões para uma conta da AWS, identifique a conta usando o seguinte formato.

```
"AWS" : "account-ARN"
```

Por exemplo:

```
"Principal": {"AWS": "arn:aws:iam::AccountNumber-WithoutHyphens:root"}
```

O Amazon S3 também oferece suporte a um ID de usuário canônico, que é uma forma complexa do ID de conta da AWS. Você pode especificar esse ID usando o formato a seguir.

```
"CanonicalUser": "64-digit-alphanumeric-value"
```

Por exemplo:

```
"Principal": {"CanonicalUser": "64-digit-alphanumeric-value"}
```

Para obter informações sobre como encontrar o ID de usuário canônico da conta, consulte [Encontrar o ID de usuário canônico da conta](#).

### Important

Quando você usa um ID de usuário canônico em uma política, o Amazon S3 pode alterar o ID canônico para o ID da conta da AWS correspondente. Isso afeta a política, pois os dois IDs identificam a mesma conta.

- Para conceder permissão para um usuário do IAM na sua conta, você deve fornecer um par de nome-valor `"AWS" : "user-ARN"`.

```
"Principal": {"AWS": "arn:aws:iam::account-number-without-hyphens:user/username"}
```

- Para conceder permissão a todos, também denominada acesso anônimo, defina o curinga, `"*"`, com o valor `Principal`. Por exemplo, se você configura seu bucket como um site, quer que todos os objetos no bucket sejam publicamente acessíveis. Os seguintes são equivalentes:

```
"Principal": "*"
```

```
"Principal": {"AWS": "*"}"
```

### Warning

Tenha cuidado ao conceder acesso anônimo ao bucket do S3. Quando você concede acesso anônimo, qualquer pessoa no mundo pode acessar seu bucket. É altamente recomendável que você nunca conceda nenhum tipo de acesso anônimo de gravação ao seu bucket do S3.

- Você pode solicitar que os usuários acessam o conteúdo do Amazon S3 usando URLs do Amazon CloudFront (em vez de URLs do Amazon S3). Para fazer isso, crie uma identidade de acesso de origem (OAI) do CloudFront e, em seguida, altere as permissões no seu bucket ou nos objetos no seu bucket. O formato para especificar a OAI em uma instância `Principal` é o seguinte:

```
"Principal": {"CanonicalUser": "Amazon S3 Canonical User ID assigned to origin access identity"}
```

Para obter mais informações, consulte [Usar uma identidade de acesso de origem para restringir o acesso ao conteúdo do Amazon S3](#) no Guia do desenvolvedor do Amazon CloudFront.

Para obter mais informações sobre outros elementos de linguagem de políticas de acesso, consulte [Visão geral da linguagem da política de acesso](#) (p. 326).

## Especificação de permissões em uma política

O Amazon S3 define um conjunto de permissões que você pode especificar em uma política. Elas são palavras-chave, cada uma mapeando operações específicas do Amazon S3 (consulte [Operações em buckets](#) e [Operações em objetos](#) no Amazon Simple Storage Service API Reference).

### Tópicos

- [Permissões para operações de objeto](#) (p. 330)
- [Permissões relacionadas a operações de bucket](#) (p. 331)
- [Permissões relacionadas a operações de sub-recurso de bucket](#) (p. 332)
- [Permissões relacionadas a operações de conta](#) (p. 334)

## Permissões para operações de objeto

Esta seção fornece uma lista de permissões para operações de objeto que você pode especificar em uma política.

### Permissões para operações de objeto do Amazon S3

Permissões	Operações do Amazon S3
s3:AbortMultipartUpload	<a href="#">Aplicar upload multipart</a>
s3:DeleteObject	<a href="#">Objeto DELETE</a>
s3:DeleteObjectTagging	<a href="#">DELETE atribuição de tags de objeto</a>
s3:DeleteObjectVersion	<a href="#">DELETE objeto (uma versão específica do objeto)</a>
s3:DeleteObjectVersionTagging	<a href="#">DELETE atribuição de tags de objeto (para uma versão específica do objeto)</a>
s3:GetObject	<a href="#">Objeto GET, Objeto HEAD, Conteúdo do objeto SELECT</a>  Quando você concede essa permissão a um bucket habilitado para versão, sempre obtém a última versão dos dados.
s3:GetObjectAcl	<a href="#">ACL de objeto GET</a>
s3:GetObjectTagging	<a href="#">GET atribuição de tags de objeto</a>
s3:GetObjectTorrent	<a href="#">GET torrent de objeto</a>
s3:GetObjectVersion	<a href="#">Objeto GET, Objeto HEAD</a>  Para conceder permissão para dados de objeto específicos da versão, você deve conceder esta permissão. Em outras palavras, quando você especifica um número de versão ao fazer essas solicitações, precisa dessa permissão do Amazon S3.

Permissões	Operações do Amazon S3
s3:GetObjectVersion <a href="#">GET ACL</a>	(para uma versão específica do objeto)
s3:GetObjectVersion <a href="#">GET Tagging</a>	(attribution de tags de objeto (para uma versão específica do objeto))
s3:GetObjectVersion <a href="#">GET Torrent</a>	(download de torrent de objeto)
s3>ListMultipartUploads <a href="#">ListUploadParts</a>	
s3:PutObject	PUT objeto, POST objeto, Iniciar multipart upload, Upload de parte, Concluir multipart upload, PUT objeto - Copiar
s3:PutObjectAcl	ACL de objeto PUT
s3:PutObjectTagging <a href="#">PUT Tagging</a>	(atribuição de tags de objeto)
s3:PutObjectVersionAcl <a href="#">PUT ACL</a>	(para uma versão específica do objeto)
s3:PutObjectVersionTagging <a href="#">PUT Tagging</a>	(para uma versão específica do objeto)
s3:RestoreObject	POST restauração de objeto

O seguinte exemplo de política de bucket concede as permissões s3:PutObject e s3:PutObjectAcl a um usuário (Dave). Se você remover o elemento Principal, poderá anexar a política a um usuário. São operações de objeto e, de acordo com isso, a porção relativa ao ID do ARN Resource identifica os objetos (examplebucket/\*). Para obter mais informações, consulte [Especificação de recursos em uma política \(p. 327\)](#).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "statement1",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::AccountB-ID:user/Dave"
            },
            "Action": ["s3:PutObject", "s3:PutObjectAcl"],
            "Resource": "arn:aws:s3:::examplebucket/*"
        }
    ]
}
```

Você pode usar um curinga para conceder permissão para todas as ações do Amazon S3.

```
"Action": "*"
```

## Permissões relacionadas a operações de bucket

Esta seção fornece uma lista de permissões relacionadas a operações de bucket que você pode especificar em uma política.

### Permissões do Amazon S3 relacionadas a operações de bucket

Palavras-chave de permissão	Operações do Amazon S3 cobertas
s3:CreateBucket	PUT Bucket

Palavras-chave de permissão	Operações do Amazon S3 cobertas
s3:DeleteBucket	<a href="#">DELETE bucket</a>
s3>ListBucket	<a href="#">GET bucket (listar objetos), HEAD bucket</a>
s3>ListBucketVersions	<a href="#">GET versões de objeto do bucket</a>
s3>ListAllMyBuckets	<a href="#">Serviço GET</a>
s3>ListBucketMultipartUploads	<a href="#">Listar uploads</a>

O exemplo a seguir de política de usuário concede as permissões s3:CreateBucket, s3>ListAllMyBuckets e s3:GetBucketLocation a um usuário. Observe que, para todas essas permissões, você define a parte relativa ao ID do ARN Resource como "\*". Para todas as outras ações de bucket, você deve especificar um nome de bucket. Para obter mais informações, consulte [Especificação de recursos em uma política \(p. 327\)](#).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "statement1",
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket", "s3>ListAllMyBuckets", "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::*"
            ]
        }
    ]
}
```

Se seu usuário usar o console para exibir buckets e visualizar o conteúdo de qualquer um desses buckets, o usuário deve ter as permissões s3>ListAllMyBuckets e s3:GetBucketLocation. Para obter um exemplo, consulte "Política de acesso ao console" em [Escrever políticas do IAM: como conceder acesso a um bucket do Amazon S3](#).

## Permissões relacionadas a operações de sub-recurso de bucket

Esta seção fornece uma lista de permissões relacionadas a operações de sub-recurso de bucket que você pode especificar em uma política.

### Permissões do Amazon S3 relacionadas a operações de sub-recurso de bucket

Permissões	Operações do Amazon S3 cobertas
s3>DeleteBucketPolicy	<a href="#">DELETE política de bucket</a>
s3>DeleteBucketWebsite	<a href="#">Site de DELETE Bucket</a>
s3>GetAccelerateConfiguration	<a href="#">GET Aceleração do bucket</a>
s3>GetAnalyticsConfiguration	<a href="#">GET análise de bucket, Listar configurações de análise de bucket</a>
s3>GetBucketAcl	<a href="#">GET Bucket acl</a>

Permissões	Operações do Amazon S3 cobertas
s3:GetBucketCORS	Cors de GET Bucket
s3:GetBucketLocation	GET localização do bucket
s3:GetBucketLogging	Registro de GET Bucket
s3:GetBucketNotification	Notificação de GET Bucket
s3:GetBucketPolicy	GET política de bucket
s3:GetBucketPolicyStatus	GET BucketPolicyStatus
s3:GetBucketPublicAccessBlock	GET PublicAccessBlock
s3:GetBucketRequestPayment	GET requestPayment de bucket
s3:GetBucketTagging	GET marcação de bucket
s3:GetBucketVersioning	Versionamento de GET Bucket
s3:GetBucketWebsite	GET em site de bucket
s3:GetEncryptionConfiguration	GET Bucket encryption
s3:GetInventoryConfiguration	GET Bucket inventory, Listar configurações de inventário de bucket
s3:GetLifecycleConfiguration	Ciclo de vida de GET Bucket
s3:GetMetricsConfiguration	GET Bucket metrics, Listar configurações de métricas de bucket
s3:GetReplicationConfiguration	Replicação do GET Bucket
s3:PutAccelerateConfiguration	PUT Bucket accelerate
s3:PutAnalyticsConfiguration	PUT Bucket analytics, DELETE Bucket analytics
s3:PutBucketAcl	acl de PUT Bucket
s3:PutBucketCORS	PUT bucket cors, DELETE bucket cors
s3:PutBucketLogging	PUT registro de bucket
s3:PutBucketNotification	Notificação de PUT Bucket
s3:PutBucketPolicy	PUT política de bucket
s3:PutBucketPublicAccessBlock	PUT PublicAccessBlock, DELETE PublicAccessBlock
s3:PutBucketRequestPayment	PUT requestPayment de bucket
s3:PutBucketTagging	DELETE atribuição de tags de bucket, PUT atribuição de tags de bucket
s3:PutBucketVersioning	Versionamento de PUT Bucket
s3:PutBucketWebsite	PUT em site de bucket
s3:PutEncryptionConfiguration	PUT Bucket encryption, DELETE Bucket encryption

Permissões	Operações do Amazon S3 cobertas
s3:PutInventoryConfiguration	PUT inventário de bucket, DELETE inventário de bucket
s3:PutLifecycleConfiguration	PUT Bucket lifecycle, DELETE Bucket lifecycle
s3:PutMetricsConfiguration	PUT métricas de bucket, DELETE métricas de bucket
s3:PutReplicationConfiguration	PUT Bucket replication, DELETE Bucket replication

O seguinte exemplo de política de usuário concede a permissão s3:GetBucketAcl no bucket examplebucket ao usuário Dave.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "statement1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::Account-ID:user/Dave"
      },
      "Action": [
        "s3:GetObjectVersion",
        "s3:GetBucketAcl"
      ],
      "Resource": "arn:aws:s3:::examplebucket"
    }
  ]
}
```

Você pode excluir objetos chamando explicitamente a API DELETE Object ou configurando seu ciclo de vida (consulte [Gerenciamento do ciclo de vida de objetos \(p. 122\)](#)) de modo que o Amazon S3 possa remover os objetos quando seu ciclo de vida expirar. Para bloquear explicitamente usuários ou contas para exclusão de objetos, você deve negar explicitamente as permissões s3>DeleteObject, s3>DeleteObjectVersion e s3:PutLifecycleConfiguration. Por padrão, os usuários não têm nenhuma permissão. Mas, à medida que você cria usuários, adiciona usuários a grupos e concede permissões, é possível que os usuários obtenham certas permissões que você não pretendia conceder. É aí que você pode usar a negação explícita, que se sobrepõe a todas as outras permissões que um usuário pode ter e nega ao usuário permissões para ações específicas.

## Permissões relacionadas a operações de conta

Esta seção fornece uma lista de permissões relacionadas a operações de conta que você pode especificar em uma política.

### Permissões do Amazon S3 relacionadas a operações de conta

Palavras-chave de permissão	Operações do Amazon S3 cobertas
s3:GetAccountPublicAccessBlock	GET PublicAccessBlock
s3:PutAccountPublicAccessBlock	PUT PublicAccessBlock, DELETE PublicAccessBlock

O exemplo a seguir de política de usuário concede a permissão s3:GetAccountPublicAccessBlock a um usuário. Para todas essas permissões você define o valor Resource como "\*". Para obter mais informações, consulte [Especificação de recursos em uma política \(p. 327\)](#).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "statement1",  
            "Effect": "Allow",  
            "Action": [  
                "s3:GetAccountPublicAccessBlock"  
            ],  
            "Resource": [  
                "*"  
            ]  
        }  
    ]  
}
```

## Especificação de condições em uma política

A linguagem de políticas de acesso permite que você especifique condições ao conceder permissões. O elemento `Condition` (ou o bloco `Condition`) permite que você especifique as condições para quando uma política está em vigor. No elemento `Condition`, que é opcional, você cria expressões em que usa operadores booleanos (`equal`, `less than`, etc.) para fazer a correspondência da sua condição com os valores na solicitação. Por exemplo, ao conceder a um usuário permissão para fazer upload de um objeto, o proprietário do bucket pode exigir que o objeto seja publicamente legível adicionando a condição `StringEquals` conforme mostrado aqui:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "statement1",  
            "Effect": "Allow",  
            "Action": [  
                "s3:PutObject"  
            ],  
            "Resource": [  
                "arn:aws:s3:::examplebucket/*"  
            ],  
            "Condition": {  
                "StringEquals": {  
                    "s3:x-amz-acl": [  
                        "public-read"  
                    ]  
                }  
            }  
        }  
    ]  
}
```

O bloco `Condition` especifica a condição `StringEquals` que é aplicada ao par de chave-valor especificado, `"s3:x-amz-acl": ["public-read"]`. Existe um conjunto de chaves predefinidas que você pode usar para expressar uma condição. O exemplo usa a chave de condição `s3:x-amz-acl`. Essa condição exige que o usuário inclua o cabeçalho `x-amz-acl` com o valor `public-read` em toda solicitação PUT objeto.

Para obter mais informações sobre como especificar condições em uma linguagem de política de acesso, consulte [Condição](#) no Guia do usuário do IAM.

Os tópicos a seguir descrevem as chaves de condição da AWS e específicas do Amazon S3– e fornecem exemplos de políticas.

## Tópicos

- [Chaves de condição disponíveis \(p. 336\)](#)
- [Chaves de condição do Amazon S3 para operações de objeto \(p. 338\)](#)
- [Chaves de condição do Amazon S3 para operações de bucket \(p. 351\)](#)

## Chaves de condição disponíveis

As chaves predefinidas disponíveis para especificação de condições em uma política de acesso do Amazon S3 podem ser classificadas assim:

- Chaves de toda a AWS – A AWS fornece um conjunto de chaves comuns que recebe o suporte de todos serviços da AWS que, por sua vez, dão suporte às políticas. Essas chaves comuns a todos serviços são chamadas de chaves da AWS e usam o prefixo `aws:`. Para obter uma lista completa das chaves de toda a AWS, consulte [Chaves disponíveis para condições no Guia do usuário do IAM](#). Também há chaves específicas do Amazon S3, que usam o prefixo `s3:`. As chaves específicas do Amazon S3 são abordadas no próximo item.

As novas chaves de condição `aws:sourceVpce` e `aws:sourceVpc` são usadas em políticas de bucket para VPC endpoints. Para ver exemplos de uso dessas chaves de condição, consulte [Exemplo de políticas de bucket para VPC endpoints para o Amazon S3 \(p. 365\)](#).

O exemplo de política de bucket a seguir concede a usuários autenticados permissão para usar a ação `s3:GetObject`, se a solicitação for proveniente de um intervalo específico de endereços IP (192.168.143.\*), a menos que o endereço IP seja 192.168.143.188. No bloco de condição, `IpAddress` e `NotIpAddress` são condições, e cada condição recebe um par chave-valor para avaliação. Neste exemplo os dois pares de chave-valor usam a chave da AWS `aws:SourceIp`.

### Note

Os valores de chave `IpAddress` e `NotIpAddress` especificados na condição usam notação CIDR conforme descrito na RFC 4632. Para obter mais informações, acesse <http://www.rfc-editor.org/rfc/rfc4632.txt>.

```
{  
    "Version": "2012-10-17",  
    "Id": "S3PolicyId1",  
    "Statement": [  
        {  
            "Sid": "statement1",  
            "Effect": "Allow",  
            "Principal": "*",  
            "Action": ["s3:GetObject"],  
            "Resource": "arn:aws:s3:::examplebucket/*",  
            "Condition": {  
                "IpAddress": {  
                    "aws:SourceIp": "192.168.143.0/24"  
                },  
                "NotIpAddress": {  
                    "aws:SourceIp": "192.168.143.188/32"  
                }  
            }  
        }  
    ]  
}
```

- Chaves específicas do Amazon S3 – além das chaves de toda a AWS, as chaves de condição a seguir são aplicáveis somente no contexto de concessão de permissões específicas do Amazon S3. Essas chaves específicas do Amazon S3 usam o prefixo s3:.
  - s3:x-amz-acl
  - s3:x-amz-copy-source
  - s3:x-amz-metadata-directive
  - s3:x-amz-server-side-encryption
  - s3:VersionId
  - s3:LocationConstraint
  - s3:delimiter
  - s3:max-keys
  - s3:prefix
  - s3:x-amz-server-side-encryption-aws-kms-key-id
  - s3:ExistingObjectTag/*<tag-key>*

Para ver exemplos de política que usam chaves de condição relacionadas a tags de objeto, consulte [Marcação de objetos e políticas de controle de acesso \(p. 116\)](#).

- s3:RequestObjectTagKeys
- s3:RequestObjectTag/*<tag-key>*

Por exemplo, a política de bucket a seguir concederá a permissão s3:PutObject para duas contas da AWS se a solicitação incluir o cabeçalho x-amz-acl tornando o objeto publicamente legível.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AddCannedAcl",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": ["arn:aws:iam::account1-ID:root", "arn:aws:iam::account2-ID:root"]  
            },  
            "Action": ["s3:PutObject"],  
            "Resource": ["arn:aws:s3:::examplebucket/*"],  
            "Condition": {  
                "StringEquals": {  
                    "s3:x-amz-acl": ["public-read"]  
                }  
            }  
        }  
    ]  
}
```

O bloco Condition usa a condição StringEquals e recebe um par de chave/valor, "s3:x-amz-acl": ["public-read"], para avaliação. No par de chave-valor, s3:x-amz-acl é uma chave específica do Amazon S3, conforme indicado pelo prefixo s3:.

#### Important

Nem todas as condições fazem sentido para todas as ações. Por exemplo, faz sentido incluir uma condição s3:LocationConstraint em uma política que conceda a permissão s3:CreateBucket Amazon S3, mas não a permissão s3:GetObject. O Amazon S3 pode testar a existência de erros semânticos desse tipo que envolvam condições específicas do

Amazon S3. Contudo, se você estiver criando uma política para um usuário do IAM e incluir uma condição do Amazon S3 semanticamente inválida, nenhum erro será reportado porque o IAM não pode validar condições do Amazon S3.

A seção a seguir descreve as chaves de condição que podem ser usadas para conceder permissão condicional para operações de bucket e objeto. Além disso, existem chaves de condição relacionadas à autenticação do Signature versão 4 do Amazon S3. Para obter mais informações, acesse [Chaves de política específicas de autenticação do Signature versão 4 do Amazon S3](#) no Amazon Simple Storage Service API Reference.

## Chaves de condição do Amazon S3 para operações de objeto

A tabela a seguir mostra quais condições do Amazon S3 você pode usar com quais ações do Amazon S3. Políticas de exemplo são fornecidas na tabela a seguir. Observe o seguinte sobre as chaves de condição específicas do Amazon S3 descritas na tabela a seguir:

- Os nomes de chave de condição são precedidos pelo prefixo `s3::`. Por exemplo, `s3:x-amz-acl`
- Cada chave de condição mapeia para o mesmo cabeçalho de solicitação de nome permitido pela API na qual a condição pode ser definida. Essas chaves de condição ditam o comportamento dos mesmos cabeçalhos de solicitação de nome. Por exemplo:
  - A chave de condição `s3:x-amz-acl` que você usa para conceder permissão de condição para a permissão `s3:PutObject` define o comportamento do cabeçalho de solicitação `x-amz-acl` para o qual a API PUT objeto oferece suporte.
  - A chave de condição `s3:VersionId` que você usa para conceder permissão condicional para a permissão `s3:GetObjectVersion` define o comportamento do parâmetro de consulta `versionId` que você define em uma solicitação de objeto GET.

Permissão	Chaves de condição aplicáveis (ou palavras-chave)	Descrição
<code>s3:PutObject</code>	<ul style="list-style-type: none"><li>• <code>s3:x-amz-acl</code> (para permissões de ACL padrão)</li><li>• <code>s3:x-amz-grant-permission</code> (para permissões explícitas), onde a <code>permissão</code> pode ser: <code>read, write, read-acp, write-acp, full-control</code></li></ul>	<p>A operação <a href="#">PUT Object</a> permite cabeçalhos específicos da lista de controle de acesso que você usa para conceder permissões baseadas em ACL. Usando essas chaves, o proprietário do bucket pode definir uma condição para exigir permissões de acesso específicas quando o usuário faz upload de um objeto.</p> <p>Para ver um exemplo de política, consulte <a href="#">Exemplo 1: Concessão da permissão s3:PutObject com uma condição que exige que o proprietário do bucket obtenha controle total (p. 346)</a>.</p> <p>Para obter mais informações sobre ACLs, consulte <a href="#">Visão geral da Lista de controle de acesso (ACL) (p. 390)</a>.</p>

Permissão	Chaves de condição aplicáveis (ou palavras-chave)	Descrição
	<code>s3:x-amz-copy-source</code>	<p>Para copiar um objeto, use a API PUT Object (consulte <a href="#">PUT Object</a>) e especifique a origem usando o cabeçalho <code>x-amz-copy-source</code>. Usando essa chave, o proprietário do bucket pode restringir a origem da cópia a um bucket específico, uma pasta específica no bucket ou um objeto específico em um bucket.</p> <p>Para ver um exemplo de política, consulte <a href="#">Exemplo 3: Concessão da permissão s3:PutObject para copiar objetos com uma restrição na origem da cópia (p. 349)</a>.</p>
	<code>s3:x-amz-server-side-encryption</code>	<p>Ao fazer upload de um objeto, você pode usar o cabeçalho <code>x-amz-server-side-encryption</code> para solicitar que o Amazon S3 criptografe o objeto quando ele é salvo, usando uma chave de criptografia de envelope gerenciada pelo AWS Key Management Service (AWS KMS) ou pelo Amazon S3 (consulte <a href="#">Proteção de dados usando criptografia no lado do servidor (p. 410)</a>).</p> <p>Ao conceder a permissão <code>s3:PutObject</code>, o proprietário do bucket pode adicionar uma condição usando essa chave para exigir que o usuário especifique esse cabeçalho na solicitação. Um proprietário de bucket pode conceder tal permissão condicional para garantir que os objetos que o usuário carrega sejam criptografados ao serem salvos.</p> <p>Para ver um exemplo de política, consulte <a href="#">Exemplo 1: Concessão da permissão s3:PutObject com uma condição que exige que o proprietário do bucket obtenha controle total (p. 346)</a>.</p>

Permissão	Chaves de condição aplicáveis (ou palavras-chave)	Descrição
	<b>s3:x-amz-server-side-encryption-aws-kms-key-id</b>	<p>Ao fazer upload de um objeto, você pode usar o cabeçalho <code>x-amz-server-side-encryption-aws-kms-key-id</code> para solicitar que o Amazon S3 criptografe o objeto usando a chave do AWS KMS especificada ao salvar o objeto (consulte <a href="#">Proteção de dados usando criptografia do servidor com chaves gerenciadas pelo AWS KMS–(SSE-KMS) (p. 410)</a>).</p> <p>Ao conceder a permissão <code>s3:PutObject</code>, o proprietário do bucket pode adicionar uma condição usando essa chave para restringir o ID da chave do AWS KMS usado para a criptografia do objeto com um valor específico.</p> <p>Um proprietário de bucket pode conceder tal permissão condicional para garantir que os objetos que o usuário carrega sejam criptografados com uma chave específica o serem salvos.</p> <p>A chave do AWS KMS especificada na política deve usar o seguinte formato:</p> <pre>arn:aws:kms:<b>region:acct-id</b>:key/<b>key-id</b></pre>
	<b>s3:x-amz-metadata-directive</b>	<p>Ao copiar um objeto usando a API PUT Object (consulte <a href="#">PUT Object</a>), você pode adicionar o cabeçalho <code>x-amz-metadata-directive</code> para especificar se os metadados do objeto serão copiados do objeto de origem ou substituídos pelos metadados fornecidos na solicitação.</p> <p>Usando esse bucket de chave, um proprietário pode adicionar uma condição para impor certo comportamento quando objetos são carregados.</p> <p>Valores válidos: <code>COPY   REPLACE</code>. O padrão é <code>COPY</code>.</p>

Permissão	Chaves de condição aplicáveis (ou palavras-chave)	Descrição
	<b>s3:x-amz-storage-class</b>	<p>Por padrão, <code>s3:PutObject</code> armazena os objetos usando a classe de armazenamento STANDARD, mas você pode usar o cabeçalho de solicitação <code>x-amz-storage-class</code> para especificar uma classe de armazenamento diferente.</p> <p>Ao conceder a permissão <code>s3:PutObject</code>, você pode usar a chave de condição <code>s3:x-amz-storage-class</code> para restringir qual classe de armazenamento usar ao armazenar objetos carregados. Para obter mais informações sobre classes de armazenamento, consulte <a href="#">Classes de armazenamento</a>.</p> <p>Para ver um exemplo de política, consulte <a href="#">Exemplo 5: Restringir uploads de objetos com uma classe de armazenamento específica</a> (p. 351).</p> <p>Para obter os valores válidos, consulte <a href="#">Solicitações de objeto PUT do Amazon S3</a>.</p>
	<ul style="list-style-type: none"> <li>• <code>s3:RequestObjectTagKeys</code></li> <li>• <code>s3:RequestObjectTag/&lt;tag-key&gt;</code></li> </ul>	<p>Usando essa chave de condição, você pode restringir a permissão para a ação <code>s3:PutObject</code> limitando as tags de objeto permitidas na solicitação. Para ver exemplos de uso dessas chaves de condição, consulte <a href="#">Marcação de objetos e políticas de controle de acesso</a> (p. 116).</p>

Permissão	Chaves de condição aplicáveis (ou palavras-chave)	Descrição
s3:PutObjectAcl	<ul style="list-style-type: none"> <li>• s3:x-amz-acl (para permissões de ACL padrão)</li> <li>• s3:x-amz-grant-permission (para permissões explícitas), onde a <b>permissão</b> pode ser:  read, write, read-acp, write-acp, grant-full-control</li> </ul>	<p>A API <a href="#">ACL de objeto PUT</a> define a lista de controle de acesso no objeto especificado. A operação oferece suporte a cabeçalhos relacionados a ACL. Ao conceder essa permissão, o proprietário do bucket pode adicionar condições usando essas chaves para exigir certas permissões. Para obter mais informações sobre ACLs, consulte <a href="#">Visão geral da Lista de controle de acesso (ACL) (p. 390)</a>.</p> <p>Por exemplo, o proprietário do bucket pode querer manter controle do objeto independentemente de quem seja proprietário do objeto. Para fazer isso, o proprietário do bucket pode adicionar uma condição usando uma destas chaves para exigir que o usuário inclua permissões específicas para o proprietário do bucket.</p>
	s3:ExistingObjectTag/ <tag-key>	Usando essa chave de condição, você pode restringir a permissão para a ação <a href="#">s3:PutObjectAcl</a> somente para objetos que tenham uma chave e um valor de tag específicos. Para ver exemplos, consulte <a href="#">Marcação de objetos e políticas de controle de acesso (p. 116)</a> .
s3:PutObjectTagging	<ul style="list-style-type: none"> <li>• s3:RequestObjectTagKeys</li> <li>• s3:RequestObjectTag/ &lt;tag-key&gt;</li> </ul>	Usando essa chave de condição, você pode restringir a permissão para a ação <a href="#">s3:PutObjectTagging</a> limitando as tags de objeto permitidas na solicitação. Para ver exemplos de uso dessas chaves de condição, consulte <a href="#">Marcação de objetos e políticas de controle de acesso (p. 116)</a> .
	s3:ExistingObjectTag/ <tag-key>	Usando essa chave de condição, você pode restringir a permissão somente para objetos que tenham uma chave e um valor de tag específicos. Para ver exemplos, consulte <a href="#">Marcação de objetos e políticas de controle de acesso (p. 116)</a> .

Permissão	Chaves de condição aplicáveis (ou palavras-chave)	Descrição
s3:PutObjectVersionTagging	<ul style="list-style-type: none"> <li>• s3:RequestObjectTagKeys</li> <li>• s3:RequestObjectTag/&lt;tag-key&gt;</li> </ul>	Usando essa chave de condição, você pode restringir a permissão para a ação <code>s3:PutObjectVersionTagging</code> limitando as tags de objeto permitidas na solicitação. Para ver exemplos de uso dessas chaves de condição, consulte <a href="#">Marcação de objetos e políticas de controle de acesso (p. 116)</a> .
	s3:VersionId	Usando essa chave de condição, você pode restringir a permissão para a ação <code>s3:PutObjectVersionTagging</code> para uma versão específica do objeto. Para ver um exemplo de política, consulte <a href="#">Exemplo 4: Concessão de acesso a uma versão específica de um objeto (p. 350)</a> .
	s3:ExistingObjectTag/<tag-key>	Usando essa chave de condição, você pode restringir a permissão somente para objetos que tenham uma chave e um valor de tag específicos. Para ver exemplos, consulte <a href="#">Marcação de objetos e políticas de controle de acesso (p. 116)</a> .
s3:GetObjectVersion	s3:VersionId	<p>Esta permissão do Amazon S3 permite que o usuário execute um conjunto de operações de API do Amazon S3 (consulte <a href="#">Permissões para operações de objeto do Amazon S3 (p. 330)</a>). Para um bucket habilitado para versão, você pode especificar a versão do objeto para a qual recuperar dados.</p> <p>Adicionando uma condição usando essa chave, o proprietário do bucket pode restringir o acesso do usuário a dados de apenas uma versão específica do objeto. Para ver um exemplo de política, consulte <a href="#">Exemplo 4: Concessão de acesso a uma versão específica de um objeto (p. 350)</a>.</p>
	s3:ExistingObjectTag/<tag-key>	Usando essa chave de condição, você pode restringir a permissão somente para objetos que tenham uma chave e um valor de tag específicos. Para ver exemplos, consulte <a href="#">Marcação de objetos e políticas de controle de acesso (p. 116)</a> .

Permissão	Chaves de condição aplicáveis (ou palavras-chave)	Descrição
s3:GetObject	s3:ExistingObjectTag/ <tag-key>	Usando essa chave de condição, você pode restringir a permissão somente para objetos que tenham uma chave e um valor de tag específicos. Para ver exemplos, consulte <a href="#">Marcação de objetos e políticas de controle de acesso (p. 116)</a> .
s3:GetObjectAcl	s3:ExistingObjectTag/ <tag-key>	Usando essa chave de condição, você pode restringir a permissão somente para objetos que tenham uma chave e um valor de tag específicos. Para ver exemplos, consulte <a href="#">Marcação de objetos e políticas de controle de acesso (p. 116)</a> .
s3:GetObjectVersionAcl	s3:VersionId	<p>Você pode recuperar a lista de controle de acesso (ACL) de uma versão específica do objeto usando a API <a href="#">GET Object acl</a>. O usuário deve ter a permissão para a ação s3:GetObjectVersionAcl. Para um bucket habilitado para versão, essa permissão do Amazon S3 permite que um usuário obtenha a ACL de uma versão específica do objeto.</p> <p>O proprietário do bucket pode adicionar uma condição usando a chave para restringir o usuário a uma versão específica do objeto.</p>
	s3:ExistingObjectTag/ <tag-key>	Usando essa chave de condição, você pode restringir a permissão somente para objetos que tenham uma chave e um valor de tag específicos. Para ver exemplos, consulte <a href="#">Marcação de objetos e políticas de controle de acesso (p. 116)</a> .
s3:PutObjectVersionAcl	s3:VersionId	Para um bucket habilitado para versão, você pode especificar a versão do objeto na solicitação <a href="#">ACL de objeto PUT</a> para definir a ACL em uma versão específica do objeto. Usando essa condição, o proprietário do bucket pode restringir o usuário para definir uma ACL apenas em uma versão de um objeto.

Permissão	Chaves de condição aplicáveis (ou palavras-chave)	Descrição
<code>s3:DeleteObjectVersion</code>	<ul style="list-style-type: none"> <li><code>s3:x-amz-acl</code> (para permissões de ACL padrão)</li> <li><code>s3:x-amz-grant-permission</code> (para permissões explícitas), onde a <i>permissão</i> pode ser: <code>read, write, read-acp, write-acp, grant-full-control</code></li> </ul>	<p>Para um bucket habilitado para versão, essa permissão do Amazon S3 permite que você defina uma ACL em uma versão específica do objeto.</p> <p>Para ver uma descrição dessas chaves de condição, consulte a permissão <code>s3:PutObjectACL</code> nesta tabela.</p>
	<code>s3:ExistingObjectTag/&lt;tag-key&gt;</code>	Usando essa chave de condição, você pode restringir a permissão somente para objetos que tenham uma chave e um valor de tag específicos. Para ver exemplos, consulte <a href="#">Marcação de objetos e políticas de controle de acesso (p. 116)</a> .
<code>s3:DeleteObjectVersion</code>	<code>s3:VersionId</code>	<p>Para um bucket habilitado para versão, essa permissão do Amazon S3 permite que o usuário exclua uma versão específica do objeto.</p> <p>O proprietário do bucket pode adicionar uma condição usando essa chave para limitar a capacidade do usuário de excluir apenas uma versão específica do objeto.</p> <p>Para ver um exemplo de uso dessa chave de condição, consulte <a href="#">Exemplo 4: Concessão de acesso a uma versão específica de um objeto (p. 350)</a>. O exemplo trata da concessão da ação <code>s3:GetObjectVersion</code>, mas a política mostra o uso dessa chave de condição.</p>
<code>s3:DeleteObjectTagging</code>	<code>s3:ExistingObjectTag/&lt;tag-key&gt;</code>	Usando essa chave de condição, você pode restringir a permissão somente para objetos que tenham uma chave e um valor de tag específicos. Para ver exemplos, consulte <a href="#">Marcação de objetos e políticas de controle de acesso (p. 116)</a> .

Permissão	Chaves de condição aplicáveis (ou palavras-chave)	Descrição
s3:DeleteObjectVersion	TagExistingObjectTag/ <tag-key>	Usando essa chave de condição, você pode restringir a permissão somente para objetos que tenham uma chave e um valor de tag específicos. Para ver exemplos, consulte <a href="#">Marcação de objetos e políticas de controle de acesso (p. 116)</a> .
	s3:VersionId	Usando essa chave de condição, você pode restringir a permissão para a ação s3:DeleteObjectVersionTagging para uma versão específica do objeto. Para ver um exemplo de política, consulte <a href="#">Exemplo 4: Concessão de acesso a uma versão específica de um objeto (p. 350)</a> .
s3:GetObjectTagging	s3:ExistingObjectTag/ <tag-key>	Usando essa chave de condição, você pode restringir a permissão somente para objetos que tenham uma chave e um valor de tag específicos. Para ver exemplos, consulte <a href="#">Marcação de objetos e políticas de controle de acesso (p. 116)</a> .
s3GetObjectVersionTagging	TagExistingObjectTag/ <tag-key>	Usando essa chave de condição, você pode restringir a permissão somente para objetos que tenham uma chave e um valor de tag específicos. Para ver exemplos, consulte <a href="#">Marcação de objetos e políticas de controle de acesso (p. 116)</a> .
	s3:VersionId	Usando essa chave de condição, você pode restringir a permissão para a ação s3:GetObjectVersionTagging para uma versão específica do objeto. Para ver um exemplo de política, consulte <a href="#">Exemplo 4: Concessão de acesso a uma versão específica de um objeto (p. 350)</a> .

[Exemplo 1: Concessão da permissão s3:PutObject com uma condição que exige que o proprietário do bucket obtenha controle total](#)

Vamos supor que a conta A seja proprietária de um bucket e que o administrador da conta queira conceder a Dave, um usuário na conta B, permissões para fazer upload de objetos. Por padrão, os objetos que Dave carrega são de propriedade da conta B, e a conta A não tem permissões nesses objetos. Como o proprietário do bucket é quem paga a fatura, ele quer permissões completas nos objetos que Dave carrega. O administrador da conta A pode fazer isso concedendo a Dave a permissão s3:PutObject, com a condição de que a solicitação inclua cabeçalhos específicos da ACL, que concede permissão total explícita ou usa uma ACL pré-configurada (consulte [PUT Object](#)).

- Exija o cabeçalho `x-amz-full-control` na solicitação com permissão de controle total do proprietário do bucket.

A política de bucket a seguir concede ao usuário Dave a permissão `s3:PutObject` com uma condição de uso da chave de condição `s3:x-amz-grant-full-control`, que exige que a solicitação inclua o cabeçalho `x-amz-full-control`.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "statement1",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::AccountB-ID:user/Dave"  
            },  
            "Action": "s3:PutObject",  
            "Resource": "arn:aws:s3:::examplebucket/*",  
            "Condition": {  
                "StringEquals": {  
                    "s3:x-amz-grant-full-control": "id=AccountA-CanonicalUserID"  
                }  
            }  
        }  
    ]  
}
```

#### Note

Este exemplo trata da permissão entre contas. No entanto, se Dave, que está recebendo a permissão, pertence à conta da AWS que é proprietária do bucket, essa permissão condicional não é necessária, pois a conta pai à qual Dave pertence é proprietária dos objetos que o usuário faz upload.

A política de bucket anterior concede permissão condicional ao usuário Dave na conta B. Enquanto essa política estiver em vigor, Dave poderá obter a mesma permissão sem nenhuma condição por meio de alguma outra política. Por exemplo, Dave pode pertencer a um grupo e você concede ao grupo a permissão `s3:PutObject` sem nenhuma condição. Para evitar essas brechas de permissão, você pode elaborar uma política de acesso mais estrita adicionando uma negação explícita. Neste exemplo, você nega explicitamente a permissão de upload ao usuário Dave se ele não incluir os cabeçalhos necessários na solicitação, concedendo permissões totais ao proprietário do bucket. A negação explícita sempre se sobrepõe a qualquer outra permissão concedida. Veja a seguir um exemplo de política de acesso revisada.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "statement1",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::AccountB-ID:user/AccountBadmin"  
            },  
            "Action": "s3:PutObject",  
            "Resource": "arn:aws:s3:::examplebucket/*",  
            "Condition": {  
                "StringEquals": {  
                    "s3:x-amz-grant-full-control": "id=AccountA-CanonicalUserID"  
                }  
            }  
        },  
        {  
            "Sid": "statement2",  
            "Effect": "Deny",  
            "Principal": "*/*",  
            "Action": "s3:PutObject",  
            "Resource": "arn:aws:s3:::examplebucket/*",  
            "Condition": {}  
        }  
    ]  
}
```

```
{  
    "Sid": "statement2",  
    "Effect": "Deny",  
    "Principal": {  
        "AWS": "arn:aws:iam::AccountB-ID:user/AccountBadmin"  
    },  
    "Action": "s3:PutObject",  
    "Resource": "arn:aws:s3:::examplebucket/*",  
    "Condition": {  
        "StringNotEquals": {  
            "s3:x-amz-grant-full-control": "id=AccountA-CanonicalUserID"  
        }  
    }  
}  
]  
}
```

Se você tem duas contas da AWS, teste a política usando a AWS Command Line Interface (AWS CLI). Anexe a política e, com as credenciais de Dave, teste a permissão usando o seguinte comando da AWS CLI `put-object`. Você fornece as credenciais de Dave adicionando o parâmetro `--profile`. Você concede permissão de controle total ao proprietário do bucket adicionando o parâmetro `--grant-full-control`. Para obter mais informações sobre a configuração e o uso da AWS CLI, consulte [Configurar as ferramentas para as demonstrações com exemplos \(p. 298\)](#).

```
aws s3api put-object --bucket examplebucket --key HappyFace.jpg --body c:\HappyFace.jpg  
--grant-full-control id="AccountA-CanonicalUserID" --profile AccountBUserProfile
```

- Exija o cabeçalho `x-amz-acl` com uma ACL padrão que concede permissão de controle total ao proprietário do bucket.

Para exigir o cabeçalho `x-amz-acl` na solicitação, você pode substituir o par de chave-valor no bloco `Condition` e especificar a chave de condição `s3:x-amz-acl`, conforme exibido abaixo.

```
"Condition": {  
    "StringNotEquals": {  
        "s3:x-amz-acl": "bucket-owner-full-control"  
    }  
}
```

Para testar a permissão usando a AWS CLI, especifique o parâmetro `--acl`. Em seguida, a AWS CLI adiciona o cabeçalho `x-amz-acl` ao enviar a solicitação.

```
aws s3api put-object --bucket examplebucket --key HappyFace.jpg --body c:\HappyFace.jpg  
--acl "bucket-owner-full-control" --profile AccountBadmin
```

### Exemplo 2: Concessão da permissão s3:PutObject que exige os objetos armazenados usando criptografia no lado do servidor

Vamos supor que a conta A é proprietária de um bucket. O administrador da conta deseja conceder a Jane, uma usuária na conta A, a permissão para fazer upload de objetos com a condição de que Jane sempre solicite criptografia no lado do servidor, de modo que o Amazon S3 salve objetos criptografados. O administrador da conta A pode fazer isso usando a chave de condição `s3:x-amz-server-side-encryption`, conforme exibido. O par de chave-valor no bloco `Condition` especifica a chave `s3:x-amz-server-side-encryption`.

```
"Condition": {  
    "StringNotEquals": {  
        "s3:x-amz-server-side-encryption": "AES256"
```

}

Ao testar a permissão usando a AWS CLI, adicione o parâmetro obrigatório usando o parâmetro `--server-side-encryption`.

```
aws s3api put-object --bucket examplebucket --key HappyFace.jpg --body c:\HappyFace.jpg --server-side-encryption "AES256" --profile AccountBadmin
```

### Exemplo 3: Concessão da permissão s3:PutObject para copiar objetos com uma restrição na origem da cópia

Na solicitação PUT objeto, quando você especifica o objeto de origem, isso é uma operação de cópia (consulte [PUT objeto - Copiar](#)). De acordo com isso, o proprietário do bucket pode conceder ao usuário permissão para copiar objetos com restrições na origem. Por exemplo:

- Permite a cópia de objetos somente do bucket sourcebucket.
- Permite a cópia de objetos do bucket sourcebucket e somente dos objetos cujo prefixo de nome de chave comece com public/. Por exemplo, sourcebucket/public/\*
- Permite a cópia somente de um objeto específico do sourcebucket. Por exemplo, sourcebucket/example.jpg.

A política de bucket a seguir concede ao usuário Dave a permissão s3:PutObject que permite copiar apenas objetos com uma condição de que a solicitação inclua o cabeçalho s3:x-amz-copy-source e o valor do cabeçalho especifique o prefixo de nome de chave /examplebucket/public/\*.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "cross-account permission to user in your own account",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::AccountA-ID:user/Dave"
            },
            "Action": ["s3:PutObject"],
            "Resource": "arn:aws:s3:::examplebucket/*"
        },
        {
            "Sid": "Deny your user permission to upload object if copy source is not / bucket/folder",
            "Effect": "Deny",
            "Principal": {
                "AWS": "arn:aws:iam::AccountA-ID:user/Dave"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::examplebucket/*",
            "Condition": {
                "StringNotLike": {
                    "s3:x-amz-copy-source": "examplebucket/public/*"
                }
            }
        }
    ]
}
```

Você pode testar a permissão usando o comando da AWS CLI `copy-object`. Especifique a origem adicionando o parâmetro `--copy-source`. O prefixo de nome de chave deve corresponder com o prefixo permitido na política. Você precisa inserir as credenciais do usuário de Dave usando o parâmetro

--profile. Para obter mais informações sobre a configuração da AWS CLI, consulte [Configurar as ferramentas para as demonstrações com exemplos \(p. 298\)](#).

```
aws s3api copy-object --bucket examplebucket --key HappyFace.jpg  
--copy-source examplebucket/public/PublicHappyFace1.jpg --profile AccountADave
```

Observe que a política anterior usa a condição `StringNotLike`. Para conceder a permissão para copiar apenas um objeto em específico, altere a condição de `StringNotLike` para `StringNotEquals` e, em seguida, especifique a chave de objeto exata, conforme exibido.

```
"Condition": {  
    "StringNotEquals": {  
        "s3:x-amz-copy-source": "examplebucket/public/PublicHappyFace1.jpg"  
    }  
}
```

#### Exemplo 4: Concessão de acesso a uma versão específica de um objeto

Vamos supor que a conta A seja proprietária de um bucket habilitado para versão. O bucket tem várias versões do objeto `HappyFace.jpg`. O administrador da conta agora deseja conceder ao seu usuário (Dave) permissão para obter apenas uma versão específica do objeto. O administrador da conta pode fazer isso concedendo a Dave a permissão condicional `s3:GetObjectVersion`, conforme exibido. O par de chave-valor no bloco `Condition` especifica a chave de condição `s3:VersionId`.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "statement1",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::AccountA-ID:user/Dave"  
            },  
            "Action": ["s3:GetObjectVersion"],  
            "Resource": "arn:aws:s3:::examplebucketversionenabled/HappyFace.jpg"  
        },  
        {  
            "Sid": "statement2",  
            "Effect": "Deny",  
            "Principal": {  
                "AWS": "arn:aws:iam::AccountA-ID:user/Dave"  
            },  
            "Action": ["s3:GetObjectVersion"],  
            "Resource": "arn:aws:s3:::examplebucketversionenabled/HappyFace.jpg",  
            "Condition": {  
                "StringNotEquals": {  
                    "s3:VersionId": "AaaHbAQitwiL_h47_44lR02DDfLlB05e"  
                }  
            }  
        }  
    ]  
}
```

Neste caso, Dave precisa saber o ID de versão do objeto exato para recuperar o objeto.

Você pode testar as permissões usando o comando da AWS CLI `get-object` com o parâmetro `--version-id` que identifica a versão específica do objeto. O comando recupera o objeto e o salva no arquivo `OutputFile.jpg`.

```
aws s3api get-object --bucket examplebucketversionenabled --key HappyFace.jpg
OutputFile.jpg --version-id AaaHbAQitwiL_h47_44lRO2DDfLlBO5e --profile AccountADave
```

### Exemplo 5: Restringir uploads de objetos com uma classe de armazenamento específica

Vamos supor que a conta A é proprietária de um bucket. O administrador da conta deseja restringir Dave, um usuário na conta A, a fazer upload somente de objetos no bucket que serão armazenados com a classe de armazenamento STANDARD\_IA. O administrador da conta A pode fazer isso usando a chave de condição `s3:x-amz-storage-class`, conforme exibido no exemplo de política de bucket a seguir.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "statement1",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::AccountA-ID:user/Dave"
            },
            "Action": "s3:PutObject",
            "Resource": [
                "arn:aws:s3:::examplebucket/*"
            ],
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-storage-class": [
                        "STANDARD_IA"
                    ]
                }
            }
        }
    ]
}
```

### Chaves de condição do Amazon S3 para operações de bucket

A tabela a seguir mostra a lista de permissões específicas de operação de bucket que você pode conceder em políticas e, para cada permissão, as chaves disponíveis que você pode usar na especificação de uma condição.

Permissão	Chaves de condição aplicáveis	Descrição
<code>s3:CreateBucket</code>	<ul style="list-style-type: none"> <li><code>s3:x-amz-acl</code> (para permissões de ACL padrão)</li> <li><code>s3:x-amz-grant-permission</code> (para permissões explícitas), onde a <code>permissão</code> pode ser: <code>read</code>, <code>write</code>, <code>read-acp</code>, <code>write-acp</code>, <code>full-control</code></li> </ul>	A API de criação de bucket (consulte <a href="#">PUT bucket</a> ) oferece suporte a cabeçalhos específicos da ACL. Usando essas chaves de condição, você pode exigir que um usuário defina esses cabeçalhos na solicitação que concede permissões específicas.
	<code>s3:LocationConstraint</code>	Usando essa chave de condição, você pode restringir o usuário a criar buckets em uma região específica da AWS. Para ver um exemplo de política, consulte <a href="#">Exemplo 1: Permitir que um</a>

Permissão	Chaves de condição aplicáveis	Descrição
		<p>usuário crie um bucket, mas apenas em uma região específica (p. 354).</p>
s3>ListBucket	s3:prefix	<p>Usando essa chave de condição, você pode restringir a resposta da API Get Bucket (listar objetos) (consulte <a href="#">GET Bucket (listar objetos)</a>) a nomes de chave com um prefixo específico.</p> <p>A API Get Bucket (listar objetos) retorna uma lista de chaves de objeto no bucket especificado. Essa API oferece suporte ao cabeçalho <code>prefix</code> para recuperar apenas as chaves de objeto com um prefixo específico. Essa chave de condição está relacionada ao cabeçalho <code>prefix</code>.</p> <p>Por exemplo, o console do Amazon S3 oferece suporte ao conceito de pasta usando prefixos de nome de chave. Assim, se você tiver dois objetos com nomes de chave <code>public/object1.jpg</code> e <code>public/object2.jpg</code>, o console mostrará os objetos na pasta <code>public</code>. Se você organizar suas chaves de objeto usando tais prefixos, poderá conceder a permissão <code>s3&gt;ListBucket</code> com a condição que permitirá que o usuário obtenha uma lista de nomes de chave com um prefixo específico.</p> <p>Para ver um exemplo de política, consulte <a href="#">Exemplo 2: Permitir que um usuário obtenha uma lista de objetos em um bucket de acordo com um prefixo específico (p. 356)</a>.</p>
	s3:delimiter	<p>Se você organizar seus nomes de chave de objeto usando prefixos e delimitadores, poderá usar essa chave de condição para exigir que o usuário especifique o parâmetro <code>delimiter</code> na solicitação Get bucket (listar objetos). Neste caso, a resposta que o Amazon S3 retorna é uma lista de chaves de objeto com prefixos comuns agrupados. Para ver um exemplo de uso de prefixos e delimitadores, acesse <a href="#">Get bucket (listar objetos)</a>.</p>

Permissão	Chaves de condição aplicáveis	Descrição
	<code>s3:max-keys</code>	<p>Usando essa condição, você pode limitar o número de chaves que o Amazon S3 retorna na resposta à solicitação Get Bucket (Listar objetos) exigindo que o usuário especifique o parâmetro <code>max-keys</code>. Por padrão, a API retorna até 1.000 nomes de chave. Para ver uma lista de condições numéricas que você pode usar, consulte <a href="#">Operadores numéricos de condição</a> no Guia do usuário do IAM.</p>
<code>s3&gt;ListBucketVersions</code>	<code>s3:prefix</code>	<p>Se seu bucket for habilitado para versão, você poderá usar a API GET Bucket Object versions (consulte <a href="#">GET versões de objeto do bucket</a>) para recuperar os metadados de todas as versões de objetos. Para essa API, o proprietário do bucket deve conceder a permissão <code>s3&gt;ListBucketVersions</code> na política.</p> <p>Usando essa chave de condição, você pode limitar a resposta da API a nomes de chave com um prefixo específico exigindo que o usuário especifique o parâmetro <code>prefix</code> na solicitação com um valor específico.</p> <p>Por exemplo, o console do Amazon S3 oferece suporte ao conceito de pasta usando prefixos de nome de chave. Se você tiver dois objetos com nomes de chave <code>public/object1.jpg</code> e <code>public/object2.jpg</code>, o console mostrará os objetos na pasta <code>public</code>. Se você organizar suas chaves de objeto usando tais prefixos, poderá conceder a permissão <code>s3&gt;ListBucket</code> com a condição que permitirá que o usuário obtenha uma lista de nomes de chave com um prefixo específico.</p> <p>Para ver um exemplo de política, consulte <a href="#">Exemplo 2: Permitir que um usuário obtenha uma lista de objetos em um bucket de acordo com um prefixo específico</a> (p. 356).</p>

Permissão	Chaves de condição aplicáveis	Descrição
	s3:delimiter	Se você organizar seus nomes de chave de objeto usando prefixos e delimitadores, poderá usar essa chave de condição para exigir que o usuário especifique o parâmetro <code>delimiter</code> na solicitação GET versões de objeto do bucket. Neste caso, a resposta que o Amazon S3 retorna é uma lista de chaves de objeto com prefixos comuns agrupados.
	s3:max-keys	Usando essa condição, você pode restringir o número de chaves que o Amazon S3 retorna em resposta à solicitação das versões GET Bucket Object exigindo que o usuário especifique o parâmetro <code>max-keys</code> . Por padrão, a API retorna até 1.000 nomes de chave. Para ver uma lista de condições numéricas que você pode usar, consulte <a href="#">Operadores numéricos de condição</a> no Guia do usuário do IAM.
s3:PutBucketAcl	<ul style="list-style-type: none"> <li>• s3:x-amz-acl (para permissões de ACL padrão)</li> <li>• s3:x-amz-grant-permission (para permissões explícitas), onde a <code>permissão</code> pode ser: <code>read, write, read-acp, write-acp, full-control</code></li> </ul>	A API PUT Bucket acl (consulte <a href="#">PUT bucket</a> ) oferece suporte a cabeçalhos específicos de ACL. Você pode usar essas chaves de condição para exigir que um usuário defina esses cabeçalhos na solicitação.

#### Exemplo 1: Permitir que um usuário crie um bucket, mas apenas em uma região específica

Vamos supor que um administrador de uma conta da AWS queira conceder ao seu usuário (Dave) a permissão para criar um bucket somente na região América do Sul (São Paulo). O administrador da conta pode anexar a política de usuário a seguir que concede a permissão s3:CreateBucket com uma condição, conforme exibido. O par de chave-valor no bloco Condition especifica a chave s3:LocationConstraint e a região sa-east-1 como seu valor.

##### Note

Neste exemplo, o proprietário do bucket concede permissão para um de seus usuários, de modo que tanto uma política de bucket quanto uma política de usuário podem ser usadas. Este exemplo mostra uma política de usuário.

Para obter uma lista das regiões do Amazon S3, acesse [Regiões e endpoints](#), no Referência geral do Amazon Web Services.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "s3:CreateBucket",
      "Condition": {
        "s3:LocationConstraint": "sa-east-1"
      },
      "Resource": "arn:aws:s3:::mybucket"
    }
  ]
}
```

```
    "Sid":"statement1",
    "Effect":"Allow",
    "Action":[
        "s3:CreateBucket"
    ],
    "Resource":[
        "arn:aws:s3::::*"
    ],
    "Condition": {
        "StringLike": {
            "s3:LocationConstraint": "sa-east-1"
        }
    }
}
]
```

Esta política impede que o usuário crie um bucket em qualquer outra região, exceto sa-east-1. No entanto, é possível que outra política conceda ao usuário a permissão para criar buckets em outra região. Por exemplo, se ele pertencer a um grupo, o grupo pode ter uma política anexada que permite que todos seus usuários tenham a permissão para criar buckets em outra região. Para garantir que o usuário não tenha permissão para criar buckets em nenhuma outra região, adicione uma declaração de negação explícita nesta política.

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"statement1",
            "Effect":"Allow",
            "Action":[
                "s3:CreateBucket"
            ],
            "Resource":[
                "arn:aws:s3::::*"
            ],
            "Condition": {
                "StringLike": {
                    "s3:LocationConstraint": "sa-east-1"
                }
            }
        },
        {
            "Sid":"statement2",
            "Effect":"Deny",
            "Action":[
                "s3:CreateBucket"
            ],
            "Resource":[
                "arn:aws:s3::::*"
            ],
            "Condition": {
                "StringNotLike": {
                    "s3:LocationConstraint": "sa-east-1"
                }
            }
        }
    ]
}
```

A declaração Deny usa a condição StringNotLike. Isto é, uma solicitação de criação de bucket será negada se a restrição de localização não for "sa-east-1". A negação explícita não permite que o usuário crie um bucket em nenhuma outra região, independentemente de outras permissões que o usuário receba.

Você pode testar a política usando o seguinte comando `create-bucket` da AWS CLI. Este exemplo usa o arquivo `bucketconfig.txt` para especificar a restrição de localização. Observe o caminho do arquivo do Windows. Você precisa atualizar o nome e o caminho do bucket conforme apropriado. Você deve fornecer as credenciais do usuário usando o parâmetro `--profile`. Para obter mais informações sobre a configuração e o uso da AWS CLI, consulte [Configurar as ferramentas para as demonstrações com exemplos \(p. 298\)](#).

```
aws s3api create-bucket --bucket examplebucket --profile AccountADave --create-bucket-configuration file://c:/Users/someUser/bucketconfig.txt
```

O arquivo `bucketconfig.txt` especifica a configuração da seguinte maneira:

```
{"LocationConstraint": "sa-east-1"}
```

### Exemplo 2: Permitir que um usuário obtenha uma lista de objetos em um bucket de acordo com um prefixo específico

Um proprietário de bucket pode impedir que um usuário liste o conteúdo de uma pasta específica no bucket. Será útil se os objetos no bucket forem organizados por prefixos de nome de chave. O console do Amazon S3 usa os prefixos para mostrar uma hierarquia de pastas (somente o console oferece suporte ao conceito de pastas; a API do Amazon S3 oferece suporte apenas a buckets e objetos).

Neste exemplo, o proprietário do bucket e a conta pai à qual o usuário pertence são os mesmos. Assim, o proprietário do bucket pode usar uma política de bucket ou uma política de usuário. Primeiro, exibimos uma política de usuário.

A política de usuário a seguir concede a permissão `s3>ListBucket` (consulte [GET bucket \(listar objetos\)](#)) com uma condição que exige que o usuário especifique o `prefix` na solicitação com o valor `projects`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "statement1",
            "Effect": "Allow",
            "Action": [
                "s3>ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::examplebucket"
            ],
            "Condition": {
                "StringEquals": {
                    "s3:prefix": "projects"
                }
            }
        },
        {
            "Sid": "statement2",
            "Effect": "Deny",
            "Action": [
                "s3>ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::examplebucket"
            ],
            "Condition": {
                "StringNotEquals": {
                    "s3:prefix": "projects"
                }
            }
        }
    ]
}
```

```
        }
    ]  
}
```

A condição restringe o usuário a listar chaves de objeto com o prefixo `projects`. A negação explícita adicionada nega a solicitação do usuário para listar chaves com qualquer outro prefixo, independentemente de outras permissões que o usuário possa ter. Por exemplo, é possível que o usuário receba a permissão para listar chaves de objeto sem nenhuma restrição, tanto por meio de atualizações na política de usuário anterior quanto por meio de uma política de bucket. Mas, como a negação explícita sempre prevalece, a solicitação do usuário para listar outras chaves além do prefixo `project` é negada.

A política anterior é uma política de usuário. Se você adicionar o elemento `Principal` à política, identificando o usuário, agora terá uma política de bucket conforme exibido.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "statement1",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::BucketOwner-accountID:user/user-name"
            },
            "Action": [
                "s3>ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::examplebucket"
            ],
            "Condition": {
                "StringEquals": {
                    "s3:prefix": "examplefolder"
                }
            }
        },
        {
            "Sid": "statement2",
            "Effect": "Deny",
            "Principal": {
                "AWS": "arn:aws:iam::BucketOwner-AccountID:user/user-name"
            },
            "Action": [
                "s3>ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::examplebucket"
            ],
            "Condition": {
                "StringNotEquals": {
                    "s3:prefix": "examplefolder"
                }
            }
        }
    ]
}
```

Você pode testar a política usando o seguinte comando `list-object` da AWS CLI. No comando, você fornece as credenciais do usuário usando o parâmetro `--profile`. Para obter mais informações sobre a configuração e o uso da AWS CLI, consulte [Configurar as ferramentas para as demonstrações com exemplos \(p. 298\)](#).

```
aws s3api list-objects --bucket examplebucket --prefix examplefolder --profile AccountADave
```

Agora, se o bucket tiver o versionamento habilitado, para listar os objetos no bucket, em vez da permissão s3:ListBucket, você deverá conceder a permissão s3:ListBucketVersions na política anterior. Essa permissão também oferece suporte à chave de condição s3:prefix.

## Exemplos de políticas de bucket

Esta seção apresenta alguns exemplos de casos de uso típicos de políticas de bucket. As políticas usam as sequências `bucket` e `examplebucket` no valor do recurso. Para testar essas políticas, você precisará substituir essas sequências pelo nome do bucket. Para obter informações sobre a linguagem de políticas de acesso, consulte [Visão geral da linguagem da política de acesso \(p. 326\)](#).

### Note

As políticas de bucket são limitadas a 20 KB.

Você pode usar o [Gerador de políticas da AWS](#) para criar uma política de bucket para o bucket do Amazon S3. Em seguida, você pode usar o documento gerado para definir a política de bucket usando o [Console do Amazon S3](#), várias ferramentas de terceiros ou seu aplicativo.

### Important

Ao testar as permissões usando o console do Amazon S3, você precisará conceder permissões adicionais necessárias para o console — as permissões —s3>ListAllMyBuckets, s3:GetBucketLocation e s3>ListBucket. Para obter um exemplo de passo a passo que concede permissões aos usuários e testa-as usando o console, consulte [Uma demonstração de exemplo: uso de políticas de usuário para controlar o acesso ao bucket \(p. 372\)](#).

### Tópicos

- [Conceder permissões a várias contas com condições adicionadas \(p. 358\)](#)
- [Conceder permissão somente leitura para um usuário anônimo \(p. 359\)](#)
- [Restringir o acesso a endereços IP específicos \(p. 359\)](#)
- [Restringir o acesso a um indicador HTTP específico \(p. 360\)](#)
- [Conceder permissão a uma identidade de origem do Amazon CloudFront \(p. 361\)](#)
- [Adição de uma política de bucket para exigir MFA \(p. 362\)](#)
- [Conceder permissões entre contas para fazer upload de objetos garantindo que o proprietário do bucket tenha controle total \(p. 364\)](#)
- [Conceder permissões para o inventário do Amazon S3 e a análise do Amazon S3 \(p. 364\)](#)
- [Exemplo de políticas de bucket para VPC endpoints para o Amazon S3 \(p. 365\)](#)

## Conceder permissões a várias contas com condições adicionadas

A política de exemplo a seguir concede as permissões s3:PutObject e s3:PutObjectAcl a várias contas da AWS e exige que todas as solicitações para essas operações incluam a ACL public-read predefinida. Para obter mais informações, consulte [Especificação de permissões em uma política \(p. 330\)](#) e [Especificação de condições em uma política \(p. 335\)](#).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AddCannedAcl",  
            "Effect": "Allow",  
            "Principal": {"AWS":  
                ["arn:aws:iam::111122223333:root", "arn:aws:iam::444455556666:root"]},  
            "Action": "s3:PutObject",  
            "Resource": "arn:aws:s3:::examplebucket/*",  
            "Condition": {"StringLike": {"aws:SourceIdentity": "public-read"},  
                "StringNotLike": {"aws:SourceIdentity": "public-read-write"}},  
            "ConditionOperator": "And"  
        },  
        {  
            "Sid": "ListBucket",  
            "Effect": "Allow",  
            "Principal": {"AWS":  
                ["arn:aws:iam::111122223333:root", "arn:aws:iam::444455556666:root"]},  
            "Action": "s3:ListBucket",  
            "Resource": "arn:aws:s3:::examplebucket",  
            "Condition": {"StringLike": {"aws:SourceIdentity": "public-read"},  
                "StringNotLike": {"aws:SourceIdentity": "public-read-write"}},  
            "ConditionOperator": "And"  
        }  
    ]  
}
```

```
        "Action": ["s3:PutObject", "s3:PutObjectAcl"],
        "Resource": ["arn:aws:s3:::examplebucket/*"],
        "Condition": {"StringEquals": {"s3:x-amz-acl": ["public-read"]}}}
    }
}
```

## Conceder permissão somente leitura para um usuário anônimo

A política de exemplo a seguir concede a permissão `s3:GetObject` aos usuários anônimos públicos. Para obter uma lista de permissões e as operações que elas permitem, consulte [Especificação de permissões em uma política \(p. 330\)](#). Essa permissão permite que qualquer pessoa leia os dados do objeto, o que é útil quando você configura o bucket como um site e deseja que todos possam ler os objetos do bucket.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AddPerm",
      "Effect": "Allow",
      "Principal": "*",
      "Action": ["s3:GetObject"],
      "Resource": ["arn:aws:s3:::examplebucket/*"]
    }
  ]
}
```

### Warning

Tenha cuidado ao conceder acesso anônimo ao bucket do S3. Quando você concede acesso anônimo, qualquer pessoa no mundo pode acessar seu bucket. É altamente recomendável que você nunca conceda nenhum tipo de acesso anônimo de gravação ao seu bucket do S3.

## Restringir o acesso a endereços IP específicos

O exemplo a seguir concede permissões a qualquer usuário para executar qualquer operação do Amazon S3 em objetos no bucket especificado. No entanto, a solicitação deve se originar no intervalo de endereços IP especificados na condição.

A condição nesta instrução identifica o intervalo 54.240.143.\* de endereços IP do protocolo de internet versão 4 (IPv4), com uma exceção: 54.240.143.188.

O bloco da Condition usa as condições `IpAddress` e `NotIpAddress` e a chave de condição `aws:SourceIp`, que é uma chave de condição que abrange toda a AWS. Para obter mais informações sobre essas chaves de condição, consulte [Especificação de condições em uma política \(p. 335\)](#). Os valores IPv4 `aws:sourceIp` usam a notação CIDR padrão. Para obter mais informações, consulte [Operadores de condição de endereço IP](#) no Guia do usuário do IAM

```
{
  "Version": "2012-10-17",
  "Id": "S3PolicyId1",
  "Statement": [
    {
      "Sid": "IPAllow",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::examplebucket/*",
      "Condition": {
        "IpAddress": "54.240.143.188/32"
      }
    }
  ]
}
```

```
        "Condition": {
            "IpAddress": {"aws:SourceIp": "54.240.143.0/24"},
            "NotIpAddress": {"aws:SourceIp": "54.240.143.188/32"}
        }
    }
}
```

## Permitir endereços IPv4 e IPv6

Ao começar a usar os endereços IPv6, recomendamos que você atualize todas as políticas da sua organização com os intervalos de endereços IPv6 além dos intervalos de IPv4 existentes para garantir que as políticas continuem a funcionar ao fazer a transição para o IPv6.

O exemplo da política de bucket a seguir mostra como misturar intervalos de endereços IPv4 e IPv6 para cobrir todos os endereços IP válidos de sua organização. A política de exemplo permitirá acesso aos endereços IP de exemplo 54.240.143.1 e 2001:DB8:1234:5678::1 e negará o acesso para os endereços 54.240.143.129 e 2001:DB8:1234:5678:ABCD::1.

Os valores de IPv6 para aws:sourceIp devem estar em formato CIDR padrão. Para IPv6, oferecemos suporte ao uso de :: para representar um intervalo de 0s, por exemplo 2032001:DB8:1234:5678::/64. Para obter mais informações, consulte [Operadores de condição de endereço IP](#) no Guia do usuário do IAM.

```
{
    "Id": "PolicyId2",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowIPmix",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": "arn:aws:s3:::examplebucket/*",
            "Condition": {
                "IpAddress": [
                    "aws:SourceIp": [
                        "54.240.143.0/24",
                        "2001:DB8:1234:5678::/64"
                    ]
                ],
                "NotIpAddress": [
                    "aws:SourceIp": [
                        "54.240.143.128/30",
                        "2001:DB8:1234:5678:ABCD::/80"
                    ]
                ]
            }
        }
    ]
}
```

## Restringir o acesso a um indicador HTTP específico

Suponha que você tem um site com nome de domínio ([www.example.com](http://www.example.com) ou [example.com](http://example.com)) com links para fotos e vídeos armazenados em seu bucket do S3, [examplebucket](#). Por padrão, todos os recursos do S3 são privados, portanto, somente a conta da AWS que criou os recursos pode acessá-los. Para permitir acesso de leitura a esses objetos em seu site, você pode adicionar uma política de bucket que conceda a permissão s3:GetObject com uma condição, usando a chave aws:referer, de que a solicitação get deve se originar de páginas específicas da web. A política a seguir especifica a condição StringLike com a chave aws:Referer.

```
{  
    "Version": "2012-10-17",  
    "Id": "http referer policy example",  
    "Statement": [  
        {  
            "Sid": "Allow get requests originating from www.example.com and example.com.",  
            "Effect": "Allow",  
            "Principal": "*",  
            "Action": "s3:GetObject",  
            "Resource": "arn:aws:s3:::examplebucket/*",  
            "Condition": {  
                "StringLike": {"aws:Referer": ["http://www.example.com/*", "http://example.com/*"]}  
            }  
        }  
    ]  
}
```

Certifique-se de que os navegadores que você usa incluem o cabeçalho `referer` HTTP na solicitação.

Você pode proteger ainda mais o acesso aos objetos no bucket `examplebucket` adicionando negação explícita à política de bucket conforme mostrado no exemplo a seguir. A negação explícita substitui qualquer permissão que você possa conceder no bucket `examplebucket` usando outros meios, como ACLs ou políticas de usuário.

```
{  
    "Version": "2012-10-17",  
    "Id": "http referer policy example",  
    "Statement": [  
        {  
            "Sid": "Allow get requests referred by www.example.com and example.com.",  
            "Effect": "Allow",  
            "Principal": "*",  
            "Action": "s3:GetObject",  
            "Resource": "arn:aws:s3:::examplebucket/*",  
            "Condition": {  
                "StringLike": {"aws:Referer": ["http://www.example.com/*", "http://example.com/*"]}  
            }  
        },  
        {  
            "Sid": "Explicit deny to ensure requests are allowed only from specific referer.",  
            "Effect": "Deny",  
            "Principal": "*",  
            "Action": "s3:*",  
            "Resource": "arn:aws:s3:::examplebucket/*",  
            "Condition": {  
                "StringNotLike": {"aws:Referer": ["http://www.example.com/*", "http://example.com/*"]}  
            }  
        }  
    ]  
}
```

## Conceder permissão a uma identidade de origem do Amazon CloudFront

A política de bucket do exemplo a seguir concede uma permissão de identidade de origem do CloudFront para obter (listar) todos os objetos no bucket do Amazon S3. A identidade de origem do CloudFront é usada para permitir o recurso de conteúdo privado do CloudFront. A política usa o prefixo CanonicalUser, em vez de AWS, para especificar um ID canônico de usuário. Para saber mais sobre o suporte ao CloudFront para atender a conteúdo privado, visite o tópico [Atender a conteúdo privado](#) no Guia do

desenvolvedor do Amazon CloudFront. Você deve especificar o ID canônico do usuário para a identidade de acesso de origem de sua distribuição do CloudFront. Para obter instruções sobre como localizar o ID canônico do usuário, consulte [Especificação de um principal em uma política \(p. 329\)](#).

```
{  
    "Version": "2012-10-17",  
    "Id": "PolicyForCloudFrontPrivateContent",  
    "Statement": [  
        {  
            "Sid": " Grant a CloudFront Origin Identity access to support private content",  
            "Effect": "Allow",  
            "Principal": {"CanonicalUser": "CloudFront Origin Identity Canonical User ID"},  
            "Action": "s3:GetObject",  
            "Resource": "arn:aws:s3:::examplebucket/*"  
        }  
    ]  
}
```

## Adição de uma política de bucket para exigir MFA

O Amazon S3 oferece suporte ao acesso de API protegido por MFA, um recurso que pode impor a autenticação multifator (MFA) para acessar os recursos do Amazon S3. A autenticação multifator fornece um nível extra de segurança que pode ser aplicado a seu ambiente da AWS. É um recurso de segurança que exige que os usuários comprovem a posse física de um dispositivo MFA fornecendo um código válido de MFA. Para obter mais informações, consulte [AWS Multi-Factor Authentication](#). Você pode exigir a autenticação MFA para todas as solicitações de acesso a seus recursos do Amazon S3.

Você pode impor o requisito de autenticação MFA usando a chave `aws:MultiFactorAuthAge` em uma política de bucket. Os usuários do IAM podem acessar recursos do Amazon S3 usando as credenciais temporárias emitidas pelo AWS Security Token Service (STS). Você fornece o código da MFA no momento da solicitação do STS.

Quando o Amazon S3 recebe uma solicitação com autenticação MFA, a chave `aws:MultiFactorAuthAge` fornece um valor numérico que indica há quanto tempo (em segundos) a credencial temporária foi criada. Se a credencial temporária fornecida na solicitação não foi criada usando um dispositivo MFA, esse valor de chave será nulo (ausente). Em uma política de bucket, você pode adicionar uma condição para verificar esse valor, conforme mostrado no exemplo de política de bucket a seguir. A política negará qualquer operação do Amazon S3 na pasta `/taxdocuments` no bucket `examplebucket` se a solicitação não for autenticada pela MFA. Para saber mais sobre a autenticação MFA, consulte [Usar a autenticação multifator \(MFA\) na AWS](#) no Guia do usuário do IAM.

```
{  
    "Version": "2012-10-17",  
    "Id": "123",  
    "Statement": [  
        {  
            "Sid": "",  
            "Effect": "Deny",  
            "Principal": "*",  
            "Action": "s3:*",  
            "Resource": "arn:aws:s3:::examplebucket/taxdocuments/*",  
            "Condition": { "Null": { "aws:MultiFactorAuthAge": true } }  
        }  
    ]  
}
```

A condição `Null` no bloco `Condition` é avaliada como verdadeira se o valor da chave `aws:MultiFactorAuthAge` for nulo indicando que as credenciais de segurança temporárias na solicitação foram criadas sem a chave de MFA.

A política de bucket a seguir é uma extensão da política de bucket anterior. A política inclui duas declarações de política. Uma declaração concede a permissão s3:GetObject em um bucket (examplebucket) para todos, e outra declaração restringe ainda mais o acesso à pasta examplebucket/taxdocuments no bucket exigindo a autenticação MFA.

```
{  
    "Version": "2012-10-17",  
    "Id": "123",  
    "Statement": [  
        {  
            "Sid": "",  
            "Effect": "Deny",  
            "Principal": "*",  
            "Action": "s3:*",  
            "Resource": "arn:aws:s3:::examplebucket/taxdocuments/*",  
            "Condition": { "Null": { "aws:MultiFactorAuthAge": true } }  
        },  
        {  
            "Sid": "",  
            "Effect": "Allow",  
            "Principal": "*",  
            "Action": ["s3:GetObject"],  
            "Resource": "arn:aws:s3:::examplebucket/*"  
        }  
    ]  
}
```

Opcionalmente, você pode usar uma condição numérica para limitar a duração na qual a chave aws:MultiFactorAuthAge é válida, independentemente do ciclo de vida da credencial de segurança temporária usada para autenticar a solicitação. Por exemplo, a seguinte política de bucket, além de exigir autenticação MFA, também verifica há quanto tempo a sessão temporária foi criada. A política negará qualquer operação se o valor da chave aws:MultiFactorAuthAge indicar que a sessão temporária foi criada há mais de uma hora (3.600 segundos).

```
{  
    "Version": "2012-10-17",  
    "Id": "123",  
    "Statement": [  
        {  
            "Sid": "",  
            "Effect": "Deny",  
            "Principal": "*",  
            "Action": "s3:*",  
            "Resource": "arn:aws:s3:::examplebucket/taxdocuments/*",  
            "Condition": { "Null": { "aws:MultiFactorAuthAge": true } }  
        },  
        {  
            "Sid": "",  
            "Effect": "Deny",  
            "Principal": "*",  
            "Action": "s3:*",  
            "Resource": "arn:aws:s3:::examplebucket/taxdocuments/*",  
            "Condition": { "NumericGreaterThan": { "aws:MultiFactorAuthAge": 3600 } }  
        },  
        {  
            "Sid": "",  
            "Effect": "Allow",  
            "Principal": "*",  
            "Action": ["s3:GetObject"],  
            "Resource": "arn:aws:s3:::examplebucket/*"  
        }  
    ]  
}
```

```
}
```

## Conceder permissões entre contas para fazer upload de objetos garantindo que o proprietário do bucket tenha controle total

Você pode permitir que outra conta da AWS faça upload de objetos em seu bucket. No entanto, você pode decidir que, como proprietário do bucket, você deve ter controle total sobre os objetos carregados no bucket. A política a seguir impõe que uma conta específica da AWS (111111111111) não tenha permissão para fazer upload de objetos a menos que essa conta conceda acesso de controle total ao proprietário do bucket identificado pelo endereço de e-mail (xyz@amazon.com). A condição `StringNotEquals` na política especifica a chave de condição `s3:x-amz-grant-full-control` para expressar a exigência (consulte [Especificação de condições em uma política \(p. 335\)](#)).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "111",
            "Effect": "Allow",
            "Principal": {"AWS": "111111111111"},
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::examplebucket/*"
        },
        {
            "Sid": "112",
            "Effect": "Deny",
            "Principal": {"AWS": "111111111111" },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::examplebucket/*",
            "Condition": {
                "StringNotEquals": {"s3:x-amz-grant-full-control": ["emailAddress=xyz@amazon.com"]}
            }
        }
    ]
}
```

## Conceder permissões para o inventário do Amazon S3 e a análise do Amazon S3

O inventário do Amazon S3 cria listas dos objetos em um bucket do S3, e a exportação da análise do Amazon S3 cria arquivos de saída dos dados usados na análise. O bucket para o qual o inventário lista objetos é chamado de bucket de origem. O bucket onde o arquivo de inventário é gravado e o bucket onde o arquivo de exportação da análise é gravado é chamado de bucket de destino. Você deve criar uma política de bucket para o bucket de destino ao configurar o inventário de um bucket do S3 e ao configurar a exportação da análise. Para obter mais informações, consulte [Inventário do Amazon S3 \(p. 273\)](#) e [Análise do Amazon S3 – análise de classe de armazenamento \(p. 267\)](#).

O exemplo da política de bucket a seguir concede permissão ao Amazon S3 para gravar objetos (PUTs) da conta do bucket de origem para o bucket de destino. Você usa uma política de bucket como essa no bucket de destino ao configurar o inventário do Amazon S3 e a exportação da análise do Amazon S3.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "InventoryAndAnalyticsExamplePolicy",
            "Effect": "Allow",
            "Principal": {"Service": "s3.amazonaws.com"},
```

```
    "Action": ["s3:PutObject"],
    "Resource": ["arn:aws:s3:::destination-bucket/*"],
    "Condition": {
        "ArnLike": {
            "aws:SourceArn": "arn:aws:s3:::source-bucket"
        },
        "StringEquals": {
            "aws:SourceAccount": "1234567890",
            "s3:x-amz-acl": "bucket-owner-full-control"
        }
    }
}
```

## Exemplo de políticas de bucket para VPC endpoints para o Amazon S3

Você pode usar políticas de bucket do Amazon S3 para controlar o acesso a buckets de Amazon Virtual Private Cloud endpoints específicos (Amazon VPC) ou VPCs específicas. Esta seção contém políticas de bucket de exemplo que podem ser usadas para controlar o acesso ao bucket do S3 de VPC endpoints. Para saber como configurar VPC endpoints, consulte [VPC endpoints](#) no Guia do usuário da Amazon VPC.

A Amazon VPC permite executar os recursos da Amazon Web Services (AWS) em uma rede virtual definida por você. Um VPC endpoint permite que você crie uma conexão privada entre sua VPC e outro serviço da AWS sem exigir acesso pela Internet por meio de uma conexão VPN, de uma Instância NAT ou do AWS Direct Connect.

Um VPC endpoint para o Amazon S3 é uma entidade lógica em uma VPC que permite conectividade somente com o Amazon S3. O VPC endpoint roteia as solicitações para o Amazon S3 e roteia as respostas de rotas de volta para a VPC. Os VPC endpoints só alteram a maneira como as solicitações são roteadas. Os endpoints públicos do Amazon S3 e os nomes DNS continuam a funcionar com os VPC endpoints. Para obter informações importantes sobre o uso de Amazon VPC endpoints com o Amazon S3, consulte [VPC endpoints do gateway](#) e [Endpoints para Amazon S3](#) no Guia do usuário da Amazon VPC.

Os VPC endpoints para o Amazon S3 fornecem duas maneiras de controlar o acesso aos dados do Amazon S3:

- É possível controlar as solicitações, os usuários ou os grupos permitidos por um VPC endpoint específico. Para obter informações sobre esse tipo de controle de acesso, consulte [Controlar acesso a serviços com VPC Endpoints](#) no Guia do usuário da Amazon VPC.
- Você pode controlar quais VPCs ou VPC endpoints têm acesso a seus buckets do S3 usando as políticas de bucket do S3. Para obter exemplos desse tipo de controle de acesso de política de bucket, consulte os seguintes tópicos sobre restrição de acesso.

### Tópicos

- [Restringir o acesso a um VPC endpoint específico \(p. 366\)](#)
- [Restringir o acesso a uma VPC específica \(p. 366\)](#)
- [Recursos relacionados \(p. 367\)](#)

### Important

Ao aplicar políticas de bucket do S3 para os VPC endpoints descritos nesta seção, você pode bloquear o acesso ao bucket inadvertidamente. As permissões de bucket destinadas a especificamente limitar o acesso do bucket às conexões originadas do seu VPC endpoint podem bloquear todas as conexões ao bucket. Para obter informações sobre como corrigir esse

problema, consulte [Como recuperar o acesso a um bucket do Amazon S3 depois de aplicar uma política ao bucket que restringe o acesso ao meu VPC endpoint?](#) no AWS Support Knowledge Center.

## Restringir o acesso a um VPC endpoint específico

O seguinte é um exemplo de um política de bucket do S3 que restringe o acesso a um bucket específico, examplebucket, somente no VPC endpoint com o ID vpce-1a2b3c4d. Essa política negará todo acesso ao bucket se o endpoint especificado não estiver sendo usado. A condição aws:sourceVpce é usada para especificar o endpoint. A condição aws:sourceVpc não requer um ARN para o recurso do VPC endpoint, somente o ID do VPC endpoint. Para obter mais informações sobre o uso de condições em uma política, consulte [Especificação de condições em uma política \(p. 335\)](#).

```
{  
    "Version": "2012-10-17",  
    "Id": "Policy1415115909152",  
    "Statement": [  
        {  
            "Sid": "Access-to-specific-VPCE-only",  
            "Principal": "*",  
            "Action": "s3:*",  
            "Effect": "Deny",  
            "Resource": ["arn:aws:s3:::examplebucket",  
                        "arn:aws:s3:::examplebucket/*"],  
            "Condition": {  
                "StringNotEquals": {  
                    "aws:sourceVpce": "vpce-1a2b3c4d"  
                }  
            }  
        }  
    ]  
}
```

## Restringir o acesso a uma VPC específica

Você pode criar uma política de bucket que restringe o acesso a uma VPC específica usando a condição aws:sourceVpc. Isso será útil se você tiver vários VPC endpoints configurados na mesma VPC e desejar gerenciar o acesso aos buckets do S3 para todos os endpoints. A seguir encontra-se um exemplo de política que permite que a VPC vpc-111bbb22 acesse examplebucket e seus objetos. Essa política negará todo acesso ao bucket se o endpoint a VPC especificada não estiver sendo usado. A chave de condição vpc-111bbb22 não requer um ARN para o recurso da VPC, somente o ID da VPC.

```
{  
    "Version": "2012-10-17",  
    "Id": "Policy1415115909153",  
    "Statement": [  
        {  
            "Sid": "Access-to-specific-VPC-only",  
            "Principal": "*",  
            "Action": "s3:*",  
            "Effect": "Deny",  
            "Resource": ["arn:aws:s3:::examplebucket",  
                        "arn:aws:s3:::examplebucket/*"],  
            "Condition": {  
                "StringNotEquals": {  
                    "aws:sourceVpc": "vpc-111bbb22"  
                }  
            }  
        }  
    ]  
}
```

## Recursos relacionados

- [Exemplos de políticas de bucket \(p. 358\)](#)
- [VPC endpoints no Guia do usuário da Amazon VPC](#)
- [Como recuperar o acesso a um bucket do Amazon S3 depois de aplicar uma política ao bucket que restringe o acesso ao meu VPC endpoint?](#)

## Exemplos de política de usuário

Esta seção mostra diversas políticas de usuário do IAM para controle de acesso de usuário ao Amazon S3. Para obter informações sobre a linguagem de políticas de acesso, consulte [Visão geral da linguagem da política de acesso \(p. 326\)](#).

As políticas de exemplo a seguir funcionarão se você testá-las por programação. Contudo, para usá-las com o console do Amazon S3, você precisará conceder permissões adicionais que são exigidas pelo console. Para obter informações sobre o uso de políticas como essas com o console do Amazon S3, consulte [Uma demonstração de exemplo: uso de políticas de usuário para controlar o acesso ao bucket \(p. 372\)](#).

### Tópicos

- [Exemplo: Permitir que um usuário do IAM acesse um dos seus buckets \(p. 367\)](#)
- [Exemplo: Permitir que cada usuário do IAM acesse uma pasta em um bucket \(p. 368\)](#)
- [Exemplo: Permitir que um grupo tenha uma pasta compartilhada no Amazon S3 \(p. 371\)](#)
- [Exemplo: Permitir que todos os seus usuários leiam objetos em uma parte do bucket corporativo \(p. 371\)](#)
- [Exemplo: Permitir que um parceiro solte arquivos em uma parte específica do bucket corporativo \(p. 371\)](#)
- [Uma demonstração de exemplo: uso de políticas de usuário para controlar o acesso ao bucket \(p. 372\)](#)

## Exemplo: Permitir que um usuário do IAM acesse um dos seus buckets

Neste exemplo, você pode conceder a um usuário do IAM na sua conta da AWS acesso a um dos seus buckets, `examplebucket`, e permitir que o usuário adicione, atualize e exclua objetos.

Além de conceder as permissões `s3:PutObject`, `s3:GetObject` e `s3:DeleteObject` ao usuário, a política também concede as permissões `s3>ListAllMyBuckets`, `s3:GetBucketLocation` e `s3>ListBucket`. Estas são permissões adicionais, exigidas pelo console. As ações `s3:PutObjectAcl` e `s3:GetObjectAcl` também são necessárias para copiar, recortar e colar objetos no console. Para obter um exemplo de passo a passo que concede permissões aos usuários e testa-as usando o console, consulte [Uma demonstração de exemplo: uso de políticas de usuário para controlar o acesso ao bucket \(p. 372\)](#).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3>ListAllMyBuckets"  
            ],  
            "Resource": "arn:aws:s3:::*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:PutObject",  
                "s3:GetObject",  
                "s3:DeleteObject"  
            ],  
            "Resource": "arn:aws:s3:::examplebucket/*"  
        }  
    ]  
}
```

```
{  
    "Effect": "Allow",  
    "Action": [  
        "s3>ListBucket",  
        "s3:GetBucketLocation"  
    ],  
    "Resource": "arn:aws:s3:::examplebucket"  
},  
{  
    "Effect": "Allow",  
    "Action": [  
        "s3:PutObject",  
        "s3:PutObjectAcl",  
        "s3:GetObject",  
        "s3:GetObjectAcl",  
        "s3>DeleteObject"  
    ],  
    "Resource": "arn:aws:s3:::examplebucket/*"  
}  
]  
}
```

## Exemplo: Permitir que cada usuário do IAM acesse uma pasta em um bucket

Neste exemplo, você quer que dois usuários do IAM, Alice e Bob, tenham acesso ao seu bucket, examplebucket, para que possam adicionar, atualizar e excluir objetos. Contudo, você quer restringir o acesso de cada usuário a uma única pasta no bucket. Você pode querer criar pastas com nomes que conhecida com os nomes de usuário.

```
examplebucket  
Alice/  
Bob/
```

Para conceder a cada usuário acesso à sua pasta, você pode escrever uma política para cada usuário e anexá-la individualmente. Por exemplo, você pode anexar a seguinte política ao usuário Alice para conceder a ela permissões específicas do Amazon S3 na pasta examplebucket/Alice.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:PutObject",  
                "s3:GetObject",  
                "s3:GetObjectVersion",  
                "s3>DeleteObject",  
                "s3>DeleteObjectVersion"  
            ],  
            "Resource": "arn:aws:s3:::examplebucket/Alice/*"  
        }  
    ]  
}
```

Você anexa uma política similar ao usuário Bob, identificando a pasta Bob no valor Resource.

Em vez de anexar políticas a usuários individuais, você pode escrever uma única política que usa uma variável da política e anexá-la a um grupo. Você precisará primeiro criar um grupo e adicionar os usuários Alice e Bob ao grupo. O exemplo de política a seguir concede um conjunto de permissões do

Amazon S3 na pasta `examplebucket/${aws:username}`. Quando a política é avaliada, a variável `${aws:username}` é substituída pelo nome do usuário do solicitante. Por exemplo, se Alice enviar uma solicitação para colocar um objeto, a operação será permitida apenas se Alice estiver fazendo upload do objeto na pasta `examplebucket/Alice`.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:PutObject",  
                "s3:GetObject",  
                "s3:GetObjectVersion",  
                "s3:DeleteObject",  
                "s3:DeleteObjectVersion"  
            ],  
            "Resource": "arn:aws:s3:::examplebucket/${aws:username}/*"  
        }  
    ]  
}
```

#### Note

Ao usar variáveis de política, você deve especificar explicitamente a versão 2012-10-17 na política. A versão padrão da linguagem da política de acesso, 2008-10-17, não oferece suporte a variáveis de política.

Se você quiser testar a política anterior no console do Amazon S3, o console exigirá permissão para permissões adicionais do Amazon S3, como exibido na política a seguir. Para obter informações sobre como o console usa essas permissões, consulte [Uma demonstração de exemplo: uso de políticas de usuário para controlar o acesso ao bucket \(p. 372\)](#).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowGroupToSeeBucketListInTheConsole",  
            "Action": [ "s3>ListAllMyBuckets", "s3:GetBucketLocation" ],  
            "Effect": "Allow",  
            "Resource": [ "arn:aws:s3:::*" ]  
        },  
        {  
            "Sid": "AllowRootLevelListingOfTheBucket",  
            "Action": [ "s3>ListBucket" ],  
            "Effect": "Allow",  
            "Resource": [ "arn:aws:s3:::examplebucket" ],  
            "Condition":{  
                "StringEquals":{  
                    "s3:prefix":[""], "s3:delimiter":[ "/"]  
                }  
            }  
        },  
        {  
            "Sid": "AllowListBucketOfASpecificUserPrefix",  
            "Action": [ "s3>ListBucket" ],  
            "Effect": "Allow",  
            "Resource": [ "arn:aws:s3:::examplebucket" ],  
            "Condition":{ "StringLike":{ "s3:prefix":["${aws:username}/*"] } }  
        },  
        {  
            "Sid": "AllowUserSpecificActionsOnlyInTheSpecificUserPrefix",  
            "Action": [ "s3:PutObject", "s3:GetObject", "s3:GetObjectVersion", "s3:DeleteObject", "s3:DeleteObjectVersion" ],  
            "Effect": "Allow",  
            "Resource": "arn:aws:s3:::examplebucket/${aws:username}/*"  
        }  
    ]  
}
```

```
        "Effect": "Allow",
        "Action": [
            "s3:PutObject",
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion"
        ],
        "Resource": "arn:aws:s3:::examplebucket/${aws:username}/*"
    }
}
```

#### Note

Na versão 2012-10-17 da política, as variáveis de política começam com \$. Essa mudança na sintaxe poderá criar um conflito se sua chave de objeto incluir um \$. Por exemplo, para incluir uma chave de objeto my\$file em uma política, você especifica o caractere \$ com \${\$}, my\${\$}file.

Embora os nomes de usuário do IAM sejam identificadores amigáveis e legíveis, eles não devem ser globalmente exclusivos. Por exemplo, se o usuário Bob deixar a empresa e outro Bob entrar, o novo Bob poderá acessar as informações do antigo Bob. Em vez de usar nomes de usuário, você pode criar pastas baseadas nos IDs de usuário. Cada ID de usuário é único. Neste caso, você deve modificar a política anterior para usar a variável de política \${aws:userid}. Para obter mais informações sobre identificadores de usuário, consulte [Identificadores do IAM](#) no Guia do usuário do IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:GetObjectVersion",
                "s3:DeleteObject",
                "s3:DeleteObjectVersion"
            ],
            "Resource": "arn:aws:s3:::my_corporate_bucket/home/${aws:userid}/*"
        }
    ]
}
```

## Permitir que não usuários do IAM (usuários de aplicativos móveis) accessem pastas em um bucket

Vamos supor que você queira desenvolver um aplicativo móvel, um jogo que armazena dados dos usuários em um bucket do S3. Para cada usuário do aplicativo, você quer criar uma pasta em seu bucket. Você também quer limitar o acesso de cada usuário à sua própria pasta. Mas você não pode criar pastas antes que alguém baixe seu aplicativo e comece a jogar, porque não tem um ID de usuário.

Neste caso, você pode exigir que os usuários criem uma conta em seu aplicativo usando provedores públicos de identidade, como Login with Amazon, Facebook ou Google. Depois que os usuários criarem uma conta em seu aplicativo por meio de um desses provedores, eles terão um ID de usuário que você poderá usar para criar pastas específicas de usuário em tempo de execução.

Você pode usar a federação de identidades da web no AWS Security Token Service para integrar informações do provedor de identidade com seu aplicativo e para obter credenciais de segurança temporárias para cada usuário. Você pode criar políticas do IAM que permitem que o aplicativo acesse seu bucket e execute operações como criação de pastas específicas de usuário e upload de dados. Para obter

mais informações sobre federação de identidades da web, consulte [Sobre federação de identidades da web](#) no Guia do usuário do IAM.

## Exemplo: Permitir que um grupo tenha uma pasta compartilhada no Amazon S3

Anexar a política a seguir ao grupo concede a todos no grupo acesso à seguinte pasta no Amazon S3: `my_corporate_bucket/share/marketing`. Os membros do grupo têm permissão para acessar apenas permissões específicas do Amazon S3 exibidas na política e apenas para objetos na pasta especificada.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:PutObject",  
                "s3:GetObject",  
                "s3:GetObjectVersion",  
                "s3:DeleteObject",  
                "s3:DeleteObjectVersion"  
            ],  
            "Resource": "arn:aws:s3:::my_corporate_bucket/share/marketing/*"  
        }  
    ]  
}
```

## Exemplo: Permitir que todos os seus usuários leiam objetos em uma parte do bucket corporativo

Neste exemplo, crie um grupo chamado `AllUsers`, que contém todos os usuários do IAM que pertencem à conta da AWS. Em seguida, anexe uma política que concede ao grupo acesso a `GetObject` e `GetObjectVersion`, mas somente para objetos na pasta `my_corporate_bucket/readonly`.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:GetObject",  
                "s3:GetObjectVersion"  
            ],  
            "Resource": "arn:aws:s3:::my_corporate_bucket/readonly/*"  
        }  
    ]  
}
```

## Exemplo: Permitir que um parceiro solte arquivos em uma parte específica do bucket corporativo

Neste exemplo, crie um grupo chamado `WidgetCo` que representa uma empresa parceira. Crie um usuário do IAM para a pessoa ou aplicativo específico na empresa parceira que precisa de acesso. Em seguida, coloque o usuário no grupo.

Depois, anexe uma política que concede ao grupo acesso `PutObject` à seguinte pasta no bucket corporativo: `my_corporate_bucket/uploads/widgetco`.

Impeça que o grupo `WidgetCo` faça qualquer outra coisa no bucket adicionando uma declaração que nega explicitamente outras permissões do Amazon S3, exceto `PutObject`, em qualquer recurso do Amazon S3 na conta da AWS. Esta etapa será necessária apenas se houver uma política abrangente em uso em outro lugar na sua conta da AWS que dê aos usuários amplo acesso a recursos do Amazon S3.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "s3:PutObject",  
            "Resource": "arn:aws:s3:::my_corporate_bucket/uploads/widgetco/*"  
        },  
        {  
            "Effect": "Deny",  
            "NotAction": "s3:PutObject",  
            "Resource": "arn:aws:s3:::my_corporate_bucket/uploads/widgetco/*"  
        },  
        {  
            "Effect": "Deny",  
            "Action": "s3:*",  
            "NotResource": "arn:aws:s3:::my_corporate_bucket/uploads/widgetco/*"  
        }  
    ]  
}
```

## Uma demonstração de exemplo: uso de políticas de usuário para controlar o acesso ao bucket

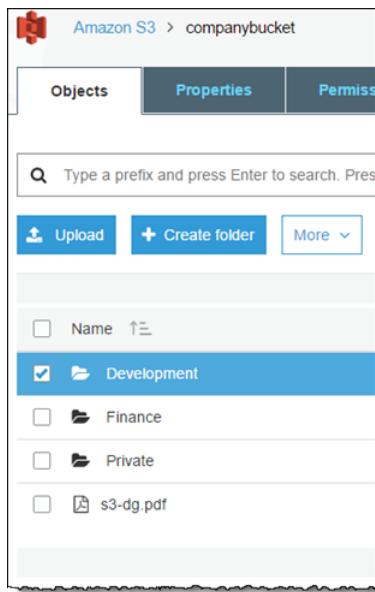
Esta demonstração explica como as permissões de usuário funcionam com o Amazon S3. Criaremos um bucket com pastas e, em seguida, criaremos usuários do AWS Identity and Access Management em sua conta da AWS e concederemos a eles permissões incrementais no seu bucket do Amazon S3 e nas pastas nele contidas.

### Tópicos

- [Contexto: noções básicas de buckets e pastas \(p. 372\)](#)
- [Exemplo de demonstração \(p. 374\)](#)
- [Etapa 0: Preparação para a demonstração \(p. 375\)](#)
- [Etapa 1: criar um bucket \(p. 375\)](#)
- [Etapa 2: criar usuários do IAM e um grupo \(p. 376\)](#)
- [Etapa 3: verificar se os usuários do IAM não têm nenhuma permissão \(p. 376\)](#)
- [Etapa 4: conceder permissões no nível do grupo \(p. 377\)](#)
- [Etapa 5: conceder permissões específicas do usuário do IAM Alice \(p. 383\)](#)
- [Etapa 6: conceder permissões específicas do usuário do IAM Bob \(p. 387\)](#)
- [Etapa 7: proteger a pasta Private \(p. 387\)](#)
- [Limpeza \(p. 389\)](#)
- [Recursos relacionados \(p. 389\)](#)

### Contexto: noções básicas de buckets e pastas

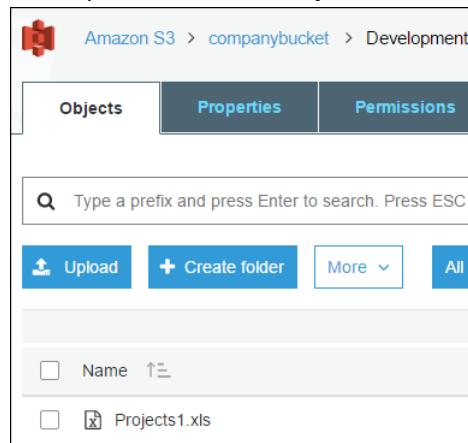
O modelo de dados do Amazon S3 é uma estrutura plana: você cria um bucket e o bucket armazena objetos. Não há hierarquia de buckets ou de subpastas. Contudo, você pode emular uma hierarquia de pastas. Ferramentas como o console do Amazon S3 podem apresentar uma exibição dessas pastas e subpastas lógicas em seu bucket, como mostrado aqui:



O console mostra que um bucket chamado companybucket tem três pastas, Private, Development e Finance, e um objeto s3-dg.pdf. O console usa os nomes de objeto (chaves) para criar uma hierarquia lógica com pastas e subpastas. Considere os seguintes exemplos:

- Ao criar a pasta Development, o console cria um objeto com a chave Development/. Observe o delimitador '/' na parte final.
- Quando você carregar um objeto chamado Projects1.xls na pasta Development, o console carregará o objeto e fornecerá a chave Development/Projects1.xls.

Na chave, Development é o prefixo e ' / ' é o delimitador. A API do Amazon S3 oferece suporte a prefixes e delimitadores em suas operações. Por exemplo, você pode obter uma lista de todos os objetos no bucket com um prefixo e um delimitador específicos. No console, quando você clica duas vezes na pasta Development, o console lista os objetos na pasta. No exemplo a seguir, a pasta Development contém um objeto.



Quando o console lista a pasta Development no bucket companybucket, ele envia uma solicitação para o Amazon S3 na qual especifica o prefixo Development e o delimitador '/' na solicitação. A resposta do console parece uma lista de pastas no sistema de arquivos de seu computador. O exemplo anterior mostra que o bucket companybucket tem um objeto com a chave Development/Projects1.xls.

O console está usando chaves de objeto para pressupor uma hierarquia lógica; o Amazon S3 não tem nenhuma hierarquia física, somente os buckets que contêm objetos em uma estrutura de arquivo simples. Quando você cria objetos usando a API do Amazon S3, pode usar chaves de objeto que implicam uma hierarquia lógica.

Quando você cria uma hierarquia lógica de objetos, pode gerenciar o acesso a pastas individuais, como faremos nesta demonstração.

Antes de ver a demonstração, você precisará se familiarizar com mais um conceito: conteúdo de bucket no "nível raiz". Suponha que seu bucket `companybucket` contenha os seguintes objetos:

`Private/privDoc1.txt`

`Private/privDoc2.zip`

`Development/project1.xls`

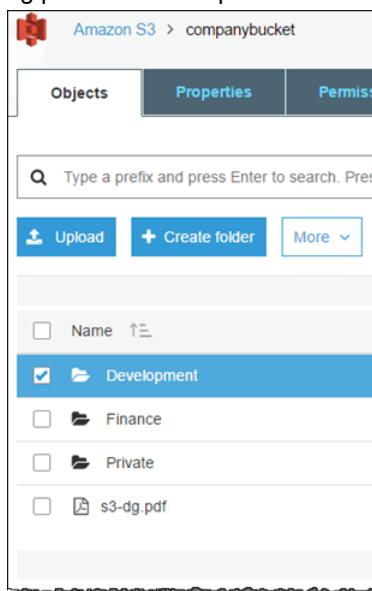
`Development/project2.xls`

`Finance/Tax2011/document1.pdf`

`Finance/Tax2011/document2.pdf`

`s3-dg.pdf`

Essas chaves de objeto criam uma hierarquia lógica com `Private`, `Development` e `Finance` como pastas no nível raiz e `s3-dg.pdf` como um objeto no nível raiz. Quando você clica no nome do bucket no console do Amazon S3, os itens no nível raiz são exibidos como mostrado. O console mostra os prefixos de nível superior (`Private/`, `Development/` e `Finance/`) como pastas no nível raiz. A chave de objeto `s3-dg.pdf` não tem um prefixo. Portanto, aparece como um item no nível raiz.



## Exemplo de demonstração

O exemplo desta demonstração é o seguinte:

- Você cria um bucket e adiciona três pastas (`Private`, `Development` e `Finance`).
- Você tem dois usuários, Alice e Bob. Você quer que Alice acesse somente a pasta `Development` e Bob acesse somente a pasta `Finance`, e deseja manter o conteúdo da pasta `Private` como private. Na

demonstração, você gerencia o acesso criando usuários do AWS Identity and Access Management (IAM) (usaremos os mesmos nomes de usuário, Alice e Bob) e concede as permissões necessárias.

O IAM também permite criar grupos de usuários e conceder permissões no nível do grupo que se aplicam a todos os usuários no grupo. Isso ajuda a gerenciar melhor as permissões. Para este exercício, Alice e Bob precisarão de algumas permissões comuns. Então você também criará um grupo chamado Consultores e adicionará Alice e Bob ao grupo. Primeiro, você concede permissões anexando uma política de grupo ao grupo. Depois, você adicionará permissões específicas do usuário anexando políticas a usuários específicos.

#### Note

A demonstração usa `companybucket` como o nome de bucket, Alice e Bob como os usuários do IAM e Consultores como o nome do grupo. Como o Amazon S3 exige que os nomes de bucket sejam globalmente exclusivos, você precisará substituir o nome de bucket pelo nome que criar.

### Etapa 0: Preparação para a demonstração

Neste exemplo, você usará suas credenciais de conta da AWS para criar usuários do IAM. Inicialmente, esses usuários não têm nenhuma permissão. Você concederá gradualmente a esses usuários permissões para executar ações específicas do Amazon S3. Para testar essas permissões, você entrará no console com as credenciais de cada usuário. Conforme conceder gradualmente permissões como proprietário da conta da AWS e testar permissões como um usuário do IAM, você precisará entrar e sair, sempre usando credenciais diferentes. Você pode fazer esses testes com um navegador, mas o processo será mais rápido se você usar dois navegadores diferentes: use um navegador para se conectar ao Console de gerenciamento da AWS com suas credenciais de conta da AWS e outro para se conectar com credenciais de usuário do IAM.

Para entrar no Console de gerenciamento da AWS com suas credenciais de conta da AWS, acesse <https://console.aws.amazon.com/>. Um usuário do IAM não pode fazer login usando o mesmo link. Um usuário do IAM deve usar uma página de login habilitada para o IAM. Como proprietário da conta, você pode fornecer este link para seus usuários.

Para obter mais informações sobre o IAM, acesse [a página de login do Console de gerenciamento da AWS](#) no Guia do usuário do IAM.

#### Para fornecer um link de login para usuários do IAM

1. Faça login no Console de gerenciamento da AWS e abra o console da IAM em <https://console.aws.amazon.com/iam/>.
2. No painel Navigation (Navegação), clique em IAM Dashboard (Painel do IAM).
3. Observe o URL no link de login de usuários do IAM:. Você dará este link para usuários do IAM entrarem no console com seu nome de usuário e senha do IAM.

### Etapa 1: criar um bucket

Nesta etapa, você entrará no console do Amazon S3 com suas credenciais de conta da AWS, criará um bucket, adicionará pastas (Development, Finance, Private) ao bucket e fará upload de um ou dois documentos de exemplo em cada pasta.

1. Faça login no Console de gerenciamento da AWS e abra o console da Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Crie um bucket.

Para obter instruções passo a passo, consulte [Como criar um bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

3. Faça upload de um documento no bucket.

Este exercício pressupõe que você tem o documento `s3-dg.pdf` no nível raiz desse bucket. Se você carregar um documento diferente, substitua o nome de arquivo para `s3-dg.pdf`.

4. Adicione três pastas chamadas Private, Finance e Development ao bucket.
5. Faça upload de um ou dois documentos em cada pasta.

Para este exercício, suponha que você tenha carregado alguns documentos em cada pasta, resultando no bucket com objetos com as seguintes chaves:

Private/privDoc1.txt

Private/privDoc2.zip

Development/project1.xls

Development/project2.xls

Finance/Tax2011/document1.pdf

Finance/Tax2011/document2.pdf

s3-dg.pdf

Para obter instruções passo a passo, consulte [Como fazer upload de arquivos e pastas em um bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

## Etapa 2: criar usuários do IAM e um grupo

Agora use o console do IAM para adicionar dois usuários do IAM, Alice e Bob, à sua conta da AWS. Crie também um grupo administrativo chamado Consultores e adicione ambos os usuários ao grupo.

Warning

Quando você adicionar usuários e um grupo, não anexe políticas que concedem permissões a esses usuários. No início, esses usuários não terão nenhuma permissão. Nas próximas seções, você concederá permissões gradualmente. Primeiro verifique se atribuiu senhas a esses usuários do IAM. Você utilizará essas credenciais de usuário para testar ações do Amazon S3 e verificar se as permissões funcionam como esperado.

Para obter instruções detalhadas sobre como criar um novo usuário do IAM, consulte [Criar um usuário do IAM em sua conta da AWS](#) no Guia do usuário do IAM. Ao criar usuários para esta apresentação, verifique o "acesso ao Console de Gerenciamento da AWS" e deixe "acesso Programático" desmarcado.

Para obter instruções detalhadas sobre como criar um grupo administrativo, consulte a seção [Criar o primeiro usuário do IAM e do grupo de administradores](#) no Guia do usuário do IAM.

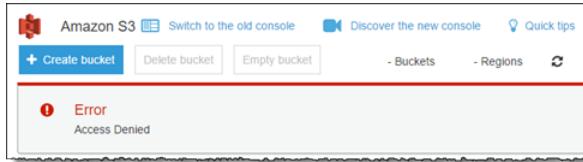
## Etapa 3: verificar se os usuários do IAM não têm nenhuma permissão

Se você estiver usando dois navegadores, agora poderá usar o segundo navegador para entrar no console usando uma das credenciais de usuário do IAM.

1. Usando o link de login de usuário do IAM (consulte [Para fornecer um link de login para usuários do IAM \(p. 375\)](#)), entre no console da AWS usando qualquer uma das credenciais de usuário do IAM.
2. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.

---

~~Verifique a seguinte mensagem de console que diz que o acesso foi negado.~~  
Versão da API 2006-03-01



Agora, vamos começar a conceder permissões incrementais aos usuários. Primeiro, você anexará uma política de grupo que concede permissões que ambos os usuários devem ter.

## Etapa 4: conceder permissões no nível do grupo

Queremos que todos os usuários possam fazer o seguinte:

- Listar todos os buckets de propriedade da conta-pai. Para fazer isso, Bob e Alice devem ter permissão para a ação s3>ListAllMyBuckets.
- Listar itens, pastas e objetos no nível raiz no bucket companybucket. Para fazer isso, Bob e Alice devem ter permissão para a ação s3>ListBucket no bucket companybucket.

Agora criaremos uma política que concede essas permissões e depois a anexaremos ao grupo Consultores.

### Etapa 4.1: conceder permissão para listar todos os buckets

Nesta etapa, você criará uma política gerenciada que concede aos usuários permissões mínimas para listar todos os buckets de propriedade da conta-pai e depois anexará a política ao grupo Consultores. Ao anexar a política gerenciada a um usuário ou um grupo, você concede ao usuário ou grupo permissão para obter uma lista de bucket na conta-pai da AWS.

1. Faça login no Console de gerenciamento da AWS e abra o console da IAM em <https://console.aws.amazon.com/iam/>.

#### Note

Como você concederá permissões de usuário, entre com suas credenciais de conta da AWS, não como um usuário do IAM.

2. Crie a política gerenciada.
  - a. No painel de navegação à esquerda, clique em Policies (Políticas) e, em seguida, clique em Create Policy (Criar política).
  - b. Clique na guia JSON.
  - c. Copie a política de acesso a seguir e cole-a no campo de texto da política:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowGroupToSeeBucketListInTheConsole",  
            "Action": ["s3>ListAllMyBuckets"],  
            "Effect": "Allow",  
            "Resource": ["arn:aws:s3:::*"]  
        }  
    ]  
}
```

Uma política é um documento JSON. No documento, uma Statement é uma matriz de objetos, cada um descrevendo uma permissão usando uma coleção de pares de valor-nome. A política

anterior descreve uma permissão específica. A `Action` especifica o tipo de acesso. Na política, `s3>ListAllMyBuckets` é uma ação predefinida do Amazon S3. Esta ação abrange a operação GET serviço do Amazon S3, que retorna uma lista de todos os buckets no remetente autenticado. O valor de elemento `Effect` determina se a permissão específica é permitida ou negada.

- d. Clique em Review Policy (Revisar política). Na próxima página, insira `AllowGroupToSeeBucketListInTheConsole` no campo Name (Nome), e depois clique em Create policy (Criar política).

Note

A entrada Summary (Resumo) exibirá uma mensagem afirmando que a política não dá quaisquer permissões. Para esta apresentação, você pode, de maneira segura, ignorar essa mensagem.

3. Anexe a política gerenciada `AllowGroupToSeeBucketListInTheConsole` que você criou para o grupo Consultores.

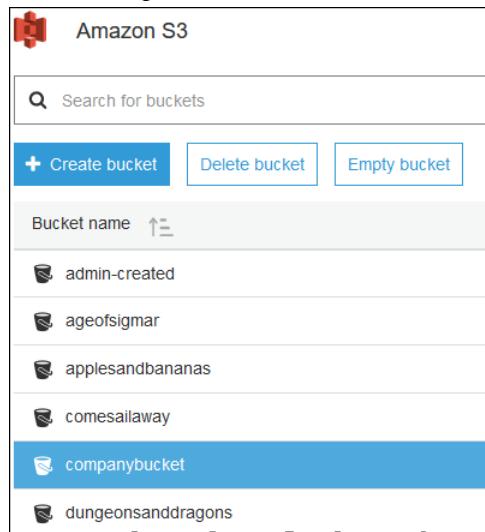
Para obter instruções detalhadas para anexar uma política gerenciada, consulte [Adicionar e remover políticas do IAM \(Console\)](#) no Guia do usuário do IAM.

Anexe documentos de política a usuários e grupos do IAM no console do IAM. Como queremos que ambos os usuários possam listar os buckets, anexamos a política ao grupo.

4. Teste a permissão.

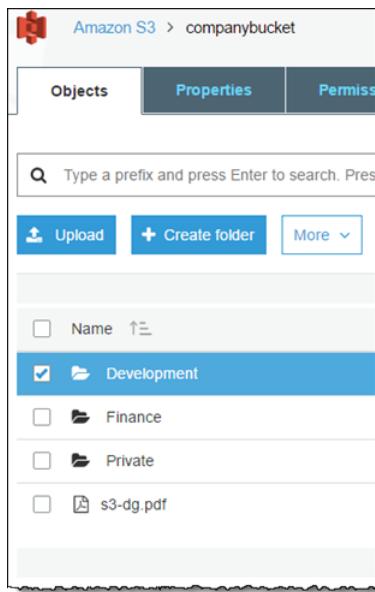
- a. Usando o link de login de usuário do IAM (consulte [Para fornecer um link de login para usuários do IAM \(p. 375\)](#)), entre no console da AWS usando qualquer uma das credenciais de usuário do IAM.
- b. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.

O console agora deve listar todos os buckets, mas não os objetos dos buckets.



#### [Etapa 4.2: permitir que os usuários listem o conteúdo do nível raiz de um bucket](#)

Agora vamos permitir que todos os usuários no grupo Consultores listem itens do bucket `companybucket` no nível raiz. Quando um usuário clica no bucket da empresa no console do Amazon S3, pode visualizar os itens no nível raiz do bucket.



Lembre-se de que estamos usando `companybucket` para ilustração. Você deve usar o nome do bucket que criou para este exercício.

Para entender qual solicitação o console envia ao Amazon S3 quando você clica no nome de um bucket, a resposta que o Amazon S3 retorna e como o console interpreta a resposta, é necessário ir um pouco além.

Quando você clica no nome de um bucket, o console envia a solicitação [GET Bucket \(Listar objetos\)](#) ao Amazon S3. Essa solicitação inclui os parâmetros a seguir:

- Parâmetro `prefix` com uma string vazia como seu valor.
- Parâmetro `delimiter` com `/` como seu valor.

Veja a seguir uma solicitação de exemplo:

```
GET ?prefix=&delimiter=/ HTTP/1.1
Host: companybucket.s3.amazonaws.com
Date: Wed, 01 Aug 2012 12:00:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:xQE0diMbLRepdf3YB+FIEXAMPLE=
```

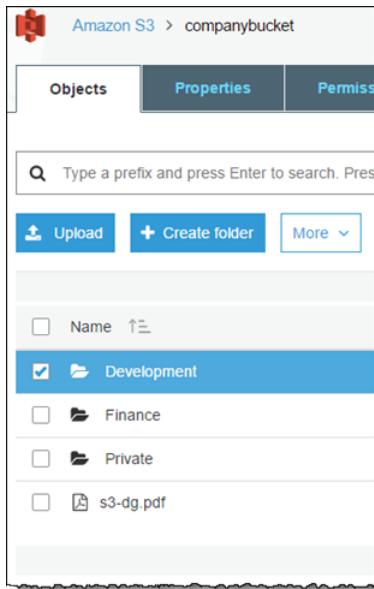
O Amazon S3 retorna uma resposta que inclui o seguinte elemento `<ListBucketResult>`:

```
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<Name>companybucket</Name>
<Prefix></Prefix>
<Delimiter></Delimiter>
...
<Contents>
<Key>s3-dg.pdf</Key>
...
</Contents>
<CommonPrefixes>
<Prefix>Development/<Prefix>
</CommonPrefixes>
<CommonPrefixes>
<Prefix>Finance/<Prefix>
</CommonPrefixes>
<CommonPrefixes>
```

```
<Prefix>Private/</Prefix>
</CommonPrefixes>
</ListBucketResult>
```

A chave `s3-dg.pdf` não contém o delimitador `'/'`, e o Amazon S3 retorna a chave no elemento `<Contents>`. Contudo, todas as outras chaves no bucket de exemplo contêm o delimitador `'/'`. O Amazon S3 agrupa essas chaves e retorna um elemento `<CommonPrefixes>` para cada um dos valores distintos de prefixo `Development/`, `Finance/` e `Private/`, isto é, uma substring desde o início dessas chaves até a primeira ocorrência do delimitador `'/'` especificado.

O console interpreta este resultado e exibe os itens no nível raiz como três pastas e uma chave de objeto.



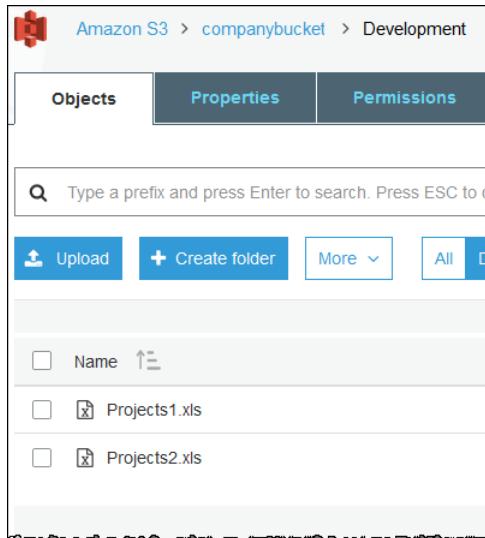
Agora, se Bob ou Alice clicarem duas vezes na pasta `Development`, o console enviará a solicitação [GET Bucket \(Listar objetos\)](#) ao Amazon S3 com os parâmetros `prefix` e `delimiter` definidos como os seguintes valores:

- Parâmetro `prefix` com o valor `Development/`.
- Parâmetro `delimiter` com o valor `'/'`.

Em resposta, o Amazon S3 retorna as chaves de objeto que começam com o prefixo especificado.

```
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<Name>companybucket</Name>
<Prefix>Development</Prefix>
<Delimiter>/<Delimiter>
...
<Contents>
<Key>Project1.xls</Key>
...
</Contents>
<Contents>
<Key>Project2.xls</Key>
...
</Contents>
</ListBucketResult>
```

O console mostra as chaves de objeto:



Agora, vamos voltar a conceder aos usuários permissão para listar itens de bucket no nível raiz. Para listar conteúdo de bucket, os usuários precisam de permissão para chamar a ação `s3>ListBucket`, conforme exibido na seguinte declaração de política. Para garantir que eles vejam somente o conteúdo no nível raiz, adicionamos uma condição de que os usuários devem especificar um `prefix` vazio na solicitação — isto é, eles não podem clicar duas vezes em nenhuma das pastas de nível raiz. Finalmente, vamos adicionar uma condição para solicitar acesso de pasta exigindo que as solicitações de usuário incluam o parâmetro `delimiter` com o valor '/'.

```
{  
    "Sid": "AllowRootLevelListingOfCompanyBucket",  
    "Action": ["s3>ListBucket"],  
    "Effect": "Allow",  
    "Resource": ["arn:aws:s3:::companybucket"],  
    "Condition": {  
        "StringEquals": {  
            "s3:prefix": [""], "s3:delimiter": ["/"]  
        }  
    }  
}
```

Ao usar o console do Amazon S3, observe que, ao clicar em um bucket, o console primeiro envia a solicitação [GET localização do bucket](#) para encontrar a região da AWS onde o bucket está implantado. Em seguida, o console usa o endpoint específico da região para o bucket enviar a solicitação [GET Bucket \(listar objetos\)](#). Como resultado, se os usuários usarem o console, você deverá conceder permissão para a ação `s3:GetBucketLocation` conforme exibido na seguinte declaração de política:

```
{  
    "Sid": "RequiredByS3Console",  
    "Action": ["s3:GetBucketLocation"],  
    "Effect": "Allow",  
    "Resource": ["arn:aws:s3:::*"]  
}
```

Para permitir que os usuários listem o conteúdo do nível raiz do bucket

1. Faça login no Console de gerenciamento da AWS e abra o console da Amazon S3 em <https://console.aws.amazon.com/s3/>.

Use suas credenciais de conta da AWS, não as credenciais de um usuário do IAM, para entrar no console.

2. Substitua a política gerenciada existente `AllowGroupToSeeBucketListInTheConsole` que está anexada ao grupo Consultores pela política a seguir, que também permite a ação `s3>ListBucket`. Lembre-se de substituir `companybucket` na política `Resource` pelo nome do bucket.

Para obter instruções detalhadas, consulte [Editar políticas gerenciadas pelo cliente](#) no Guia do usuário do IAM. Ao seguir as instruções detalhadas, não se esqueça de seguir as orientações para aplicar alterações em todas as entidades principais às quais a política está anexada.

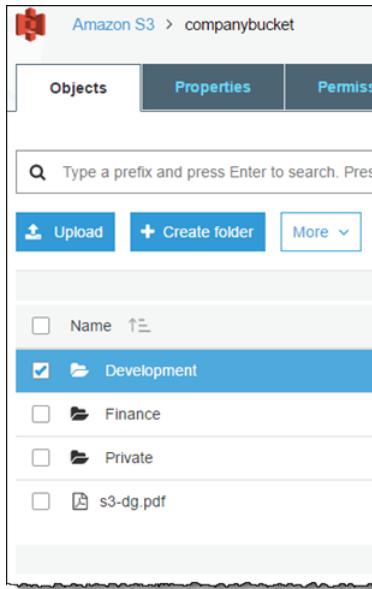
```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid":  
                "AllowGroupToSeeBucketListAndAlsoAllowGetBucketLocationRequiredForListBucket",  
                "Action": [ "s3>ListAllMyBuckets", "s3:GetBucketLocation" ],  
                "Effect": "Allow",  
                "Resource": [ "arn:aws:s3:::*" ]  
        },  
        {  
            "Sid": "AllowRootLevelListingOfCompanyBucket",  
            "Action": [ "s3>ListBucket" ],  
            "Effect": "Allow",  
            "Resource": [ "arn:aws:s3:::companybucket" ],  
            "Condition":{  
                "StringEquals":{  
                    "s3:prefix":[""], "s3:delimiter":["/"]  
                }  
            }  
        }  
    ]  
}
```

3. Teste as permissões atualizadas.

- a. Usando o link de login de usuário do IAM (consulte [Para fornecer um link de login para usuários do IAM \(p. 375\)](#)), entre no Console de gerenciamento da AWS.

Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.

- b. Clique no bucket que você criou para este exercício, e o console agora mostrará os itens de bucket no nível raiz. Se você clicar em alguma pasta no bucket, não poderá ver o conteúdo da pasta, pois ainda não concedeu essas permissões.



Este teste funciona quando os usuários usam o console do Amazon S3, pois, ao clicar em um bucket no console, a implementação do console envia uma solicitação que inclui o parâmetro `prefix` com uma string vazia como seu valor e o parâmetro `delimiter` com '/' como seu valor.

#### Etapa 4.3: resumo da política de grupo

O efeito final da política de grupo que você adicionou é conceder aos usuários do IAM Alice e Bob as seguintes permissões mínimas:

- Listar todos os buckets de propriedade da conta pai.
- Ver itens no nível raiz no bucket `companybucket`.

Contudo, os usuários ainda não podem fazer muito. Vamos conceder permissões específicas do usuário da seguinte maneira:

- Permitir que Alice obtenha e coloque objetos na pasta `Development`.
- Permitir que Bob obtenha e coloque objetos na pasta `Finance`.

Para permissões específicas do usuário, anexe uma política ao usuário específico, não ao grupo. Na próxima seção, você concede permissão para Alice trabalhar na pasta `Development`. Você pode repetir as etapas para conceder a Bob permissão semelhante para trabalhar na pasta `Finance`.

#### Etapa 5: conceder permissões específicas do usuário do IAM Alice

Agora concedemos permissões adicionais a Alice para que ela possa ver o conteúdo da pasta `Development` e obter e colocar objetos nessa pasta.

##### Etapa 5.1: conceder permissão ao usuário do IAM Alice para listar o conteúdo da pasta `Development`

Para que Alice liste o conteúdo da pasta `Development`, você deve aplicar uma política ao usuário Alice que concede permissão para a ação `s3>ListBucket` no bucket `companybucket`, contanto que a solicitação inclua o prefixo `Development/`. Como queremos que essa política seja aplicada somente ao usuário

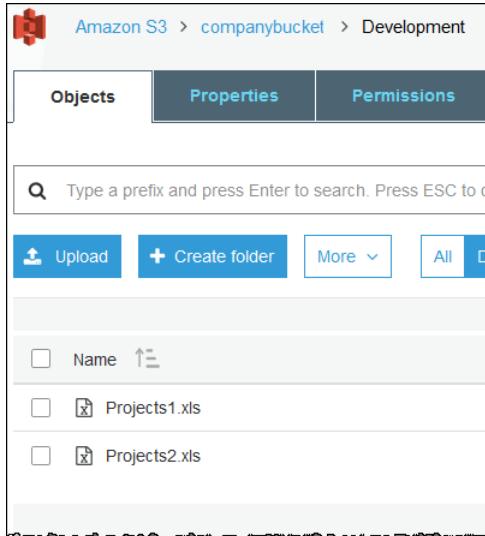
Alice, usaremos uma política em linha. Para obter mais informações sobre políticas em linha, consulte [Políticas gerenciadas e em linha](#) no Guia do usuário do IAM.

1. Faça login no Console de gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
- Use suas credenciais de conta da AWS, não as credenciais de um usuário do IAM, para entrar no console.
2. Crie uma política em linha para conceder ao usuário Alice permissão para listar o conteúdo da pasta Development.
  - a. No painel de navegação à esquerda, clique em Users (Usuários).
  - b. Clique no nome de usuário Alice.
  - c. Na página de detalhes do usuário, selecione a guia Permissions (Permissões) e, em seguida, clique em Add inline policy (Adicionar política em linha).
  - d. Clique na guia JSON.
  - e. Cole a seguinte política no campo Texto da política:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowListBucketIfSpecificPrefixIsIncludedInRequest",  
            "Action": ["s3>ListBucket"],  
            "Effect": "Allow",  
            "Resource": ["arn:aws:s3:::companybucket"],  
            "Condition": { "StringLike": {"s3:prefix": ["Development/*"] } }  
        }  
    ]  
}
```

- f. Clique em Review Policy (Revisar política). Na próxima página, insira um nome no campo Name (Nome), e depois clique em Create policy (Criar política).
3. Teste a alteração nas permissões de Alice:
  - a. Usando o link de login de usuário do IAM (consulte [Para fornecer um link de login para usuários do IAM \(p. 375\)](#)), entre no Console de gerenciamento da AWS.
  - b. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
  - c. No console do Amazon S3, verifique se Alice pode visualizar a lista de objetos na pasta Development/ no bucket.

Quando o usuário clica na pasta /Development para visualizar a lista de objetos nela, o console do Amazon S3 envia a solicitação ListObjects ao Amazon S3 com o prefixo / Development. Como o usuário recebe permissão para visualizar a lista de objetos com o prefixo Development e o delimitador '/', o Amazon S3 retorna a lista de objetos com o prefixo de chave Development/ e o console é exibido na lista.



#### Etapa 5.2: conceder permissões ao usuário do IAM Alice para obter e colocar objetos na pasta Development

Para que Alice obtenha e coloque objetos na pasta Development, ela precisa de permissão para chamar as ações s3:GetObject e s3:PutObject. As declarações de política a seguir concedem essas permissões, contanto que a solicitação inclua o parâmetro prefix com um valor de Development/.

```
{  
    "Sid": "AllowUserToReadWriteObjectData",  
    "Action": [ "s3:GetObject", "s3:PutObject" ],  
    "Effect": "Allow",  
    "Resource": [ "arn:aws:s3:::companybucket/Development/*" ]  
}
```

1. Faça login no Console de gerenciamento da AWS e abra o console da Amazon S3 em <https://console.aws.amazon.com/s3/>.

Use suas credenciais de conta da AWS, não as credenciais de um usuário do IAM, para entrar no console.

2. Edite a política em linha criada na etapa anterior.
  - a. No painel de navegação à esquerda, clique em Users (Usuários).
  - b. Clique no nome de usuário Alice.
  - c. Na página detalhes do usuário, selecione a guia Permissions (Permissões) e, em seguida, expanda a seção Inline Policies (Políticas em linha).
  - d. Clique em Edit Policy (Editar política) ao lado do nome da política que você criou na etapa anterior.
  - e. Copie a política a seguir no campo Texto de política para substituir a política existente:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowListBucketIfSpecificPrefixIsIncludedInRequest",  
            "Action": [ "s3>ListBucket" ],  
            "Effect": "Allow",  
            "Resource": "arn:aws:s3:::companybucket/Development/*"  
        }  
    ]  
}
```

```
        "Resource": ["arn:aws:s3:::companybucket"],
        "Condition": {
            "StringLike": {"s3:prefix": ["Development/*"]}
        }
    },
    {
        "Sid": "AllowUserToReadWriteObjectDataInDevelopmentFolder",
        "Action": ["s3:GetObject", "s3:PutObject"],
        "Effect": "Allow",
        "Resource": ["arn:aws:s3:::companybucket/Development/*"]
    }
]
```

3. Teste a política atualizada:

- a. Usando o link de login de usuário do IAM (consulte [Para fornecer um link de login para usuários do IAM \(p. 375\)](#)), entre no Console de gerenciamento da AWS.
- b. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
- c. No console do Amazon S3, verifique se Alice agora pode adicionar um objeto e fazer download de um objeto na pasta Development.

**Etapa 5.3: negar explicitamente permissões do usuário do IAM Alice a todas as outras pastas no bucket**

O usuário Alice agora pode listar o conteúdo do nível raiz no bucket *companybucket*. Ela também pode obter e colocar objetos na pasta Development. Se você quiser realmente limitar as permissões de acesso, poderá negar explicitamente o acesso de Alice a todas as outras pastas no bucket. Se houver alguma outra política (política de bucket ou ACL) que concede a Alice acesso a outras pastas no bucket, essa negação explícita cancelará essas permissões.

É possível adicionar a seguinte declaração à política do usuário Alice, que exige que todas as solicitações que Alice enviar ao Amazon S3 incluam o parâmetro *prefix*, cujo valor pode ser *Development/\** ou uma string vazia.

```
{
    "Sid": "ExplicitlyDenyAnyRequestsForAllOtherFoldersExceptDevelopment",
    "Action": ["s3>ListBucket"],
    "Effect": "Deny",
    "Resource": ["arn:aws:s3:::companybucket"],
    "Condition": {
        "StringNotLike": {"s3:prefix": ["Development/*", ""]},
        "Null": {"s3:prefix": false}
    }
}
```

Observe que há duas expressões condicionais no bloco *Condition*. O resultado dessas expressões condicionais é combinado usando o operador lógico AND. Se ambas as condições forem verdadeiras, o resultado da condição combinada será verdadeiro. Como o *Effect* nessa política é Deny, quando *Condition* for classificada como verdadeira, os usuários não poderão executar a *Action* especificada.

- A expressão condicional *Null* garante que as solicitações de Alice incluem o parâmetro *prefix*.

O parâmetro *prefix* requer acesso de pasta. Se você enviar uma solicitação sem o parâmetro *prefix*, o Amazon S3 retornará todas as chaves de objeto.

Se a solicitação incluir o parâmetro *prefix* com um valor nulo, a expressão avaliará como verdadeiro e, portanto, o inteiro *Condition* avaliará como verdadeiro. Você deve permitir uma string vazia como o valor do parâmetro *prefix*. Com a discussão anterior, lembre-se de que a string nula permite que Alice recupere itens de bucket no nível raiz como o console faz na discussão anterior. Para obter

mais informações, consulte [Etapa 4.2: permitir que os usuários listem o conteúdo do nível raiz de um bucket \(p. 378\)](#).

- A expressão condicional `StringNotLike` garante que, se o valor do parâmetro `prefix` for especificado e não for `Development/*`, a solicitação falhará.

Siga as etapas na seção anterior e atualize novamente a política em linha criada para o usuário Alice.

Copie a política a seguir no campo Texto de política para substituir a política existente:

```
{  
    "Statement": [  
        {  
            "Sid": "AllowListBucketIfSpecificPrefixIsIncludedInRequest",  
            "Action": ["s3>ListBucket"],  
            "Effect": "Allow",  
            "Resource": ["arn:aws:s3:::companybucket"],  
            "Condition": {  
                "StringLike": {"s3:prefix": ["Development/*"]} }  
        },  
        {  
            "Sid": "AllowUserToReadWriteObjectDataInDevelopmentFolder",  
            "Action": ["s3GetObject", "s3PutObject"],  
            "Effect": "Allow",  
            "Resource": ["arn:aws:s3:::companybucket/Development/*"]  
        },  
        {  
            "Sid": "ExplicitlyDenyAnyRequestsForAllOtherFoldersExceptDevelopment",  
            "Action": ["s3>ListBucket"],  
            "Effect": "Deny",  
            "Resource": ["arn:aws:s3:::companybucket"],  
            "Condition": {  
                "StringNotLike": {"s3:prefix": ["Development/*", ""]},  
                "Null": {"s3:prefix": false} }  
        }  
    ]  
}
```

## Etapa 6: conceder permissões específicas do usuário do IAM Bob

Agora você quer conceder a Bob permissão para a pasta Finance. Siga as etapas que você usou anteriormente para conceder permissões a Alice, mas substitua a pasta Development pela pasta Finance. Para obter instruções detalhadas, consulte [Etapa 5: conceder permissões específicas do usuário do IAM Alice \(p. 383\)](#).

## Etapa 7: proteger a pasta Private

Neste exemplo, você tem apenas dois usuários. Você concedeu todas as permissões mínimas necessárias no nível do grupo e concedeu permissões no nível do usuário somente quando realmente precisou de permissões no nível do usuário individual. Essa abordagem ajuda a minimizar o esforço de gerenciamento de permissões. Conforme o número de usuários aumenta, gerenciar permissões pode ser um problema. Por exemplo, não queremos que nenhum dos usuários neste exemplo acessasse o conteúdo da pasta Private. Como garantir que você não conceda accidentalmente uma permissão de usuário? Adicione uma política que negue explicitamente acesso à pasta. Uma negação explícita substitui todas as outras permissões. Para garantir que a pasta Private permaneça privada, você pode adicionar as duas declarações de negação a seguir à política de grupo:

- Adicione a seguinte declaração para negar explicitamente qualquer ação em recursos na pasta `Private` (`companybucket/Private/*`).

```
{  
    "Sid": "ExplicitDenyAccessToPrivateFolderToEveryoneInTheGroup",  
    "Action": ["s3:*"],  
    "Effect": "Deny",  
    "Resource": ["arn:aws:s3:::companybucket/Private/*"]  
}
```

- Você também nega permissão para a ação de listar objetos quando a solicitação especifica o prefixo `Private/`. No console, se Bob ou Alice clicar duas vezes na pasta `Private`, essa política fará Amazon S3 retornar uma resposta de erro.

```
{  
    "Sid": "DenyListBucketOnPrivateFolder",  
    "Action": ["s3>ListBucket"],  
    "Effect": "Deny",  
    "Resource": ["arn:aws:s3:::*"],  
    "Condition":{  
        "StringLike": {"s3:prefix": ["Private/"]} }  
}
```

Substitua a política do grupo Consultores por uma política atualizada que inclua as declarações de negação anteriores. Após a política atualizada ser aplicada, nenhum usuário no grupo poderá acessar a pasta `Private` em seu bucket.

1. Faça login no Console de gerenciamento da AWS e abra o console da Amazon S3 em <https://console.aws.amazon.com/s3/>.

Use suas credenciais de conta da AWS, não as credenciais de um usuário do IAM, para entrar no console.

2. Substitua a política gerenciada existente `AllowGroupToSeeBucketListInTheConsole` que está anexada ao grupo Consultores pela política a seguir. Lembre-se de substituir `companybucket` na política pelo nome do bucket.

Para obter instruções, consulte [Editar políticas gerenciadas pelo cliente](#) no Guia do usuário do IAM. Ao seguir as instruções, não se esqueça de seguir as orientações para aplicar alterações em todas as entidades principais às quais a política está anexada.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowGroupToSeeBucketListAndAlsoAllowGetBucketLocationRequiredForListBucket",  
            "Action": ["s3>ListAllMyBuckets", "s3:GetBucketLocation"],  
            "Effect": "Allow",  
            "Resource": ["arn:aws:s3:::*"]  
        },  
        {  
            "Sid": "AllowRootLevelListingOfCompanyBucket",  
            "Action": ["s3>ListBucket"],  
            "Effect": "Allow",  
            "Resource": ["arn:aws:s3:::companybucket"],  
            "Condition":{  
                "StringEquals": {"s3:prefix": [""]} }  
        },  
        {  
            "Sid": "RequireFolderStyleList",  
            "Action": ["s3>ListBucket"],  
            "Effect": "Deny",  
            "Resource": ["arn:aws:s3:::companybucket"],  
            "Condition":{  
                "StringNotEquals": {"s3:prefix": [""]} }  
        }  
    ]  
}
```

```
"Action": ["s3>ListBucket"],
"Effect": "Deny",
"Resource": ["arn:aws:s3::::*"],
"Condition": {
    "StringNotEquals": {"s3:delimiter": "/"}
},
{
    "Sid": "ExplicitDenyAccessToPrivateFolderToEveryoneInTheGroup",
    "Action": ["s3:*"],
    "Effect": "Deny",
    "Resource": ["arn:aws:s3::::companybucket/Private/*"]
},
{
    "Sid": "DenyListBucketOnPrivateFolder",
    "Action": ["s3>ListBucket"],
    "Effect": "Deny",
    "Resource": ["arn:aws:s3::::*"],
    "Condition": {
        "StringLike": {"s3:prefix": ["Private/"]}
    }
}
]
```

## Limpeza

Para limpar, acesse o console do IAM e remova os usuários Alice e Bob. Para obter instruções detalhadas, acesse [Exclusão de um usuário do IAM](#) no Guia do usuário do IAM.

Para garantir que você não seja mais cobrado pelo armazenamento, exclua também os objetos e o bucket que criou para este exercício.

## Recursos relacionados

- [Trabalhar com políticas](#) no Guia do usuário do IAM.

# Gerenciar o acesso com ACLs

## Tópicos

- [Visão geral da Lista de controle de acesso \(ACL\) \(p. 390\)](#)
- [Gerenciar ACLs \(p. 396\)](#)

As listas de controle de acesso (ACLs) são uma das opções da política de acesso baseada em recurso (consulte [Visão geral do gerenciamento de acesso \(p. 283\)](#)) que pode ser usada para gerenciar o acesso aos buckets e objetos. Use as ACLs para conceder permissões básicas de leitura/gravação a outras contas da AWS. Existem limites para gerenciar permissões usando ACLs. Por exemplo, é possível conceder permissões apenas para outras contas da AWS; você não pode conceder permissões para usuários da sua conta. Não é possível conceder permissões condicionais, nem negar permissões explicitamente. As ACLs são adequadas para cenários específicos. Por exemplo, se um proprietário do bucket permite que outras contas da AWS carreguem objetos, as permissões para esses objetos só podem ser gerenciadas com a ACL do objeto pela conta da AWS que detém o objeto.

Leia os tópicos introdutórios a seguir que explicam os conceitos básicos e as opções disponíveis para gerenciar o acesso aos recursos do Amazon S3, e fornece as diretrizes sobre quando usar cada opção da política de acesso.

- [Introdução ao gerenciamento de permissões de acesso aos recursos do Amazon S3 \(p. 282\)](#)
- [Diretrizes para usar as opções disponíveis de política de acesso \(p. 293\)](#)

## Visão geral da Lista de controle de acesso (ACL)

## Tópicos

- [Quem é o favorecido? \(p. 391\)](#)
- [Quais permissões posso conceder? \(p. 393\)](#)
- [Amostra de ACL \(p. 394\)](#)
- [ACL pré-configurada \(p. 395\)](#)
- [Como especificar uma ACL \(p. 396\)](#)

As listas de controle de acesso (ACLs) do Amazon S3 permitem o gerenciamento do acesso aos buckets e objetos. Cada bucket e objeto tem uma ACL anexada como um sub-recurso. Ela define a quais contas ou grupos da AWS é concedido acesso e o tipo de acesso. Quando um recurso é solicitado, o Amazon S3 consulta a ACL correspondente para verificar se o solicitante tem as permissões de acesso necessárias.

Quando você cria um bucket ou um objeto, o Amazon S3 cria uma ACL padrão que concede ao proprietário do recurso controle total sobre o recurso. Isso é exibido no seguinte exemplo de ACL de bucket (a ACL de objeto padrão tem a mesma estrutura):

### Example

```
<?xml version="1.0" encoding="UTF-8"?>
<AccessControlPolicy xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Owner>
    <ID>*** Owner-Canonical-User-ID ***</ID>
    <DisplayName>owner-display-name</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
xsi:type="Canonical User">
<ID>*** Owner-Canonical-User-ID ***</ID>
<DisplayName>display-name</DisplayName>
</Grantee>
<Permission>FULL_CONTROL</Permission>
</Grant>
</AccessControlList>
</AccessControlPolicy>
```

A ACL de exemplo inclui um elemento `Owner` que identifica o proprietário pelo ID de usuário canônico da conta da AWS. Para obter instruções sobre como localizar seu ID canônico do usuário, consulte [Encontrar o ID de usuário canônico da conta da AWS \(p. 391\)](#). O elemento `Grant` identifica o favorecido (uma conta da AWS ou um grupo predefinido) e a permissão concedida. Esta ACL padrão tem um elemento `Grant` para o proprietário. Conceda permissões adicionando elementos `Grant`, com cada concessão identificando o favorecido e a permissão.

#### Note

Uma ACL pode ter até 100 concessões.

## Quem é o favorecido?

O favorecido pode ser uma conta da AWS ou um dos grupos predefinidos do Amazon S3. Você concede permissão a uma conta da AWS usando o endereço de e-mail ou o ID de usuário canônico. No entanto, se você fornecer um endereço de e-mail na solicitação de concessão, o Amazon S3 encontrará o ID de usuário canônico para essa conta e o adicionará à ACL. As ACLs resultantes sempre conterão o ID de usuário canônico para a conta da AWS, e não o endereço de e-mail da conta.

#### Important

O uso de endereços de e-mail para especificar um favorecido tem suporte somente nas seguintes regiões da AWS:

- Leste dos EUA (Norte da Virgínia)
- Oeste dos EUA (Norte da Califórnia)
- Oeste dos EUA (Oregon)
- Ásia-Pacífico (Cingapura)
- Ásia-Pacífico (Sydney)
- Ásia-Pacífico (Tóquio)
- UE (Irlanda)
- América do Sul (São Paulo)

Para obter uma lista de todas as regiões e endpoints em que o Amazon S3 tem suporte, consulte [Regiões e endpoints](#) na Referência geral da AWS.

#### Warning

Ao conceder acesso aos recursos para outras contas da AWS, esteja ciente de que as contas da AWS podem delegar as permissões para usuários das suas contas. Isso é conhecido como acesso entre contas. Para obter informações sobre o uso do acesso de conta cruzada, consulte [Criar uma função para conceder permissões a um usuário do IAM](#) no Guia do usuário do IAM.

## Encontrar o ID de usuário canônico da conta da AWS

O ID de usuário canônico está associado à conta da AWS. É uma string longa, como `79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be`. Para obter

informações sobre como encontrar o ID de usuário canônico da conta, consulte [Encontrar o ID de usuário canônico da conta](#).

Você também pode pesquisar o ID de usuário canônico de uma conta da AWS lendo a ACL de um bucket ou objeto para o qual a conta da AWS tem permissões de acesso. Quando uma conta individual da AWS recebe permissões por meio de uma solicitação de concessão, uma entrada de concessão é adicionada à ACL com o ID de usuário canônico da conta da AWS.

**Note**

Se você tornar seu bucket público (não recomendado) qualquer usuário não autenticado pode carregar objetos para o bucket. Esses usuários anônimos não têm a conta AWS. Quando um usuário anônimo carrega um objeto em seu bucket, Amazon S3 adiciona um ID de usuário canônico especial (`65a011a29cdf8ec533ec3d1ccaae921c`) como o dono do objeto no ACL.

## Grupos predefinidos do Amazon S3

O Amazon S3 tem um conjunto de grupos predefinidos. Ao conceder acesso de conta a um grupo, especifique um dos URLs em vez do ID de usuário canônico. Fornecemos os seguintes grupos predefinidos:

- Grupo Usuários autenticados – representado por `http://acs.amazonaws.com/groups/global/AuthenticatedUsers`.

Este grupo representa todas as contas da AWS. A permissão de acesso a esse grupo permite que qualquer conta da AWS acesse o recurso. No entanto, todas as solicitações devem estar assinadas (autenticadas).

**Warning**

Quando você concede acesso ao grupo Authenticated Users, qualquer usuário autenticado da AWS em todo o mundo pode acessar seu recurso.

- Grupo Todos os usuários – representado por `http://acs.amazonaws.com/groups/global/AllUsers`.

A permissão de acesso a esse grupo permite que qualquer um acesse o recurso. As solicitações podem estar assinadas (autenticadas) ou não (anônimas). As solicitações não assinadas omitem o cabeçalho Autenticação na solicitação.

**Warning**

Recomendamos fortemente que você nunca conceda ao grupo All Users permissões WRITE, WRITE\_ACP ou FULL\_CONTROL. Por exemplo, as permissões WRITE permitem que qualquer pessoa armazene objetos em seu bucket, pelo que você é cobrado. Isso também permite que outras pessoas excluam objetos que você pode querer manter. Para obter mais informações sobre essas permissões, consulte a seguinte seção [Quais permissões posso conceder? \(p. 393\)](#).

- Grupo Entrega de logs – representado por `http://acs.amazonaws.com/groups/s3/LogDelivery`.

A permissão WRITE em um bucket permite que esse grupo grave logs de acesso ao servidor (consulte [Registro em log de acesso ao servidor Amazon S3 \(p. 625\)](#)) no bucket.

**Note**

Ao usar ACLs, um favorecido pode ser uma conta da AWS ou um dos grupos predefinidos do Amazon S3. No entanto, o favorecido não pode ser um usuário do IAM. Para obter mais informações sobre os usuários e as permissões da AWS no IAM, acesse [Usar o AWS Identity and Access Management](#).

## Quais permissões posso conceder?

A tabela a seguir lista o conjunto de permissões para as quais o Amazon S3 oferece suporte em uma ACL. O conjunto de permissões da ACL é o mesmo para a ACL de objetos e para a ACL de bucket. No entanto, dependendo do contexto (ACL de buckets ou ACL de objetos), essas permissões da ACL concedem permissão para operações de bucket ou objeto específicas. A tabela lista as permissões e descreve seus significados no contexto de objetos e buckets.

Permissão	Quando concedida em um bucket	Quando concedida em um objeto
READ	Permite ao favorecido listar os objetos no bucket	Permite ao favorecido ler os dados do objeto e seus metadados
WRITE	Permite ao favorecido criar, sobrescrever e excluir qualquer objeto do bucket	Não aplicável
READ_ACP	Permite ao favorecido ler a ACL do bucket	Permite ao favorecido ler a ACL do objeto
WRITE_ACP	Permite ao favorecido gravar a ACL para o bucket aplicável	Permite ao favorecido gravar a ACL para o objeto aplicável
FULL_CONTROL	Concede ao favorecido as permissões READ, WRITE, READ_ACP e WRITE_ACP no bucket	Concede ao favorecido as permissões READ, READ_ACP e WRITE_ACP no objeto

### Warning

Tenha cuidado ao conceder permissões de acesso a buckets e objetos do S3. Por exemplo, a concessão de acesso `WRITE` a um bucket permite que o favorecido crie, substitua e exclua qualquer objeto no bucket. É altamente recomendável que você leia esta seção [Visão geral da Lista de controle de acesso \(ACL\) \(p. 390\)](#) inteira antes da concessão de permissões.

## Mapeamento das permissões da ACL e das permissões da política de acesso

Conforme mostrado na tabela anterior, uma ACL concede apenas um conjunto finito de permissões, em comparação com o número de permissões que pode ser definido em uma política de acesso (consulte [Especificação de permissões em uma política \(p. 330\)](#)). Cada uma dessas permissões permite uma ou mais operações do Amazon S3.

A tabela a seguir mostra como cada uma das permissões da ACL se correlaciona com as permissões de política de acesso correspondentes. Como você pode ver, a política de acesso permite mais permissões que a ACL. Use a ACL principalmente para conceder permissões básicas de leitura/gravação, similares às permissões de sistema de arquivos. Para obter mais informações sobre quando usar a ACL, consulte [Diretrizes para usar as opções disponíveis de política de acesso \(p. 293\)](#).

Permissão da ACL	Permissões correspondentes da política de acesso quando permissões da ACL são concedidas em um bucket	Permissões correspondentes da política de acesso quando permissões da ACL são concedidas em um objeto
READ	<code>s3&gt;ListBucket</code> , <code>s3&gt;ListBucketVersions</code> , e <code>s3&gt;ListBucketMultipartUploads</code>	<code>s3GetObject</code> , <code>s3GetObjectVersion</code> , e <code>s3GetObjectTorrent</code>
WRITE	<code>s3PutObject</code> e <code>s3DeleteObject</code> .  Além disso, quando o favorecido é o proprietário do bucket, conceder	Não aplicável

Permissão da ACL	Permissões correspondentes da política de acesso quando permissões da ACL são concedidas em um bucket	Permissões correspondentes da política de acesso quando permissões da ACL são concedidas em um objeto
	a permissão WRITE em uma ACL do bucket permite que a ação s3:DeleteObjectVersion seja executada em qualquer versão naquele bucket.	
READ_ACP	s3:GetBucketAcl	s3:GetObjectAcl e s3:GetObjectVersionAcl
WRITE_ACP	s3:PutBucketAcl	s3:PutObjectAcl e s3:PutObjectVersionAcl
FULL_CONTROL	Equivalente a conceder as permissões READ, WRITE, READ_ACP e WRITE_ACP da ACL. Assim, essa permissão da ACL equivale à combinação das permissões correspondentes da política de acesso.	Equivalente a conceder as permissões READ, READ_ACP e WRITE_ACP da ACL. Assim, essa permissão da ACL equivale à combinação das permissões correspondentes da política de acesso.

## Amostra de ACL

A seguir, a amostra de ACL em um bucket identifica o proprietário do recurso e um conjunto de concessões. O formato é a representação XML de uma ACL na API REST do Amazon S3. O proprietário do bucket tem FULL\_CONTROL sobre o recurso. Além disso, a ACL mostra como as permissões são concedidas em um recurso para duas contas da AWS, identificadas pelo ID de usuário canônico, e para dois grupos predefinidos do Amazon S3 discutidos na seção anterior.

### Example

```
<?xml version="1.0" encoding="UTF-8"?>
<AccessControlPolicy xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Owner>
    <ID>Owner-canonical-user-ID</ID>
    <DisplayName>display-name</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
        <ID>Owner-canonical-user-ID</ID>
        <DisplayName>display-name</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>

    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
        <ID>user1-canonical-user-ID</ID>
        <DisplayName>display-name</DisplayName>
      </Grantee>
      <Permission>WRITE</Permission>
    </Grant>

    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
```

```

<ID>user2-canonical-user-ID</ID>
<DisplayName>display-name</DisplayName>
</Grantee>
<Permission>READ</Permission>
</Grant>

<Grant>
<Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
<URI>http://acs.amazonaws.com/groups/global/AllUsers</URI>
</Grantee>
<Permission>READ</Permission>
</Grant>
<Grant>
<Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
<URI>http://acs.amazonaws.com/groups/s3/LogDelivery</URI>
</Grantee>
<Permission>WRITE</Permission>
</Grant>

</AccessControlList>
</AccessControlPolicy>

```

## ACL pré-configurada

O Amazon S3 oferece suporte a um conjunto de concessões predefinidas, conhecidas como ACLs pré-configuradas. Cada ACL pré-configurada tem um conjunto predefinido de concessões e permissões. A tabela a seguir lista o conjunto de ACLs pré-configuradas e as concessões predefinidas associadas.

ACL pré-configurada	Aplica-se a	Permissões adicionadas à ACL
<code>private</code>	Bucket e objeto	O proprietário obtém <b>FULL_CONTROL</b> . Ninguém mais tem direitos de acesso (padrão).
<code>public-read</code>	Bucket e objeto	O proprietário obtém <b>FULL_CONTROL</b> . O grupo <code>AllUsers</code> (consulte <a href="#">Quem é o favorecido? (p. 391)</a> ) obtém acesso <code>READ</code> .
<code>public-read-write</code>	Bucket e objeto	O proprietário obtém <b>FULL_CONTROL</b> . O grupo <code>AllUsers</code> obtém os acessos <code>READ</code> e <code>WRITE</code> . Essa concessão não costuma ser recomendada em um bucket.
<code>aws-exec-read</code>	Bucket e objeto	O proprietário obtém <b>FULL_CONTROL</b> . O Amazon EC2 obtém o acesso <code>READ</code> para GET um pacote da Imagem de máquina da Amazon (AMI) a partir do Amazon S3.
<code>authenticated-read</code>	Bucket e objeto	O proprietário obtém <b>FULL_CONTROL</b> . O grupo <code>AuthenticatedUsers</code> obtém acesso <code>READ</code> .
<code>bucket-owner-read</code>	Objeto	O proprietário do objeto obtém <b>FULL_CONTROL</b> . O proprietário do bucket obtém acesso <code>READ</code> . Se você especificar essa ACL pré-configurada ao criar um bucket, o Amazon S3 a ignorará.
<code>bucket-owner-full-control</code>	Objeto	Os proprietários do objeto e do bucket obtêm <b>FULL_CONTROL</b> sobre o objeto. Se você especificar essa ACL pré-configurada ao criar um bucket, o Amazon S3 a ignorará.
<code>log-delivery-write</code>	Bucket	O grupo <code>LogDelivery</code> obtém as permissões <code>WRITE</code> e <code>READ_ACP</code> no bucket. Para obter mais informações sobre

ACL pré-configurada	Aplica-se a	Permissões adicionadas à ACL
		logs, consulte ( <a href="#">Registro em log de acesso ao servidor Amazon S3 (p. 625)</a> ).

#### Note

Você pode especificar apenas uma dessas ACLs pré-configuradas na solicitação.

Especifique uma ACL pré-configurada na solicitação usando o cabeçalho de solicitação `x-amz-acl`. Quando o Amazon S3 recebe uma solicitação com uma ACL pré-configurada, ele adiciona as concessões predefinidas à ACL do recurso.

## Como especificar uma ACL

As APIs do Amazon S3 permitem definir uma ACL quando você cria um bucket ou um objeto. O Amazon S3 também fornece uma API para definir ACL em um bucket ou um objeto existente. Estas APIs oferecem os seguintes métodos para definir uma ACL:

- Definir ACL usando cabeçalhos de solicitação— ao enviar uma solicitação para criar um recurso (bucket ou objeto), defina uma ACL usando os cabeçalhos de solicitação. Com esses cabeçalhos, você pode especificar uma ACL pré-configurada ou especificar concessões explicitamente (identificando o favorecido e as permissões de maneira explícita).
- Definir ACL usando o corpo da solicitação— ao enviar uma solicitação para definir uma ACL em um recurso existente, defina a ACL no cabeçalho da solicitação ou no corpo.

Para obter mais informações, consulte [Gerenciar ACLs \(p. 396\)](#).

## Gerenciar ACLs

### Tópicos

- [Gerenciar ACLs no Console de gerenciamento da AWS \(p. 396\)](#)
- [Gerenciar ACLs usando o AWS SDK for Java \(p. 396\)](#)
- [Gerenciar ACLs usando o AWS SDK para .NET \(p. 399\)](#)
- [Gerenciar ACLs usando a API REST \(p. 402\)](#)

Existem diversas formas para adicionar concessões à ACL do recurso. Use o Console de gerenciamento da AWS, o qual fornece uma IU para gerenciar permissões sem escrever códigos. Você pode usar a API REST ou um dos SDKs da AWS. Essas bibliotecas simplificam ainda mais as tarefas de programação.

## Gerenciar ACLs no Console de gerenciamento da AWS

O Console de gerenciamento da AWS fornece uma IU para você conceder permissões de acesso baseadas na ACL aos buckets e objetos. Para obter informações sobre como definir permissões de acesso com base em ACL no console, consulte [Como definir permissões de bucket de ACL?](#) e [Como definir permissões em um objeto?](#) no Guia do usuário do console do Amazon Simple Storage Service.

## Gerenciar ACLs usando o AWS SDK for Java

Esta seção fornece exemplos de como configurar concessões na lista de controle de acesso (ACL) em buckets e objetos. O primeiro exemplo cria um bucket com uma ACL padrão (consulte [ACL pré-configurada \(p. 395\)](#)), cria uma lista de concessões de permissão personalizadas e, em seguida, substitui

a ACL padrão por uma ACL que contém concessões personalizadas. O segundo exemplo mostra como modificar uma ACL usando o método `AccessControlList.grantPermission()`.

## Definindo concessões da ACL

### Example

Este exemplo cria um bucket. Na solicitação, o exemplo especifica uma ACL padrão que concede permissão ao grupo de Entrega de logs para gravar logs no bucket.

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

import java.io.IOException;
import java.util.ArrayList;
import java.util.Collection;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.AccessControlList;
import com.amazonaws.services.s3.model.CannedAccessControlList;
import com.amazonaws.services.s3.model.CanonicalGrantee;
import com.amazonaws.services.s3.model.CreateBucketRequest;
import com.amazonaws.services.s3.model.Grant;
import com.amazonaws.services.s3.model.GroupGrantee;
import com.amazonaws.services.s3.model.Permission;

public class CreateBucketWithACL {

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

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

            // Create a bucket with a canned ACL. This ACL will be deleted by the
            // getGrantsAsList().clear() call below. It is here for demonstration
            // purposes.
            CreateBucketRequest createBucketRequest = new CreateBucketRequest(bucketName,
clientRegion)
                .withCannedAcl(CannedAccessControlList.LogDeliveryWrite);
            s3Client.createBucket(createBucketRequest);

            // Create a collection of grants to add to the bucket.
            Collection<Grant> grantCollection = new ArrayList<Grant>();

            // Grant the account owner full control.
            Grant grant1 = new Grant(new
CanonicalGrantee(s3Client.getSQSAccountOwner().getId()), Permission.FullControl);
            grantCollection.add(grant1);

            // Grant the LogDelivery group permission to write to the bucket.
            Grant grant2 = new Grant(GroupGrantee.LogDelivery, Permission.Write);
            grantCollection.add(grant2);

            // Save (replace) grants by deleting all current ACL grants and replacing
```

```
// them with the two we just created.  
AccessControlList bucketAcl = s3Client.getBucketAcl(bucketName);  
bucketAcl.getGrantsAsList().clear();  
bucketAcl.getGrantsAsList().addAll(grantCollection);  
s3Client.setBucketAcl(bucketName, bucketAcl);  
}  
catch(AmazonServiceException e) {  
    // The call was transmitted successfully, but Amazon S3 couldn't process  
    // it and returned an error response.  
    e.printStackTrace();  
}  
catch(SdkClientException e) {  
    // Amazon S3 couldn't be contacted for a response, or the client  
    // couldn't parse the response from Amazon S3.  
    e.printStackTrace();  
}  
}  
}  
}
```

## Configurar concessões da ACL em um objeto existente

### Example

Este exemplo atualiza a ACL em um objeto. O exemplo realiza as seguintes tarefas:

- Recupera a ACL de um objeto
- Limpa a ACL removendo todas as permissões existentes
- Adiciona duas permissões: acesso total do proprietário, e WRITE\_ACP (consulte [Quais permissões posso conceder? \(p. 393\)](#)) para o usuário identificado por endereço de e-mail
- Salva a ACL no objeto

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.  
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)  
  
import java.io.IOException;  
  
import com.amazonaws.AmazonServiceException;  
import com.amazonaws.SdkClientException;  
import com.amazonaws.auth.profile.ProfileCredentialsProvider;  
import com.amazonaws.services.s3.AmazonS3;  
import com.amazonaws.services.s3.AmazonS3ClientBuilder;  
import com.amazonaws.services.s3.model.AccessControlList;  
import com.amazonaws.services.s3.model.CanonicalGrantee;  
import com.amazonaws.services.s3.model.EmailAddressGrantee;  
import com.amazonaws.services.s3.model.Permission;  
  
public class ModifyACLExistingObject {  
  
    public static void main(String[] args) throws IOException {  
        String clientRegion = "**** Client region ****";  
        String bucketName = "**** Bucket name ****";  
        String keyName = "**** Key name ****";  
        String emailGrantee = "**** user@example.com ****";  
  
        try {  
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()  
                .withCredentials(new ProfileCredentialsProvider())  
                .withRegion(clientRegion)  
                .build();  
        }  
    }  
}
```

```
// Get the existing object ACL that we want to modify.  
AccessControlList acl = s3Client.getObjectAcl(bucketName, keyName);  
  
// Clear the existing list of grants.  
acl.getGrantsAsList().clear();  
  
// Grant a sample set of permissions, using the existing ACL owner for Full  
Control permissions.  
acl.grantPermission(new CanonicalGrantee(acl.getOwner().getId()),  
Permission.FullControl);  
acl.grantPermission(new EmailAddressGrantee(emailGrantee),  
Permission.WriteAcp);  
  
// Save the modified ACL back to the object.  
s3Client.setObjectAcl(bucketName, keyName, acl);  
}  
catch(AmazonServiceException e) {  
    // The call was transmitted successfully, but Amazon S3 couldn't process  
    // it, so it returned an error response.  
    e.printStackTrace();  
}  
catch(SdkClientException e) {  
    // Amazon S3 couldn't be contacted for a response, or the client  
    // couldn't parse the response from Amazon S3.  
    e.printStackTrace();  
}  
}  
}  
}
```

## Gerenciar ACLs usando o AWS SDK para .NET

Esta seção fornece exemplos de configuração de concessões de ACL em buckets e objetos do Amazon S3.

### Exemplo 1: criar um bucket e usar uma ACL padrão para definir permissões

Este exemplo do C# cria um bucket. Na solicitação, o código também especifica uma ACL padrão que concede permissões ao grupo de Entrega de logs para gravar os logs no bucket.

Para obter instruções sobre criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.  
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)  
  
using Amazon.S3;  
using Amazon.S3.Model;  
using System;  
using System.Threading.Tasks;  
  
namespace Amazon.DocSamples.S3  
{  
    class ManagingBucketACLTTest  
    {  
        private const string newBucketName = "*** bucket name ***";  
        // Specify your bucket region (an example region is shown).  
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;  
        private static IAmazonS3 client;
```

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

private static async Task CreateBucketUseCannedACLAsync()
{
    try
    {
        // Add bucket (specify canned ACL).
        PutBucketRequest putBucketRequest = new PutBucketRequest()
        {
            BucketName = newBucketName,
            BucketRegion = S3Region.EUW1, // S3Region.US,
                                    // Add canned ACL.
            CannedACL = S3CannedACL.LogDeliveryWrite
        };
        PutBucketResponse putBucketResponse = await
client.PutBucketAsync(putBucketRequest);

        // Retrieve bucket ACL.
        GetACLResponse getACLResponse = await client.GetACLAsync(new GetACLRequest
        {
            BucketName = newBucketName
        });
        catch (AmazonS3Exception amazonS3Exception)
        {
            Console.WriteLine("S3 error occurred. Exception: " +
amazonS3Exception.ToString());
        }
        catch (Exception e)
        {
            Console.WriteLine("Exception: " + e.ToString());
        }
    }
}
```

## Exemplo 2: configurar concessões da ACL em um objeto existente

Este exemplo do C# atualiza a ACL em um objeto existente. O exemplo realiza as seguintes tarefas:

- Recupera a ACL de um objeto.
- Limpa a ACL removendo todas as permissões existentes.
- Adiciona duas permissões: acesso total do proprietário, e WRITE\_ACP para o usuário identificado por endereço de e-mail.
- Salva a ACL enviando uma solicitação PutAcl.

Para obter instruções sobre criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-
developer-guide/blob/master/LICENSE-SAMPLECODE.)

using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
```

```
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class ManagingObjectACLTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string keyName = "*** object key name ***";
        private const string emailAddress = "*** email address ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 client;
        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            TestObjectACLTestAsync().Wait();
        }

        private static async Task TestObjectACLTestAsync()
        {
            try
            {
                // Retrieve the ACL for the object.
                GetACLResponse aclResponse = await client.GetACLAsync(new GetACLRequest
                {
                    BucketName = bucketName,
                    Key = keyName
                });

                S3AccessControlList acl = aclResponse.AccessControlList;

                // Retrieve the owner (we use this to re-add permissions after we clear
                the ACL).
                Owner owner = acl.Owner;

                // Clear existing grants.
                acl.Grants.Clear();

                // Add a grant to reset the owner's full permission (the previous clear
                statement removed all permissions).
                S3Grant fullControlGrant = new S3Grant
                {
                    Grantee = new S3Grantee { CanonicalUser = owner.Id },
                    Permission = S3Permission.FULL_CONTROL
                };

                // Describe the grant for the permission using an email address.
                S3Grant grantUsingEmail = new S3Grant
                {
                    Grantee = new S3Grantee { EmailAddress = emailAddress },
                    Permission = S3Permission.WRITE_ACP
                };
                acl.Grants.AddRange(new List<S3Grant> { fullControlGrant,
                grantUsingEmail });

                // Set a new ACL.
                PutACLResponse response = await client.PutACLAsync(new PutACLRequest
                {
                    BucketName = bucketName,
                    Key = keyName,
                    AccessControlList = acl
                });
            }
            catch (AmazonS3Exception amazonS3Exception)
            {
```

```
        Console.WriteLine("An AmazonS3Exception was thrown. Exception: " +
amazonS3Exception.ToString());
    }
    catch (Exception e)
    {
        Console.WriteLine("Exception: " + e.ToString());
    }
}
}
```

## Gerenciar ACLs usando a API REST

Para obter informações sobre o suporte da API REST para o gerenciamento de ACLs, consulte as seguintes seções no Amazon Simple Storage Service API Reference:

- [GET Bucket acl](#)
- [acl de PUT Bucket](#)
- [GET Object acl](#)
- [PUT Object acl](#)
- [Objeto PUT](#)
- [Bucket PUT](#)
- [Objeto PUT - Copiar](#)
- [Iniciar multipart upload](#)

## Usar o Amazon S3 Block Public Access

O Amazon S3 fornece configurações do Block Public Access para buckets e contas a fim de ajudar a gerenciar o acesso público aos recursos do Amazon S3. Por padrão, novos buckets e objetos não permitem acesso público, mas usuários podem modificar políticas de bucket ou permissões de objeto para permitir acesso público. O Amazon S3 Block Public Access fornece configurações que substituem essas políticas e permissões, de maneira que seja possível limitar o acesso público a esses recursos. Com o Amazon S3 Block Public Access, os administradores de conta e os proprietários de bucket podem configurar facilmente controles centralizados para limitar o acesso público aos recursos do Amazon S3 impostos, independentemente de como os recursos são criados.

Ao receber uma solicitação para acessar um bucket ou um objeto, o Amazon S3 determina se o bucket ou a conta do proprietário do bucket tem uma configuração do Block Public Access. Caso haja uma configuração do Block Public Access que proíba o acesso solicitado, o Amazon S3 rejeita a solicitação. O Amazon S3 Block Public Access fornece quatro configurações. Essas configurações são independentes e podem ser usadas em qualquer combinação, e cada configuração pode ser aplicada a um bucket ou a uma conta da AWS inteira. Caso o bucket tenha configurações do Block Public Access diferentes da conta do proprietário, o Amazon S3 aplica a combinação mais restritiva das configurações nos níveis do bucket e da conta. Por isso, ao avaliar se uma operação é proibida por uma configuração do Block Public Access setting, o Amazon S3 rejeita todas as solicitações que violariam uma configuração no nível do bucket ou no nível da conta.

### Note

- É possível habilitar as configurações do Block Public Access apenas para buckets e contas da AWS. O Amazon S3 não dá suporte às configurações do Block Public Access por objeto.
- Quando você aplica configurações do Block Public Access a uma conta, as configurações se aplicam a todas as regiões da AWS globalmente. As configurações talvez não entrem em vigor

em todas as regiões imediata ou simultaneamente, mas acabam se propagando para todas as regiões.

#### Tópicos

- [Configurações do Block Public Access \(p. 403\)](#)
- [O significado de "público" \(p. 404\)](#)
- [Permissões \(p. 406\)](#)
- [Exemplos \(p. 406\)](#)

## Configurações do Block Public Access

O Amazon S3 Block Public Access fornece quatro configurações. É possível aplicar essas configurações em qualquer combinação a buckets individuais ou a contas da AWS inteiras. Caso você aplique uma configuração a uma conta, ela se aplica a todos os buckets de propriedade dessa conta. A tabela a seguir contém as configurações disponíveis.

Nome	Descrição
BlockPublicAcls	A definição dessa opção como TRUE causa o seguinte comportamento: <ul style="list-style-type: none"><li>• As chamadas PUT Bucket acl e PUT Object falharão se a Access Control List (ACL – Lista de controle de acesso) especificada for pública.</li><li>• As chamadas PUT Object falharão se a solicitação incluir uma ACL pública.</li><li>• Se essa configuração for aplicada a uma conta, as chamadas PUT Bucket falharão se a solicitação incluir uma ACL pública.</li></ul> Quando essa configuração for definida como TRUE, as operações especificadas falharão (sejam feitas por meio da API REST, da AWS CLI ou dos SDKs da AWS). Porém, as políticas e as ACLs existentes para buckets e objetos não são modificadas. Essa configuração permite se proteger contra acesso público ao mesmo tempo em que permite auditar, refinar ou alterar as políticas e as ACLs existentes para os buckets e os objetos.
IgnorePublicAcls	A definição dessa opção como TRUE faz o Amazon S3 ignorar todas as ACLs públicas em um bucket e todos os objetos contidos. Essa configuração permite bloquear com segurança acesso público concedido por ACLs ao mesmo tempo em que permite chamadas PUT Object que incluem uma ACL pública (ao contrário de BlockPublicAcls, que rejeita chamadas PUT Object que incluem uma ACL pública). A habilitação dessa configuração não afeta a persistência de ACLs existentes nem evita a definição de novas ACLs públicas.
BlockPublicPolicy	A definição dessa opção como TRUE faz o Amazon S3 rejeitar chamadas para a política PUT Bucket caso a política do bucket especificado permita acesso público. Essa configuração permite que os usuários gerenciem políticas de bucket sem permitir que compartilhem publicamente o bucket ou os objetos contidos. A habilitação dessa configuração não afeta políticas de bucket existentes. <p style="text-align: center;"><b>Important</b></p> <p>Para usar essa configuração de maneira efetiva, aplique-a no nível da conta. Como uma política de bucket pode permitir que os usuários alterem as configurações do Block Public Access de um bucket, os usuários com permissão para alterar a política de bucket podem inserir</p>

Nome	Descrição
	uma política que os permita desabilitar as configurações do Block Public Access do bucket. Caso essa configuração esteja habilitada para toda a conta, em vez de um bucket específico, o Amazon S3 bloqueia as políticas públicas, mesmo que um usuário altere a política de bucket para desabilitar essa configuração.
RestrictPublicBucketAccess	A definição dessa opção como TRUE restringe o acesso a um bucket com uma política pública apenas a serviços da AWS e a usuários autorizados dentro da conta do proprietário do bucket. Essa definição bloqueia todo o acesso entre contas ao bucket (exceto por serviços da AWS), ao mesmo tempo em que continua permitindo que usuários dentro da conta gerenciem o bucket.  A habilitação dessa configuração não afeta políticas de bucket existentes, exceto se o Amazon S3 bloquear os acessos público e entre contas derivados de qualquer política de bucket público, inclusive delegação não pública a contas específicas.

#### Important

- As chamadas para GET Bucket acl e GET Object acl sempre retornam as permissões efetivas implantadas para o bucket ou o objeto especificado. Por exemplo, suponhamos que um bucket tenha uma ACL que conceda acesso público, mas o bucket também tenha a configuração IgnorePublicAcls habilitada. Nesse caso, GET Bucket acl retorna uma ACL refletindo as permissões de acesso que o Amazon S3 está impondo, em vez da ACL real associada ao bucket.
- Como as configurações do Block Public Access não alteram as políticas ou as ACLs existentes, a remoção de uma configuração do Block Public Access disponibiliza novamente um bucket ou um objeto com uma política pública ou uma ACL.

## O significado de "público"

### ACLs

O Amazon S3 considerará uma ACL de bucket de objeto pública se ela conceder alguma permissão a membros dos grupos AllUsers ou AuthenticatedUsers predefinidos. Para obter mais informações sobre grupos predefinidos, consulte [Grupos predefinidos do Amazon S3 \(p. 392\)](#).

### Políticas

Ao avaliar uma política de bucket, o Amazon S3 começa pressupondo que a política seja pública e acaba avaliando a política para determinar se ela está qualificada como não pública. Para ser considerada não pública, uma política de bucket só deve conceder acesso a valores fixos (valores que não contenham um curinga) de um ou mais dos seguintes:

- Um conjunto de Classless Inter-Domain Routings (CIDRs – Roteamentos sem classe entre domínios) que use aws:SourceIp. Para obter mais informações sobre o CIDR, consulte [RFC 4632](#) no site RFC Editor.
- Uma entidade principal, um usuário, uma função ou uma entidade principal de serviço da AWS
  - aws:SourceArn
  - aws:SourceVpc
  - aws:SourceVpce
  - aws:SourceOwner
  - aws:SourceAccount
  - s3:x-amz-server-side-encryption-aws-kms-key-id

- aws:userid, fora do padrão "AROLEID:\*

Nessas regras, as seguintes políticas de exemplo são consideradas públicas:

```
{  
    "Principal": { "Federated": "graph.facebook.com" },  
    "Resource": "*",  
    "Action": "s3:PutObject",  
    "Effect": "Allow"  
}
```

```
{  
    "Principal": "*",  
    "Resource": "*",  
    "Action": "s3:PutObject",  
    "Effect": "Allow"  
}
```

```
{  
    "Principal": "*",  
    "Resource": "*",  
    "Action": "s3:PutObject",  
    "Effect": "Allow",  
    "Condition": { "StringLike": {"aws:SourceVpc": "vpc-*"} }  
}
```

Essas políticas podem ser tornadas não públicas com a inclusão de alguma das chaves de condição listadas anteriormente usando-se um valor fixo. Por exemplo, a última política acima pode se tornar não pública com a definição de aws:SourceVpc como um valor fixo como este:

```
{  
    "Principal": "*",  
    "Resource": "*",  
    "Action": "s3:PutObject",  
    "Effect": "Allow",  
    "Condition": { "StringEquals": {"aws:SourceVpc": "vpc-91237329"} }  
}
```

Para obter mais informações sobre políticas de bucket, consulte [Uso de políticas de bucket e políticas de usuário \(p. 326\)](#).

## Exemplo

Este exemplo mostra como o Amazon S3 avalia uma política que contém concessões de acesso público e não público.

Suponhamos que um bucket tenha uma política que conceda acesso a um conjunto de entidades principais fixas. Segundo as regras descritas anteriormente, essa política não é pública. Por isso, se você habilitar a configuração `RestrictPublicBuckets`, a política permanecerá em vigor como escrita, porque `RestrictPublicBuckets` só se aplica a buckets que tenham políticas públicas. Porém, se você adicionar uma instrução pública à política, `RestrictPublicBuckets` entrará em vigor no bucket e só permitirá que entidades principais da AWS e usuários autorizados da conta do proprietário do bucket tenham acesso ao bucket.

Como exemplo, suponhamos que um bucket de propriedade de "Account-1" tenha uma política que contenha o seguinte:

1. Uma instrução que conceda acesso ao AWS CloudTrail (uma entidade principal de serviço da AWS)

2. Uma instrução que conceda acesso à conta "Account-2"
3. Uma instrução que conceda acesso ao público, por exemplo, especificando "Principal": "\*" sem limitação de Condition

Essa política é qualificada como pública por causa da terceira instrução. Com essa política implantada e `RestrictPublicBuckets` habilitado, o Amazon S3 só permite acesso pelo CloudTrail. Embora a instrução 2 não seja pública, o S3 desabilita o acesso de "Account-2". Isso porque a instrução 3 renderiza toda a política pública. Assim, `RestrictPublicBuckets` se aplica. Dessa forma, o S3 desabilita o acesso entre contas, mesmo que a política delegue acesso a uma conta específica, "Account-2". Porém, se você remover a instrução 3 da política, esta não se qualificará como pública e `RestrictPublicBuckets` deixará de se aplicar. Por isso, "Account-2" retoma o acesso ao bucket, mesmo caso você deixe `RestrictPublicBuckets` habilitado.

## Permissões

Para usar os recursos do Amazon S3 Block Public Access, você deve ter as permissões a seguir.

Operação	Permissões obrigatórias
Status da política de bucket GET	<code>s3:GetBucketPolicyStatus</code>
Configurações do Block Public Access do bucket GET	<code>s3:GetBucketPublicAccessBlock</code>
Configurações do Block Public Access do bucket PUT	<code>s3:PutBucketPublicAccessBlock</code>
Configurações do Block Public Access do bucket DELETE	<code>s3:PutBucketPublicAccessBlock</code>
Configurações do Block Public Access da conta GET	<code>s3:GetAccountPublicAccessBlock</code>
Configurações do Block Public Access da conta PUT	<code>s3:PutAccountPublicAccessBlock</code>
Configurações do Block Public Access da conta DELETE	<code>s3:PutAccountPublicAccessBlock</code>

### Note

As operações DELETE exigem as mesmas permissões das operações PUT. Não há permissões separadas para as operações DELETE.

## Exemplos

### Usar o Block Public Access com a AWS CLI

Use o Amazon S3 Block Public Access por meio da AWS CLI. O comando usado depende do desejo de realizar uma chamada do Block Public Access em um bucket ou em uma conta. Para obter mais informações sobre como configurar e usar a AWS CLI, consulte [O que é a AWS Command Line Interface?](#)

#### Bucket

Para realizar operações do Block Public Access em um bucket, use o serviço da AWS CLI `s3api`. As operações no nível do bucket que usam esse serviço são:

- PUT PublicAccessBlock (para um bucket)
- GET PublicAccessBlock (para um bucket)
- DELETE PublicAccessBlock (para um bucket)
- GET BucketPolicyStatus

#### Conta

Para realizar operações do Block Public Access em uma conta, use o serviço da AWS CLI s3control. As operações no nível da conta que usam esse serviço são:

- PUT PublicAccessBlock (para uma conta)
- GET PublicAccessBlock (para uma conta)
- DELETE PublicAccessBlock (para uma conta)

## Usar o Block Public Access com a AWS SDK for Java

Os exemplos a seguir mostram como usar o Amazon S3 Block Public Access com o AWS SDK for Java. Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Usar o AWS SDK for Java \(p. 646\)](#).

### Exemplo 1

Este exemplo mostra como definir uma configuração do Public Access Block em um bucket do S3 usando o AWS SDK for Java.

```
AmazonS3 client = AmazonS3ClientBuilder.standard()
    .withCredentials(<credentials>)
    .build();

client.setPublicAccessBlock(new SetPublicAccessBlockRequest()
    .withBucketName(<bucket-name>)
    .withPublicAccessBlockConfiguration(new PublicAccessBlockConfiguration()
        .withBlockPublicAcls(<value>)
        .withIgnorePublicAcls(<value>)
        .withBlockPublicPolicy(<value>)
        .withRestrictPublicBuckets(<value>)));
```

#### Important

Este exemplo pertence apenas a operações no nível do bucket, que usam a classe cliente AmazonS3. Para operações no nível da conta, consulte o exemplo a seguir.

### Exemplo 2

Este exemplo mostra como colocar uma configuração do Public Access Block em uma conta do S3 usando o AWS SDK for Java.

```
AWSS3ControlClientBuilder controlClientBuilder = AWSS3ControlClientBuilder.standard();
controlClientBuilder.setRegion(<region>);
controlClientBuilder.setCredentials(<credentials>);

AWSS3Control client = controlClientBuilder.build();
client.putPublicAccessBlock(new PutPublicAccessBlockRequest()
    .withAccountId(<account-id>)
    .withPublicAccessBlockConfiguration(new PublicAccessBlockConfiguration()
        .withIgnorePublicAcls(<value>))
```

```
.withBlockPublicAcls(<value>)
.withBlockPublicPolicy(<value>)
.withRestrictPublicBuckets(<value>));
```

#### Important

Este exemplo pertence apenas a operações no nível da conta, que usam a classe cliente `AWSS3Control`. Para operações no nível do bucket, consulte o exemplo anterior.

## Usar o Block Public Access com outros SDKs da AWS

Para obter informações sobre como usar os outros SDKs da AWS, consulte [Usar os AWS SDKs, a CLI e os Explorers \(p. 639\)](#).

## Usar o Block Public Access com as APIs REST

Para obter informações sobre como usar o Amazon S3 Block Public Access por meio das APIs REST, consulte os tópicos a seguir na Amazon Simple Storage Service API Reference.

- Operações no nível da conta
  - [PUT PublicAccessBlock](#)
  - [GET PublicAccessBlock](#)
  - [DELETE PublicAccessBlock](#)
- Operações no nível do bucket
  - [PUT PublicAccessBlock](#)
  - [GET PublicAccessBlock](#)
  - [DELETE PublicAccessBlock](#)
  - [GET BucketPolicyStatus](#)

# Proteção de dados no Amazon S3

## Tópicos

- [Proteção de dados usando criptografia \(p. 409\)](#)
- [Usar versionamento \(p. 448\)](#)
- [Introdução ao Amazon S3 Object Lock \(p. 470\)](#)

O Amazon S3 fornece uma infraestrutura de armazenamento resiliente, projetada para armazenamento de dados físico de missão crítica e primários. Os objetos são armazenados, de maneira redundante, em vários dispositivos de diversas instalações em uma região do Amazon S3. Para garantir uma melhor durabilidade dos dados, as operações `PUT` e `PUT Object copy` do Amazon S3 armazenam, sincronamente, os dados em diversas instalações antes de retornar `SUCCESS`. Assim que os objetos são armazenados, o Amazon S3 mantém sua durabilidade ao detectar e reparar, rapidamente, qualquer redundância perdida.

O Amazon S3 também verifica, regularmente, a integridade dos dados armazenados usando somas de verificação. Se o Amazon S3 detectar uma corrupção, ela será reparada usando dados redundantes. Além disso, o Amazon S3 calcula somas de verificação de todo o tráfego da rede, para detectar corrupção de pacotes de dados durante o armazenamento ou a recuperação dos dados.

O armazenamento padrão do Amazon S3 é:

- Respalhado pelo [Acordo de Nível de Serviço do Amazon S3](#)
- Projetado para fornecer 99,999999999% de durabilidade e 99,99% de disponibilidade dos objetos em um determinado ano
- Projetado para sustentar a perda simultânea de dados em duas instalações

O Amazon S3 protege ainda mais seus dados usando o versionamento. O versionamento pode ser usado para preservar, recuperar e restaurar todas as versões de cada objeto armazenado no bucket do Amazon S3. Com o versionamento, você pode se recuperar, facilmente, de ações não intencionais do usuário e de falhas de aplicativo. Por padrão, as solicitações recuperam a versão gravada mais recente. Você pode recuperar as versões mais antigas de um objeto, especificando uma versão do objeto em uma solicitação.

# Proteção de dados usando criptografia

## Tópicos

- [Proteção de dados usando criptografia no lado do servidor \(p. 410\)](#)
- [Proteger dados usando criptografia no lado do cliente \(p. 440\)](#)

Proteção de dados protege os dados em trânsito (à medida que são transferidos para e do Amazon S3) e em repouso (enquanto estão armazenados em discos em datacenters do Amazon S3). Você pode proteger os dados em trânsito, utilizando SSL ou a criptografia no lado do cliente. Você tem as seguintes opções de proteção de dados em repouso no Amazon S3.

- Usar criptografia no lado do servidor – Você pede ao Amazon S3 para criptografar o objeto antes de salvá-lo em discos em seus datacenters e descriptografá-lo ao fazer download dos objetos.

- Usar criptografia no lado do cliente – Você pode criptografar dados do cliente e fazer upload dos dados criptografados no Amazon S3. Nesse caso, você gerencia o processo de criptografia, as chaves de criptografia e as ferramentas relacionadas.

## Proteção de dados usando criptografia no lado do servidor

A criptografia de servidor envolve a criptografia de dados em repouso — isto é, o Amazon S3 criptografa os dados no nível de objeto enquanto os grava em discos em seus datacenters e os descriptografa quando você os acessa. Contanto que você autentique sua solicitação e tenha permissões de acesso, não há diferença na forma de acesso aos objetos criptografados ou não criptografados. Por exemplo, se você compartilhar seus objetos usando um pre-signed URL, esse URL funcionará da mesma forma para objetos criptografados e não criptografados.

### Note

Não é possível aplicar diferentes tipos de criptografia de servidor ao mesmo objeto simultaneamente.

Você tem três opções mutuamente exclusivas, dependendo de como gerenciar as chaves de criptografia:

- Usar criptografia no lado do servidor com as chaves gerenciadas pelo Amazon S3 (SSE-S3) – Cada objeto é criptografado com uma chave exclusiva. Como proteção adicional, ele criptografa a própria chave com uma chave mestra que é atualizada regularmente. A criptografia do lado do servidor do Amazon S3 usa uma das cifras de blocos mais fortes disponíveis, um padrão de criptografia avançada de 256 bits (AES-256), para criptografar seus dados. Para obter mais informações, consulte [Proteger dados usando criptografia no lado do servidor com chaves de criptografia gerenciadas pelo Amazon S3 \(SSE-S3\)](#) (p. 416).
- Usar criptografia de servidor com chaves gerenciadas por AWS KMS (SSE-KMS) – Semelhante a SSE-S3, mas com alguns benefícios adicionais junto com algumas taxas adicionais para usar este serviço. Há outras permissões para uso de uma chave de envelope (ou seja, uma chave que protege a chave de criptografia de seus dados) que fornece uma maior proteção contra acesso não autorizado de seus objetos no S3. SSE-KMS também fornece uma trilha de auditoria que mostra quando sua chave foi usada e por quem. Além disso, você tem a opção de criar e gerenciar chaves de criptografia por conta própria ou usar uma chave padrão que é exclusiva, o serviço que está usando e a região em que está trabalhando. Para obter mais informações, consulte [Proteção de dados usando criptografia do servidor com chaves gerenciadas pelo AWS KMS– \(SSE-KMS\)](#) (p. 410).
- Usar criptografia de servidor com chaves fornecidas pelo cliente (SSE-C) – Você gerencia as chaves de criptografia e o Amazon S3 gerencia a criptografia, conforme as grava em discos; já a descriptografia, ao acessar seus objetos. Para obter mais informações, consulte [Proteção de dados usando criptografia de servidor com chaves de criptografia fornecidas pelo cliente \(SSE-C\)](#) (p. 425).

### Note

Quando você lista objetos em seu bucket, a API de lista retorna uma lista de todos os objetos, independentemente de estarem ou não criptografados.

## Proteção de dados usando criptografia do servidor com chaves gerenciadas pelo AWS KMS– (SSE-KMS)

A criptografia no lado do servidor protege os dados em repouso. O AWS Key Management Service (AWS KMS) é um serviço que combina hardware e software seguros e altamente disponíveis para fornecer um sistema de gerenciamento de chaves dimensionado para a nuvem. O AWS KMS usa chaves mestra do cliente (CMKs) para criptografar os objetos do Amazon S3. Você pode usar o AWS KMS por meio

do [Console de gerenciamento da AWS](#) ou de [APIs do AWS KMS](#) para criar centralmente chaves de criptografia, definir as políticas que controlam como as chaves podem ser usadas e auditar o uso de chaves para provar que estão sendo usadas corretamente. Você pode usar essas chaves para proteger seus dados em buckets do Amazon S3.

Na primeira vez em que você adiciona um objeto criptografado por SSE-KMS a um bucket em uma região, uma CMK padrão é criada automaticamente. Essa chave é usada para a criptografia por SSE-KMS, a menos que você selecione uma CMK criada separadamente com o AWS Key Management Service. Criar sua própria CMK oferece mais flexibilidade, incluindo a capacidade de criar, habilitar, desabilitar e definir controles de acesso, além de auditar as chaves de criptografia usadas para proteger seus dados.

Para obter mais informações, consulte [O que é AWS Key Management Service?](#) em AWS Key Management Service Developer Guide. Se você usar o AWS KMS, haverá cobranças adicionais para usar chaves AWS-KMS. Para obter mais informações, consulte [Definição de preço do AWS Key Management Service](#).

#### Note

Se você fizer upload ou acessar objetos criptografados por SSE-KMS, precisará usar a versão 4 do AWS Signature para maior segurança. Para obter mais informações sobre como fazer isso usando um AWS SDK, consulte [Especificação da versão do Signature em autenticação de solicitação](#).

Os destaques de SSE-KMS são:

- Você pode optar por criar e gerenciar chaves de criptografia por conta própria ou pode optar por usar sua chave de serviço padrão gerada, exclusivamente, em um cliente por serviço por nível de região.
- O ETag na resposta não é o MD5 dos dados de objeto.
- As chaves de dados usadas para criptografar os dados também são criptografadas e armazenadas com os dados protegidos.
- As chaves mestras auditáveis podem ser criadas, alteradas e desativadas no console do AWS KMS.
- Os controles de segurança do AWS KMS podem ajudá-lo a satisfazer os requisitos de conformidade relacionados à criptografia.

O Amazon S3 oferece suporte para políticas de bucket que você pode usar se precisar de criptografia de servidor para todos os objetos que estão armazenados em seu bucket. Por exemplo, a política de bucket a seguir negará permissão de upload de objeto (`s3:PutObject`) para todos se a solicitação não incluir o cabeçalho `x-amz-server-side-encryption` que solicita criptografia de servidor com SSE-KMS.

```
{  
    "Version": "2012-10-17",  
    "Id": "PutObjPolicy",  
    "Statement": [  
        {  
            "Sid": "DenyUnEncryptedObjectUploads",  
            "Effect": "Deny",  
            "Principal": "*",  
            "Action": "s3:PutObject",  
            "Resource": "arn:aws:s3:::YourBucket/*",  
            "Condition": {  
                "StringNotEquals": {  
                    "s3:x-amz-server-side-encryption": "aws:kms"  
                }  
            }  
        }  
    ]  
}
```

O Amazon S3 também oferece suporte para a chave de condição `s3:x-amz-server-side-encryption-aws-kms-key-id`, que você pode usar para solicitar uma chave KMS específica

para a criptografia de objeto. A chave KMS especificada na política deve usar o formato "arn:aws:kms:**region:acct-id:key/key-id**" .

**Note**

Quando você faz upload de um objeto, pode especificar a chave KMS usando o cabeçalho **x-amz-server-side-encryption-aws-kms-key-id**. Se o cabeçalho não estiver presente na solicitação, o Amazon S3 usará a chave KMS padrão. No entanto, o ID de chave KMS que o Amazon S3 usa para a criptografia de objeto deve corresponder ao ID de chave KMS na política. Caso contrário, o Amazon S3 negará a solicitação.

**Important**

Todas as solicitações GET e PUT para um objeto protegido pelo AWS KMS falharão se não forem feitas via SSL ou utilizando SigV4.

SSE-KMS criptografa somente os dados de objeto. Nenhum metadado de objeto é criptografado.

## Uso do AWS Key Management Service no console do gerenciamento do Amazon S3

Para obter mais informações sobre como usar chaves de criptografia gerenciadas pelo KMS no console de gerenciamento do Amazon S3, consulte [Fazer upload de objetos do S3](#) no Guia do usuário do Amazon Simple Storage Service.

## Supporte de API para AWS Key Management Service no Amazon S3

As APIs REST de criação de objeto (consulte [Especificação do AWS Key Management Service no Amazon S3 usando a API REST \(p. 415\)](#)) fornecem um cabeçalho de solicitação, **x-amz-server-side-encryption**, que você pode usar para solicitar SSE-KMS com o valor de **aws:kms**. Há também **x-amz-server-side-encryption-aws-kms-key-id**, que especifica o ID da chave de criptografia mestra do AWS KMS que foi usada para o objeto. A API do Amazon S3 também oferece suporte para contexto de criptografia, com o cabeçalho **x-amz-server-side-encryption-context**.

O contexto de criptografia pode ser qualquer valor desejado, desde que o cabeçalho tenha o formato JSON com codificação Base64. No entanto, como o contexto de criptografia não é criptografado e só é registrado em log se o registro em log do AWS CloudTrail estiver ativado, ele não deve incluir informações confidenciais. Recomendamos ainda que o contexto descreva os dados que estão sendo criptografados ou descriptografados para que você possa compreender os eventos do CloudTrail produzidos pelo AWS KMS. Para obter mais informações, consulte [Contexto de criptografia](#) no AWS Key Management Service Developer Guide.

Além disso, o Amazon S3 pode anexar uma chave predefinida **aws:s3:arn** com o valor igual ao ARN do objeto para o contexto de criptografia que você fornece. Isso ocorrerá somente se a chave **aws:s3:arn** ainda não estiver no contexto de criptografia fornecido. Nesse caso, essa chave predefinida é anexada quando o Amazon S3 processa suas solicitações PUT. Se essa chave **aws:s3:arn** já estiver presente no contexto de criptografia, ela não será anexada uma segunda vez ao contexto de criptografia.

Ter essa chave predefinida como parte do contexto de criptografia significa que você pode acompanhar solicitações relevantes no CloudTrail. Assim, você poderá sempre ver o ARN do objeto do S3 que foi usado com qual chave de criptografia. Além disso, essa chave predefinida como parte do contexto de criptografia garante que o contexto de criptografia não seja idêntico entre os diferentes objetos do S3, o que fornece segurança adicional para seus objetos. Seu contexto de criptografia completo será validado para ter o mesmo valor do ARN do objeto.

As APIs do Amazon S3 a seguir são compatíveis com esses cabeçalhos de solicitação.

- Operação PUT — Ao fazer upload de dados usando a API PUT (consulte [Objeto PUT](#)), você pode especificar esses cabeçalhos de solicitação.

- Iniciar multipart upload — Ao fazer upload de objetos grandes usando a API de multipart upload, você pode especificar esses cabeçalhos. Especifique esses cabeçalhos na solicitação de inicialização (consulte [Iniciar multipart upload](#)).
- Operação POST — Ao usar uma operação POST para fazer upload de um objeto (consulte [Objeto POST](#)), em vez dos cabeçalhos de solicitação, você fornece as mesmas informações nos campos de formulário.
- Operação de cópia — Quando você copia um objeto (consulte [Objeto PUT - Copiar](#)), tem um objeto de origem e um objeto de destino. Quando você transmite cabeçalhos SSE-KMS com a operação de cópia, eles são aplicados somente ao objeto de destino.

Os AWS SDKs também fornecem APIs de wrapper para solicitar SSE-KMS com o Amazon S3.

## Especificação do AWS Key Management Service no Amazon S3 usando os AWS SDKs

### Tópicos

- [AWS SDK para Java \(p. 413\)](#)
- [AWS SDK para .NET \(p. 414\)](#)

Ao usar AWS SDKs, você pode pedir ao Amazon S3 para usar chaves de criptografia gerenciadas pelo AWS Key Management Service (AWS KMS). Esta seção fornece exemplos de uso dos AWS SDKs para Java e .NET. Para obter informações sobre outros SDKs, consulte [Código de exemplo e bibliotecas](#).

### AWS SDK para Java

Esta seção explica várias operações do Amazon S3 usando o AWS SDK para Java e como usar as chaves de criptografia gerenciadas pelo AWS KMS.

#### Operação PUT

Ao fazer upload de um objeto usando o AWS SDK para Java, você pode pedir ao Amazon S3 para usar uma chave de criptografia gerenciada pelo AWS KMS adicionando a propriedade `SSEAwsKeyManagementParams`, conforme exibido na seguinte solicitação:

```
PutObjectRequest putRequest = new PutObjectRequest(bucketName,  
    keyName, file).withSSEAwsKeyManagementParams(new SSEAwsKeyManagementParams());
```

Nesse caso, o Amazon S3 usa a chave mestra padrão (consulte [Proteção de dados usando criptografia do servidor com chaves gerenciadas pelo AWS KMS– \(SSE-KMS\) \(p. 410\)](#)). É possível criar sua própria chave e especificá-la na solicitação.

```
PutObjectRequest putRequest = new PutObjectRequest(bucketName,  
    keyName, file).withSSEAwsKeyManagementParams(new SSEAwsKeyManagementParams(keyID));
```

Para obter mais informações sobre a criação de chaves, consulte [Programar a API AWS KMS no AWS Key Management Service Developer Guide](#).

Para ver exemplos de código funcionais de upload de um objeto, consulte os seguintes tópicos. Você precisará atualizar esses exemplos de código e fornecer informações de criptografia conforme exibido no fragmento de código anterior.

- Para fazer upload de um objeto em uma única operação, consulte [Faça upload de objetos usando o AWS SDK for Java \(p. 176\)](#)
- Para um multipart upload, consulte os seguintes tópicos:

- Para saber como usar a API de multipart upload de alto nível, consulte [Fazer upload de um arquivo \(p. 188\)](#)
- Se você estiver usando a API de multipart upload de baixo nível, consulte [Fazer upload de um arquivo \(p. 192\)](#)

## Operação de cópia

Ao copiar objetos, você adiciona as mesmas propriedades de solicitação (`ServerSideEncryptionMethod` e `ServerSideEncryptionKeyManagementServiceKeyId`) para pedir ao Amazon S3 para usar uma chave de criptografia gerenciada pelo AWS KMS. Para obter mais informações sobre cópia de objetos, consulte [Cópia de objetos \(p. 219\)](#).

## Pre-signed URLs

Ao criar uma pre-signed URL para um objeto criptografado usando uma chave de criptografia gerenciada pelo AWS KMS, você deve especificar, explicitamente, a versão 4 do Signature:

```
ClientConfiguration clientConfiguration = new ClientConfiguration();
clientConfiguration.setSignerOverride("AWSS3V4SignerType");
AmazonS3Client s3client = new AmazonS3Client(
    new ProfileCredentialsProvider(), clientConfiguration);
...
```

Para ver um exemplo de código, consulte [Gerar um pre-signed URL de objeto usando o AWS SDK for Java \(p. 173\)](#).

## AWS SDK para .NET

Esta seção explica várias operações do Amazon S3 usando o AWS SDK para .NET e como usar as chaves de criptografia gerenciadas pelo AWS KMS.

### Operação PUT

Ao fazer upload de um objeto usando o AWS SDK para .NET, você pode pedir ao Amazon S3 para usar uma chave de criptografia gerenciada pelo AWS KMS adicionando a propriedade `ServerSideEncryptionMethod`, conforme exibido na seguinte solicitação:

```
PutObjectRequest putRequest = new PutObjectRequest
{
    BucketName = bucketName,
    Key = keyName,
    // other properties.
    ServerSideEncryptionMethod = ServerSideEncryptionMethod.AWSKMS
};
```

Nesse caso, o Amazon S3 usa a chave mestra padrão (consulte [Proteção de dados usando criptografia do servidor com chaves gerenciadas pelo AWS KMS– \(SSE-KMS\) \(p. 410\)](#)). É possível criar sua própria chave e especificá-la na solicitação.

```
PutObjectRequest putRequest1 = new PutObjectRequest
{
    BucketName = bucketName,
    Key = keyName,
    // other properties.
    ServerSideEncryptionMethod = ServerSideEncryptionMethod.AWSKMS,
    ServerSideEncryptionKeyManagementServiceKeyId = keyId
};
```

Para obter mais informações sobre a criação de chaves, consulte [Programar a API AWS KMS no AWS Key Management Service Developer Guide](#).

Para ver exemplos de código funcionais de upload de um objeto, consulte os seguintes tópicos. Você precisará atualizar esses exemplos de código e fornecer informações de criptografia, conforme exibido no fragmento de código anterior.

- Para fazer upload de um objeto em uma única operação, consulte [Faça upload de objetos usando o AWS SDK para .NET \(p. 177\)](#)
- Para multipart upload, consulte os seguintes tópicos:
  - Para saber como usar a API de multipart upload de alto nível, consulte [Faça upload de um arquivo para um Bucket do S3 usando o AWS SDK para .NET \(API de alto nível\). \(p. 197\)](#)
  - Para saber como usar a API de multipart upload de baixo nível, consulte [Faça upload de um arquivo para um Bucket do S3 usando o AWS SDK para .NET \(API de nível baixo\). \(p. 204\)](#)

## Operação de cópia

Ao copiar objetos, você adiciona as mesmas propriedades de solicitação (`ServerSideEncryptionMethod` e `ServerSideEncryptionKeyManagementServiceKeyId`) para pedir ao Amazon S3 para usar uma chave de criptografia gerenciada pelo AWS KMS. Para obter mais informações sobre cópia de objetos, consulte [Cópia de objetos \(p. 219\)](#).

## Pre-signed URLs

Ao criar uma pre-signed URL para um objeto criptografado usando uma chave de criptografia gerenciada pelo AWS KMS, você deve especificar, explicitamente, a versão 4 do Signature:

```
AWSConfigs.S3Config.UseSignatureVersion4 = true;
```

Para ver um exemplo de código, consulte [Gerar um pre-signed URL de objeto usando o AWS SDK para .NET \(p. 174\)](#).

## Especificação do AWS Key Management Service no Amazon S3 usando a API REST

No momento da criação do objeto — isto é, quando você faz upload de um objeto novo ou faz uma cópia de um objeto existente —, você pode especificar o uso de criptografia de servidor com chaves de criptografia gerenciadas pelo AWS KMS (SSE-KMS) para criptografar seus dados adicionando o cabeçalho `x-amz-server-side-encryption` à solicitação. Defina o valor do cabeçalho como o algoritmo de criptografia `aws:kms`. O Amazon S3 confirma se o objeto está armazenado usando SSE-KMS no lado do servidor retornando o cabeçalho da resposta `x-amz-server-side-encryption`.

As seguintes APIs de upload REST aceitam o cabeçalho de solicitação `x-amz-server-side-encryption`.

- [Objeto PUT](#)
- [Objeto PUT - Copiar](#)
- [Objeto POST](#)
- [Iniciar multipart upload](#)

Ao fazer upload de objetos grandes usando a API de multipart upload, você pode especificar SSE-KMS adicionando o cabeçalho `x-amz-server-side-encryption` à solicitação Iniciar multipart upload com o valor de `aws:kms`. Ao copiar um objeto existente, independentemente de o objeto de origem ser criptografado ou não, o objeto de destino não é criptografado, a menos que você solicite explicitamente a criptografia de servidor.

Os cabeçalhos de resposta das seguintes APIs REST retornam o cabeçalho `x-amz-server-side-encryption` quando um objeto é armazenado usando criptografia de servidor.

- [Objeto PUT](#)
- [Objeto PUT - Copiar](#)
- [Objeto POST](#)
- [Iniciar multipart upload](#)
- [Upload de parte](#)
- [Upload de parte - Copiar](#)
- [Concluir multipart upload](#)
- [Objeto Get](#)
- [Objeto Head](#)

#### Note

Os cabeçalhos de solicitação de criptografia não deverão ser enviados para solicitações GET e solicitações HEAD se o objeto usar SSE-KMS ou for exibido um erro HTTP 400 BadRequest.

## Proteger dados usando criptografia no lado do servidor com chaves de criptografia gerenciadas pelo Amazon S3 (SSE-S3)

A criptografia no lado do servidor protege os dados em repouso. O Amazon S3 criptografa cada objeto com uma chave exclusiva. Como proteção adicional, ele criptografa a própria chave com uma chave mestra que é atualizada regularmente. A criptografia do lado do servidor do Amazon S3 usa uma das cifras de blocos mais fortes disponíveis, um padrão de criptografia avançada de 256 bits (AES-256), para criptografar seus dados.

Se você precisar de criptografia no lado do servidor para todos os objetos armazenados em um bucket, use uma política de bucket. Por exemplo, a política de bucket a seguir negará permissões para fazer upload de um objeto, a menos que a solicitação não inclua o cabeçalho `x-amz-server-side-encryption` a fim de solicitar criptografia no lado do servidor:

```
{
    "Version": "2012-10-17",
    "Id": "PutObjPolicy",
    "Statement": [
        {
            "Sid": "DenyIncorrectEncryptionHeader",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::YourBucket/*",
            "Condition": {
                "StringNotEquals": {
                    "s3:x-amz-server-side-encryption": "AES256"
                }
            }
        },
        {
            "Sid": "DenyUnEncryptedObjectUploads",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::YourBucket/*",
            "Condition": {
                "Null": {
                    "s3:x-amz-server-side-encryption": "true"
                }
            }
        }
    ]
}
```

```
        }  
    }  
}
```

A criptografia no lado do servidor criptografa somente os dados de objeto, não os metadados de objeto.

## Suporte de API para criptografia no lado do servidor

Para solicitar criptografia no lado do servidor usando as APIs REST de criação de objeto, forneça o cabeçalho de solicitação `x-amz-server-side-encryption`. Para obter informações sobre as APIs REST, consulte [Especificação da criptografia no lado do servidor usando a API REST \(p. 424\)](#).

As APIs do Amazon S3 a seguir são compatíveis com este cabeçalho:

- Operações PUT — Especifique o cabeçalho de solicitação ao fazer upload de dados usando a API PUT. Para obter mais informações, consulte [Objeto PUT](#).
- Iniciar multipart upload — Especifique o cabeçalho na solicitação de inicialização ao fazer upload de objetos grandes usando a API de multipart upload. Para obter mais informações, consulte [Iniciar Multipart Upload](#).
- Operações COPY—Ao copiar um objeto, você tem um objeto de origem e um objeto de destino. Para obter mais informações, consulte [Objeto PUT - Copiar](#).

### Note

Ao usar uma operação POST para fazer upload de um objeto, em vez de fornecer o cabeçalho de solicitação, você fornece as mesmas informações nos campos de formulário. Para obter mais informações, consulte [Objeto POST](#).

Os SDKs da AWS também fornecem APIs de wrapper que você pode usar para solicitar criptografia no lado do servidor. Você também pode usar o Console de gerenciamento da AWS para fazer upload de objetos e solicitar a criptografia no lado do servidor.

### Note

Você não poderá aplicar criptografia com SSE-S3 a objetos que são carregados usando pre-signed URLs. Você pode especificar a criptografia no lado do servidor somente com o Console de gerenciamento da AWS ou um cabeçalho de solicitações HTTP. Para obter mais informações, consulte [Especificação de condições em uma política \(p. 335\)](#).

## Especificação da criptografia de servidor usando o AWS SDK for Java

Ao usar o AWS SDK for Java para carregar um objeto, você pode usar criptografia no lado do servidor para criptografar o objeto. Para solicitar a criptografia no lado do servidor, use a propriedade `ObjectMetadata` da `PutObjectRequest` para configurar o cabeçalho da solicitação `x-amz-server-side-encryption`. Ao chamar o método `putObject()` do `AmazonS3Client`, o Amazon S3 criptografa e salva os dados.

Você também pode solicitar a criptografia de servidor ao fazer upload de objetos com a API multipart upload:

- Ao usar a API de alto nível de multipart upload, você usa os métodos `TransferManager` para aplicar criptografia no lado do servidor aos objetos conforme faz upload desses objetos. Você pode usar qualquer um dos métodos de upload que assumem `ObjectMetadata` como um parâmetro. Para obter mais informações, consulte [Usar o AWS Java SDK para multipart upload \(API de alto nível\) \(p. 188\)](#).
- Ao usar a API de multipart upload de baixo nível, você especifica a criptografia de servidor ao iniciar o multipart upload. Você adiciona a propriedade `ObjectMetadata` chamando o método `InitiateMultipartUploadRequest.setObjectMetadata()`. Para obter mais informações, consulte [Fazer upload de um arquivo \(p. 192\)](#).

Você não poderá alterar diretamente o estado de criptografia de um objeto (criptografando um objeto não criptografado ou descriptografando um objeto criptografado). Para alterar o estado de criptografia de um objeto, é necessário fazer uma cópia dele especificando o estado de criptografia desejado para a cópia e, em seguida, excluir o objeto original. O Amazon S3 criptografa o objeto copiado somente se você solicitar explicitamente a criptografia no lado do servidor da solicitação. Para solicitar a criptografia do objeto copiado por meio da API Java, use a propriedade `ObjectMetadata` para especificar a criptografia no lado do servidor na `CopyObjectRequest`.

### Example Exemplo

O exemplo a seguir mostra como definir a criptografia no lado do servidor usando o AWS SDK for Java. Ele mostra como executar as seguintes tarefas:

- Fazer upload de um novo objeto usando criptografia no lado do servidor
- Alterar o estado de criptografia de um objeto (neste exemplo, criptografar um objeto anteriormente não criptografado) fazendo uma cópia do objeto
- Verificar o estado de criptografia do objeto

Para obter mais informações sobre criptografia no lado do servidor, consulte [Especificação da criptografia no lado do servidor usando a API REST \(p. 424\)](#). Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

import java.io.ByteArrayInputStream;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.internal.SSEResultBase;
import com.amazonaws.services.s3.model.CopyObjectRequest;
import com.amazonaws.services.s3.model.CopyObjectResult;
import com.amazonaws.services.s3.model.ObjectMetadata;
import com.amazonaws.services.s3.model.PutObjectRequest;
import com.amazonaws.services.s3.model.PutObjectResult;

public class SpecifyServerSideEncryption {

    public static void main(String[] args) {
        String clientRegion = "*** Client region ***";
        String bucketName = "*** Bucket name ***";
        String keyNameToEncrypt = "*** Key name for an object to upload and encrypt ***";
        String keyNameToCopyAndEncrypt = "*** Key name for an unencrypted object to be encrypted by copying ***";
        String copiedObjectKeyName = "*** Key name for the encrypted copy of the unencrypted object ***";

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

            // Upload an object and encrypt it with SSE.
            uploadObjectWithSSEEncryption(s3Client, bucketName, keyNameToEncrypt);
        }
    }
}
```

```
// Upload a new unencrypted object, then change its encryption state
// to encrypted by making a copy.
changeSSEEncryptionStatusByCopying(s3Client,
                                    bucketName,
                                    keyNameToCopyAndEncrypt,
                                    copiedObjectKeyName);
}
catch(AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
}
catch(SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}

private static void uploadObjectWithSSEEncryption(AmazonS3 s3Client, String bucketName,
String keyName) {
    String objectContent = "Test object encrypted with SSE";

    // Specify server-side encryption.
    ObjectMetadata objectMetadata = new ObjectMetadata();
    objectMetadata.setContentLength(objectContent.length());
    objectMetadata.setSSEAlgorithm(ObjectMetadata.AES_256_SERVER_SIDE_ENCRYPTION);
    PutObjectRequest putRequest = new PutObjectRequest(bucketName,
                                                       keyName,
                                                       new
                                                       ByteArrayInputStream(objectContent.getBytes()),
                                                       objectMetadata);

    // Upload the object and check its encryption status.
    PutObjectResult putResult = s3Client.putObject(putRequest);
    System.out.println("Object '" + keyName + "' uploaded with SSE.");
    printEncryptionStatus(putResult);
}

private static void changeSSEEncryptionStatusByCopying(AmazonS3 s3Client,
                                                      String bucketName,
                                                      String sourceKey,
                                                      String destKey) {
    // Upload a new, unencrypted object.
    PutObjectResult putResult = s3Client.putObject(bucketName, sourceKey, "Object
example to encrypt by copying");
    System.out.println("Unencrypted object '" + sourceKey + "' uploaded.");
    printEncryptionStatus(putResult);

    // Make a copy of the object and use server-side encryption when storing the copy.
    CopyObjectRequest request = new CopyObjectRequest(bucketName,
                                                       sourceKey,
                                                       bucketName,
                                                       destKey);
    ObjectMetadata objectMetadata = new ObjectMetadata();
    objectMetadata.setSSEAlgorithm(ObjectMetadata.AES_256_SERVER_SIDE_ENCRYPTION);
    request.setNewObjectMetadata(objectMetadata);

    // Perform the copy operation and display the copy's encryption status.
    CopyObjectResult response = s3Client.copyObject(request);
    System.out.println("Object '" + destKey + "' uploaded with SSE.");
    printEncryptionStatus(response);

    // Delete the original, unencrypted object, leaving only the encrypted copy in
Amazon S3.
    s3Client.deleteObject(bucketName, sourceKey);
```

```
        System.out.println("Unencrypted object \'" + sourceKey + "\' deleted.");
    }

    private static void printEncryptionStatus(SSEResultBase response) {
        String encryptionStatus = response.getSSEAlgorithm();
        if(encryptionStatus == null) {
            encryptionStatus = "Not encrypted with SSE";
        }
        System.out.println("Object encryption status is: " + encryptionStatus);
    }
}
```

## Especificação da criptografia de servidor usando o AWS SDK para .NET

Ao fazer upload de um objeto, você pode instruir o Amazon S3 a criptografar esse objeto. Para alterar o estado de criptografia de um objeto existente, faça uma cópia do objeto e exclua o objeto de origem. Por padrão, a operação de cópia criptografa o destino somente se você solicitar explicitamente a criptografia no lado do servidor do objeto de destino. Para especificar a criptografia no lado do servidor na `CopyObjectRequest`, adicione o seguinte:

```
ServerSideEncryptionMethod = ServerSideEncryptionMethod.AES256
```

Para obter um exemplo funcional de como copiar um objeto, consulte [Copiar um objeto do Amazon S3 em uma única operação usando o AWS SDK para .NET \(p. 221\)](#).

O exemplo a seguir faz upload de um objeto. Na solicitação, o exemplo instrui o Amazon S3 a criptografar o objeto. O exemplo então recupera metadados do objeto e verifica o método de criptografia que foi usado. Para obter informações sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

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

namespace Amazon.DocSamples.S3
{
    class SpecifyServerSideEncryptionTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string keyName = "*** key name for object created ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            WritingAnObjectAsync().Wait();
        }

        static async Task WritingAnObjectAsync()
        {
            try
            {
                var putRequest = new PutObjectRequest
                {
                    BucketName = bucketName,
```

```
        Key = keyName,
        ContentBody = "sample text",
        ServerSideEncryptionMethod = ServerSideEncryptionMethod.AES256
    };

    var putResponse = await client.PutObjectAsync(putRequest);

    // Determine the encryption state of an object.
    GetObjectMetadataRequest metadataRequest = new GetObjectMetadataRequest
    {
        BucketName = bucketName,
        Key = keyName
    };
    GetObjectMetadataResponse response = await
    client.GetObjectMetadataAsync(metadataRequest);
    ServerSideEncryptionMethod objectEncryption =
    response.ServerSideEncryptionMethod;

    Console.WriteLine("Encryption method used: {0}",
objectEncryption.ToString());
}
catch (AmazonS3Exception e)
{
    Console.WriteLine("Error encountered ***. Message:'{0}' when writing an
object", e.Message);
}
catch (Exception e)
{
    Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
}
}
}
}
```

Alteração da criptografia de servidor de um objeto existente (operação de cópia)

## Especificação da criptografia de servidor usando o AWS SDK para PHP

Este tópico mostra como usar classes da versão 3 do AWS SDK para PHP para adicionar criptografia no lado do servidor a objetos que você carrega no Amazon Simple Storage Service (Amazon S3). Pressupõe-se que você já esteja seguindo as instruções para [Usar o AWS SDK para PHP e executar exemplos do PHP \(p. 649\)](#) e tenha o AWS SDK para PHP devidamente instalado.

Para fazer upload de um objeto no Amazon S3, use o método [Aws\S3\S3Client::putObject\(\)](#). Para adicionar o cabeçalho de solicitação `x-amz-server-side-encryption` à sua solicitação de upload, especifique o parâmetro `ServerSideEncryption` com o valor `AES256`, conforme exibido no seguinte exemplo de código. Para obter informações sobre solicitações de criptografia no lado do servidor, consulte [Especificação da criptografia no lado do servidor usando a API REST \(p. 424\)](#).

```
<?php

require 'vendor/autoload.php';

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

// $filepath should be an absolute path to a file on disk.
$filepath = '*** Your File Path ***';

$s3 = new S3Client([
    'region' => 'us-west-2',
    'version' => '2006-03-01',
    'credentials' => [
        'key' => 'AKIAQHDX3PZGK3JLW3A',
        'secret' => 'w1XnDfCqBvVYRyOOGjUuIwvqkMqNtPQZGQH'
    ],
    'server_side_encryption' => 'AES256'
]);
```

```
'version' => 'latest',
'region'  => 'us-east-1'
]);

// Upload a file with server-side encryption.
$result = $s3->putObject([
    'Bucket'          => $bucket,
    'Key'             => $keyname,
    'SourceFile'      => $filepath,
    'ServerSideEncryption' => 'AES256',
]);
```

Em resposta, o Amazon S3 retorna o cabeçalho `x-amz-server-side-encryption` com o valor do algoritmo de criptografia que foi usado para criptografar os dados do objeto.

Ao fazer upload de objetos grandes usando a API de multipart upload, você pode especificar a criptografia de servidor para os objetos que estiver fazendo upload, conforme segue:

- Ao usar a API de multipart upload de baixo nível, especifique a criptografia de servidor ao chamar o método [Aws\S3\S3Client::createMultipartUpload\(\)](#). Para adicionar o cabeçalho de solicitação `x-amz-server-side-encryption` à sua solicitação, especifique a chave `array` do parâmetro `ServerSideEncryption` com o valor `AES256`. Para obter mais informações sobre a API de baixo nível de multipart upload, consulte [Usar o SDK PHP da AWS para multipart upload \(API de baixo nível\) \(p. 211\)](#).
- Ao usar a API de alto nível de multipart upload, especifique a criptografia de servidor usando o parâmetro `ServerSideEncryption` do método [CreateMultipartUpload](#). Para ver um exemplo de uso do método `setOption()` com a API de alto nível de multipart upload, consulte [Usar o SDK PHP da AWS para multipart upload \(p. 209\)](#).

### Determinar algoritmo de criptografia usado

Para determinar o estado de criptografia de um objeto existente, recupere os metadados de objeto chamando o método [Aws\S3\S3Client::headObject\(\)](#) conforme exibido no seguinte exemplo de código PHP.

```
<?php

require 'vendor/autoload.php';

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// Check which server-side encryption algorithm is used.
$result = $s3->headObject([
    'Bucket' => $bucket,
    'Key'     => $keyname,
]);
echo $result['ServerSideEncryption'];
```

### Alteração da criptografia de servidor de um objeto existente (operação de cópia)

Para alterar o estado de criptografia de um objeto existente, faça uma cópia do objeto usando o método [Aws\S3\S3Client::copyObject\(\)](#) e exclua o objeto de origem. Observe que, por padrão, `copyObject()` não

criptografará o destino, a menos que você solicite explicitamente a criptografia de servidor do objeto de destino usando o parâmetro `ServerSideEncryption` com o valor `AES256`. O exemplo de código PHP a seguir faz uma cópia de um objeto e adiciona a criptografia de servidor ao objeto copiado.

```
<?php

require 'vendor/autoload.php';

use Aws\S3\S3Client;

$sourceBucket = '*** Your Source Bucket Name ***';
$sourceKeyname = '*** Your Source Object Key ***';

$targetBucket = '*** Your Target Bucket Name ***';
$targetKeyname = '*** Your Target Object Key ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// Copy an object and add server-side encryption.
$s3->copyObject([
    'Bucket'           => $targetBucket,
    'Key'              => $targetKeyname,
    'CopySource'       => "{$sourceBucket}/{$sourceKeyname}",
    'ServerSideEncryption' => 'AES256',
]);
]
```

## Recursos relacionados

- [AWS SDK para PHP para classe `Aws\S3\S3Client` do Amazon S3](#)
- [Documentação do AWS SDK para PHP](#)

## Especificação da criptografia no lado do servidor usando o AWS SDK para Ruby

Ao usar o AWS SDK para Ruby para fazer upload de um objeto, você pode especificar que o objeto seja armazenado criptografado em repouso com a criptografia no lado do servidor (SSE). Ao ler o objeto de volta, ele é descriptografado automaticamente.

O exemplo de AWS SDK para Ruby – Versão 3 a seguir demonstra como especificar que um arquivo carregado no Amazon S3 seja criptografado em repouso.

```
require 'aws-sdk-s3'

s3 = Aws::S3::Resource.new(region:'us-west-2')
obj = s3.bucket('my-bucket').object('key')
obj.upload_file('local/path/to/file', :server_side_encryption => 'AES256')
```

Para obter um exemplo que mostra como fazer upload de um objeto sem SSE, consulte [Faça upload de objetos usando o AWS SDK para Ruby \(p. 179\)](#).

## Determinar o algoritmo de criptografia usado

O exemplo de código a seguir demonstra como determinar o estado de criptografia de um objeto existente.

```
# Determine server-side encryption of an object.
require 'aws-sdk-s3'

s3 = Aws::S3::Resource.new(region:'us-west-2')
```

```
enc = s3.bucket('bucket-name').object('key').server_side_encryption
enc_state = (enc != nil) ? enc : "not set"
puts "Encryption state is #{enc_state}."
```

Se a criptografia no lado do servidor não for usada para o objeto que é armazenado no Amazon S3, o método retornará nulo.

### Alteração da criptografia de servidor de um objeto existente (operação de cópia)

Para alterar o estado de criptografia de um objeto existente, faça uma cópia do objeto e exclua o objeto de origem. Por padrão, os métodos de cópia não criptografam o destino, a menos que você solicite explicitamente a criptografia no lado do servidor. Você pode solicitar a criptografia do objeto de destino especificando o valor `server_side_encryption` no argumento de hash de opções conforme exibido no seguinte exemplo de código Ruby. O exemplo de código demonstra como copiar um objeto e criptografar a cópia.

```
require 'aws-sdk-s3'

s3 = Aws::S3::Resource.new(region:'us-west-2')
bucket1 = s3.bucket('source-bucket-name')
bucket2 = s3.bucket('target-bucket-name')
obj1 = bucket1.object('key')
obj2 = bucket2.object('key')

obj1.copy_to(obj2, :server_side_encryption => 'AES256')
```

Para obter um exemplo de como copiar um objeto sem criptografia, consulte [Copie um objeto usando o AWS SDK para Ruby \(p. 223\)](#).

### Especificação da criptografia no lado do servidor usando a API REST

No momento da criação do objeto — ou seja, quando você faz upload de um objeto novo ou faz uma cópia de um objeto existente — você pode especificar se deseja que o Amazon S3 criptografe seus dados adicionando o cabeçalho `x-amz-server-side-encryption` à solicitação. Defina o valor do cabeçalho como o algoritmo de criptografia AES256 ao qual o Amazon S3 oferece suporte. O Amazon S3 confirma se o objeto está armazenado usando a criptografia no lado do servidor retornando o cabeçalho da resposta `x-amz-server-side-encryption`.

As seguintes APIs de upload REST aceitam o cabeçalho de solicitação `x-amz-server-side-encryption`.

- [Objeto PUT](#)
- [Objeto PUT - Copiar](#)
- [Objeto POST](#)
- [Iniciar multipart upload](#)

Ao fazer upload de objetos grandes usando a API de multipart upload, você pode especificar a criptografia no lado do servidor adicionando o cabeçalho `x-amz-server-side-encryption` à solicitação [Iniciar multipart upload](#). Ao copiar um objeto existente, independentemente de o objeto de origem ser criptografado ou não, o objeto de destino não é criptografado, a menos que você solicite explicitamente a criptografia de servidor.

Os cabeçalhos de resposta das seguintes APIs REST retornam o cabeçalho `x-amz-server-side-encryption` quando um objeto é armazenado usando criptografia de servidor.

- [Objeto PUT](#)
- [Objeto PUT - Copiar](#)
- [Objeto POST](#)

- Iniciar multipart upload
- Upload de parte
- Upload de parte - Copiar
- Concluir multipart upload
- Objeto Get
- Objeto Head

#### Note

Os cabeçalhos de solicitação de criptografia não deverão ser enviados para solicitações GET e solicitações HEAD se o objeto usar SSE-S3 ou for exibido um erro HTTP 400 BadRequest.

## Especificação da criptografia no lado do servidor usando o Console de gerenciamento da AWS

Ao fazer upload de um objeto usando o Console de gerenciamento da AWS, você pode especificar a criptografia no lado do servidor. Para ver um exemplo de como fazer upload de um objeto, consulte [Upload de objetos do S3](#).

Quando você copia um objeto usando o Console de gerenciamento da AWS, o console copia o objeto no estado em que ele se encontra. Ou seja, se a origem da cópia for criptografada, o objeto de destino será criptografado. O console também permite adicionar criptografia a um objeto. Para obter mais informações, consulte [Como faço para adicionar criptografia a um objeto do S3?](#).

## Proteção de dados usando criptografia de servidor com chaves de criptografia fornecidas pelo cliente (SSE-C)

A criptografia de servidor envolve a proteção de dados em repouso. Usar a criptografia de servidor com chaves de criptografia fornecidas pelo cliente (SSE-C) permite definir suas próprias chaves de criptografia. Com a chave de criptografia que você fornece como parte de sua solicitação, o Amazon S3 gerencia a criptografia, ao gravar em discos, e descriptografia, quando você acessa seus objetos. Portanto, você não precisa manter um código para executar a criptografia e a descriptografia de dados. A única coisa a fazer é gerenciar as chaves de criptografia que você fornece.

Quando você faz upload de um objeto, o Amazon S3 usa a chave de criptografia fornecida para aplicar a criptografia AES-256 aos seus dados e elimina a chave de criptografia da memória.

#### Important

O Amazon S3 não armazena a chave de criptografia que você fornece. Em vez disso, armazenamos um valor de HMAC com salt aleatório da chave de criptografia para validar solicitações futuras. O valor de HMAC com salt não pode ser usado para derivar o valor da chave de criptografia ou para decifrar o conteúdo do objeto criptografado. Isso significa que, se você perder a chave de criptografia, perderá o objeto.

Ao recuperar um objeto, você precisa fornecer a mesma chave de criptografia como parte de sua solicitação. O Amazon S3 primeiro confirma se há correspondência da chave de criptografia fornecida e, em seguida, descriptografa o objeto antes de devolver os dados dele para você.

Os destaques de SSE-C são:

- Você deve usar https.

#### Important

O Amazon S3 rejeitará todas as solicitações feitas por http ao usar o SSE-C. Por questões de segurança, recomendamos considerar que todas as chaves enviadas erroneamente por http estejam comprometidas. Você deve descartar a chave e alternar conforme apropriado.

- O ETag na resposta não é o MD5 dos dados de objeto.
- Você gerencia um mapeamento cuja chave de criptografia foi usada para criptografar objetos. O Amazon S3 não armazena chaves de criptografia. Você é responsável por acompanhar a chave de criptografia que forneceu para um objeto.
- Se seu bucket tiver versionamento ativado, cada versão de objeto carregada usando esse recurso poderá ter sua própria chave de criptografia. Você é responsável por acompanhar a chave de criptografia usada para uma versão de objeto.
- Como gerencia chaves de criptografia no lado do cliente, você gerencia todas as proteções adicionais, como a alternância de chave, no lado do cliente.

**Warning**

Se você perder a chave de criptografia, qualquer solicitação GET de um objeto sem chave de criptografia falhará e você perderá o objeto.

## Uso de SSE-C

Ao usar a criptografia de servidor com chaves de criptografia fornecidas pelo cliente (SSE-C), você deve fornecer informações de chave de criptografia usando os seguintes cabeçalhos de solicitação.

Nome	Descrição
<code>x-amz-server-side-encryption-customer-algorithm</code>	Use esse cabeçalho para especificar o algoritmo de criptografia. O valor do cabeçalho deve ser "AES256".
<code>x-amz-server-side-encryption-customer-key</code>	Use esse cabeçalho para fornecer a chave de criptografia com codificação base64 de 256 bits para o Amazon S3 a ser usada para criptografar ou descriptografar seus dados.
<code>x-amz-server-side-encryption-customer-key-MD5</code>	Use esse cabeçalho para fornecer o resumo MD5 de 128 bits codificado em base64 da chave de criptografia de acordo com o <a href="#">RFC 1321</a> . O Amazon S3 usa esse cabeçalho para uma verificação de integridade de mensagem a fim de garantir que a chave de criptografia tenha sido transmitida sem erro.

Você pode usar bibliotecas de wrapper do AWS SDK para adicionar esses cabeçalhos à sua solicitação. Se precisar, você pode fazer com que a API REST do Amazon S3 seja chamada diretamente em seu aplicativo.

**Note**

Você não pode usar o console do Amazon S3 para fazer upload de um objeto e solicitar o SSE-C. Também não é possível usar o console para atualizar (por exemplo, alterar a classe de armazenamento ou adicionar metadados) um objeto armazenado com o SSE-C.

As APIs do Amazon S3 a seguir são compatíveis com esses cabeçalhos.

- Operação GET — Ao recuperar objetos usando a API GET (consulte [Objeto GET](#)), você pode especificar os cabeçalhos de solicitação. O Torrent não é compatível com objetos criptografados usando SSE-C.
- Operação HEAD — Para recuperar metadados de objeto usando a API HEAD (consulte [Objeto HEAD](#)), você pode especificar esses cabeçalhos de solicitação.
- Operação PUT — Ao fazer upload de dados usando a API PUT (consulte [Objeto PUT](#)), você pode especificar esses cabeçalhos de solicitação.

- Multipart upload — Ao fazer upload de objetos grandes usando a API de multipart upload, você pode especificar esses cabeçalhos. Especifique esses cabeçalhos na solicitação iniciada (consulte [Iniciar o multipart upload](#)) e em cada solicitação de upload de parte subsequente ([Fazer upload de parte](#)). Para cada solicitação de upload de parte, as informações de criptografia devem ser as mesmas que você forneceu na solicitação iniciada do multipart upload.
- Operação POST — Ao usar uma operação POST para fazer upload de um objeto (consulte [Objeto POST](#)), em vez dos cabeçalhos de solicitação, você fornece as mesmas informações nos campos de formulário.
- Operação de cópia — Ao copiar um objeto (consulte [Objeto PUT - Copiar](#)), você tem um objeto de origem e um objeto de destino. Assim, é necessário considerar o seguinte:
  - Se você quiser que o objeto de destino seja criptografado usando criptografia de servidor com chaves gerenciadas pela AWS, forneça o cabeçalho de solicitação `x-amz-server-side-encryption`.
  - Se você quiser que o objeto de destino seja criptografado usando SSE-C, forneça informações de criptografia usando os três cabeçalhos descritos na tabela anterior.
  - Se o objeto de origem for criptografado usando SSE-C, você deverá fornecer informações de chave de criptografia usando os seguintes cabeçalhos para que o Amazon S3 possa descriptografar o objeto para cópia.

Nome	Descrição
<code>x-amz-copy-source-server-side-encryption-customer-algorithm</code>	Inclua esse cabeçalho para especificar o algoritmo que o Amazon S3 deve usar para decifrar o objeto de origem. Esse valor deve ser <code>AES256</code> .
<code>x-amz-copy-source-server-side-encryption-customer-key</code>	Inclua esse cabeçalho para fornecer a chave de criptografia com codificação base64 para o Amazon S3 a ser usada para descriptografar o objeto de origem. Essa chave de criptografia deve ser a que você forneceu ao Amazon S3 quando criou o objeto de origem. Caso contrário, o Amazon S3 não poderá descriptografar o objeto.
<code>x-amz-copy-source-server-side-encryption-customer-key-MD5</code>	Inclua esse cabeçalho para fornecer o resumo MD5 com codificação base64 de 128 bits da chave de criptografia, de acordo com <a href="#">RFC 1321</a> .

## Pre-signed URL e SSE-C

Você pode gerar um pre-signed URL que pode ser usado para operações, como fazer upload de um objeto novo, recuperar um objeto existente ou metadados de objeto. Os pre-signed URLs oferecem suporte para SSE-C da seguinte maneira:

- Ao criar um pre-signed URL, você deve especificar o algoritmo, usando `x-amz-server-side-encryption-customer-algorithm` no cálculo de assinatura.
- Ao usar o pre-signed URL para fazer upload de um objeto novo, recuperar um objeto existente ou recuperar somente metadados de objeto, você deve fornecer todos os cabeçalhos de criptografia em seu aplicativo cliente.

### Note

Para objetos não SSE-C, é possível gerar um pre-signed URL e colá-lo diretamente em um navegador, por exemplo, para acessar os dados.

No entanto, isso não é válido para objetos SSE-C porque, além do pre-signed URL, você também precisa incluir cabeçalhos HTTP específicos para objetos SSE-C. Dessa forma, é possível usar o pre-signed URL para objetos SSE-C somente de maneira programática.

Para obter mais informações, consulte os tópicos a seguir:

- [Especificar criptografia de servidor com chaves de criptografia fornecidas pelo cliente usando o AWS SDK for Java \(p. 428\)](#)
- [Especificar criptografia de servidor com chaves de criptografia fornecidas pelo cliente usando o AWS SDK para .NET \(p. 432\)](#)
- [Especificação de criptografia de servidor com chaves de criptografia fornecidas pelo cliente usando a API REST \(p. 440\)](#)

## Especificar criptografia de servidor com chaves de criptografia fornecidas pelo cliente usando o AWS SDK for Java

O exemplo a seguir mostra como solicitar a criptografia de servidor com chaves fornecidas pelo cliente (SSE-C) para objetos. O exemplo executa as seguintes operações. Cada operação mostra como especificar cabeçalhos relacionados a SSE-C na solicitação:

- Objeto PUT—faz upload de um objeto e solicita a criptografia de servidor, usando uma chave de criptografia fornecida pelo cliente.
- Objeto Get—faz download do objeto carregado na etapa anterior. Na solicitação, você fornece as mesmas informações de criptografia fornecidas no upload do objeto. O Amazon S3 precisa dessas informações para descriptografar o objeto e devolvê-lo para você.
- Obter metadados do objeto—recupera os metadados do objeto. Você fornece as mesmas informações de criptografia usadas quando o objeto foi criado.
- Copiar objeto—faz uma cópia de um objeto carregado anteriormente. Como o objeto de origem é armazenado usando SSE-C, você deve fornecer suas informações de criptografia na solicitação de cópia. Por padrão, o Amazon S3 criptografa a cópia do objeto somente se você solicitar explicitamente. Esse exemplo instrui o Amazon S3 a armazenar uma cópia criptografada do objeto usando uma nova chave SSE-C.

### Note

Este exemplo mostra como fazer upload de um objeto em uma única operação. Ao usar a API de Multipart Upload para fazer upload de objetos grandes, você fornece informações de criptografia da mesma maneira que exibidas nesse exemplo. Para exemplos de multipart uploads usando o AWS SDK for Java, consulte [Usar o AWS Java SDK para multipart upload \(API de alto nível\) \(p. 188\)](#) e [Usar o AWS Java SDK para um multipart upload \(API de baixo nível\) \(p. 192\)](#).

Para adicionar informações necessárias de criptografia, inclua uma `sseCustomerKey` na solicitação. Para obter mais informações sobre a classe `sseCustomerKey`, consulte [Uso de SSE-C \(p. 426\)](#).

Para obter informações sobre SSE-C, consulte [Proteção de dados usando criptografia de servidor com chaves de criptografia fornecidas pelo cliente \(SSE-C\) \(p. 425\)](#). Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

### Example

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.  
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)  
  
import java.io.BufferedReader;  
import java.io.File;  
import java.io.IOException;  
import java.io.InputStreamReader;
```

```
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;

import javax.crypto.KeyGenerator;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.CopyObjectRequest;
import com.amazonaws.services.s3.model.GetObjectMetadataRequest;
import com.amazonaws.services.s3.model.GetObjectRequest;
import com.amazonaws.services.s3.model.ObjectMetadata;
import com.amazonaws.services.s3.model.PutObjectRequest;
import com.amazonaws.services.s3.model.S3Object;
import com.amazonaws.services.s3.model.S3ObjectInputStream;
import com.amazonaws.services.s3.model.SSECustomerKey;

public class ServerSideEncryptionUsingClientSideEncryptionKey {
    private static SSECustomerKey SSE_KEY;
    private static AmazonS3 S3_CLIENT;
    private static KeyGenerator KEY_GENERATOR;

    public static void main(String[] args) throws IOException, NoSuchAlgorithmException {
        String clientRegion = "*** Client region ***";
        String bucketName = "*** Bucket name ***";
        String keyName = "*** Key name ***";
        String uploadFileName = "*** File path ***";
        String targetKeyName = "*** Target key name ***";

        // Create an encryption key.
        KEY_GENERATOR = KeyGenerator.getInstance("AES");
        KEY_GENERATOR.init(256, new SecureRandom());
        SSE_KEY = new SSECustomerKey(KEY_GENERATOR.generateKey());

        try {
            S3_CLIENT = AmazonS3ClientBuilder.standard()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(clientRegion)
                .build();

            // Upload an object.
            uploadObject(bucketName, keyName, new File(uploadFileName));

            // Download the object.
            downloadObject(bucketName, keyName);

            // Verify that the object is properly encrypted by attempting to retrieve it
            // using the encryption key.
            retrieveObjectMetadata(bucketName, keyName);

            // Copy the object into a new object that also uses SSE-C.
            copyObject(bucketName, keyName, targetKeyName);
        }
        catch(AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        }
        catch(SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

```
private static void uploadObject(String bucketName, String keyName, File file) {
    PutObjectRequest putRequest = new PutObjectRequest(bucketName, keyName,
file).withSSECustomerKey(SSE_KEY);
    S3_CLIENT.putObject(putRequest);
    System.out.println("Object uploaded");
}

private static void downloadObject(String bucketName, String keyName) throws
IOException {
    GetObjectRequest getObjectRequest = new GetObjectRequest(bucketName,
keyName).withSSECustomerKey(SSE_KEY);
    S3Object object = S3_CLIENT.getObject(getObjectRequest);

    System.out.println("Object content: ");
    displayTextInputStream(object.getObjectContent());
}

private static void retrieveObjectMetadata(String bucketName, String keyName) {
    GetObjectMetadataRequest getMetadataRequest = new
GetObjectMetadataRequest(bucketName, keyName)

    .withSSECustomerKey(SSE_KEY);
    ObjectMetadata objectMetadata = S3_CLIENT.getObjectMetadata(getMetadataRequest);
    System.out.println("Metadata retrieved. Object size: " +
objectMetadata.getContentLength());
}

private static void copyObject(String bucketName, String keyName, String targetKeyName)
    throws NoSuchAlgorithmException {
    // Create a new encryption key for target so that the target is saved using SSE-C.
    SSECustomerKey newSSEKey = new SSECustomerKey(KEY_GENERATOR.generateKey());

    CopyObjectRequest copyRequest = new CopyObjectRequest(bucketName, keyName,
bucketName, targetKeyName)
        .withSourceSSECustomerKey(SSE_KEY)
        .withDestinationSSECustomerKey(newSSEKey);

    S3_CLIENT.copyObject(copyRequest);
    System.out.println("Object copied");
}

private static void displayTextInputStream(S3ObjectInputStream input) throws
IOException {
    // Read one line at a time from the input stream and display each line.
    BufferedReader reader = new BufferedReader(new InputStreamReader(input));
    String line;
    while ((line = reader.readLine()) != null) {
        System.out.println(line);
    }
    System.out.println();
}
}
```

## Outras Operações do Amazon S3 com SSE-C usando o AWS SDK for Java

O exemplo da seção anterior mostra como solicitar a criptografia de servidor com chaves fornecidas pelo cliente (SSE-C) nas operações PUT, GET, Head e de cópia. Esta seção descreve outras APIs que oferecem suporte para SSE-C.

Para fazer upload de objetos grandes, você pode usar a API de multipart upload (consulte [Upload de objetos usando a API de multipart upload \(p. 181\)](#)). Você pode usar APIs de alto ou baixo nível para fazer upload de objetos grandes. Essas APIs oferecem suporte para cabeçalhos relacionados à criptografia na solicitação.

- Ao usar a API do `TransferManager` de alto nível, você fornece os cabeçalhos específicos de criptografia na `PutObjectRequest` (consulte [Usar o AWS Java SDK para multipart upload \(API de alto nível\) \(p. 188\)](#)).
- Ao usar a API de baixo nível, você fornece informações relacionadas à criptografia na `InitiateMultipartUploadRequest`, seguidas por informações de criptografia idênticas em cada `UploadPartRequest`. Você não precisa fornecer cabeçalhos específicos de criptografia na `CompleteMultipartUploadRequest`. Para ver exemplos, consulte [Usar o AWS Java SDK para um multipart upload \(API de baixo nível\) \(p. 192\)](#).

O exemplo a seguir usa `TransferManager` para criar objetos e mostra como fornecer informações relacionadas a SSE-C. O exemplo faz o seguinte:

- Cria um objeto usando o método `TransferManager.upload()`. Na instância `PutObjectRequest`, você fornece informações de chave de criptografia para solicitar ao Amazon S3 para armazenar o objeto criptografado usando a chave de criptografia fornecida pelo cliente.
- Faz uma cópia do objeto, chamando o método `TransferManager.copy()`. O exemplo instrui o Amazon S3 a criptografar a cópia do objeto usando um novo `SSECustomerKey`. Como o objeto de origem é criptografado usando SSE-C, o `CopyObjectRequest` também fornece a chave de criptografia do objeto de origem para que o Amazon S3 possa descriptografar o objeto antes de copiá-lo.

#### Example

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

import java.io.File;
import java.security.SecureRandom;

import javax.crypto.KeyGenerator;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.CopyObjectRequest;
import com.amazonaws.services.s3.model.PutObjectRequest;
import com.amazonaws.services.s3.model.SSECustomerKey;
import com.amazonaws.services.s3.transfer.Copy;
import com.amazonaws.services.s3.transfer.TransferManager;
import com.amazonaws.services.s3.transfer.TransferManagerBuilder;
import com.amazonaws.services.s3.transfer.Upload;

public class ServerSideEncryptionCopyObjectUsingHLwithSSEC {

    public static void main(String[] args) throws Exception {
        String clientRegion = "**** Client region ****";
        String bucketName = "**** Bucket name ****";
        String fileToUpload = "**** File path ****";
        String keyName = "**** New object key name ****";
        String targetKeyName = "**** Key name for object copy ****";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .withRegion(clientRegion)
                .withCredentials(new ProfileCredentialsProvider())
                .build();
            TransferManager tm = TransferManagerBuilder.standard()
                .withS3Client(s3Client)
                .withTransferConfig(new TransferConfig()
                    .withMultipartThreshold(1024 * 1024 * 5))
                .build();

            PutObjectRequest putObjectRequest = new PutObjectRequest(bucketName, fileToUpload, new File(fileToUpload))
                .withSSECustomerKey(keyName);
            tm.upload(putObjectRequest);

            CopyObjectRequest copyObjectRequest = new CopyObjectRequest(bucketName, fileToUpload, bucketName, fileToUpload)
                .withSSECustomerKey(targetKeyName);
            tm.copy(copyObjectRequest);
        }
    }
}
```

```
        .build();

        // Create an object from a file.
        PutObjectRequest putObjectRequest = new PutObjectRequest(bucketName, keyName,
new File(fileToUpload));

        // Create an encryption key.
        KeyGenerator keyGenerator = KeyGenerator.getInstance("AES");
        keyGenerator.init(256, new SecureRandom());
        SSECustomerKey sseCustomerEncryptionKey = new
SSECustomerKey(keyGenerator.generateKey());

        // Upload the object. TransferManager uploads asynchronously, so this call
returns immediately.
        putObjectRequest.setSSECustomerKey(sseCustomerEncryptionKey);
        Upload upload = tm.upload(putObjectRequest);

        // Optionally, wait for the upload to finish before continuing.
        upload.waitForCompletion();
        System.out.println("Object created.");

        // Copy the object and store the copy using SSE-C with a new key.
        CopyObjectRequest copyObjectRequest = new CopyObjectRequest(bucketName,
keyName, bucketName, targetKeyName);
        SSECustomerKey sseTargetObjectEncryptionKey = new
SSECustomerKey(keyGenerator.generateKey());
        copyObjectRequest.setSourceSSECustomerKey(sseCustomerEncryptionKey);
        copyObjectRequest.setDestinationSSECustomerKey(sseTargetObjectEncryptionKey);

        // Copy the object. TransferManager copies asynchronously, so this call returns
immediately.
        Copy copy = tm.copy(copyObjectRequest);

        // Optionally, wait for the upload to finish before continuing.
        copy.waitForCompletion();
        System.out.println("Copy complete.");
    }
    catch(AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    }
    catch(SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}
```

## Especificar criptografia de servidor com chaves de criptografia fornecidas pelo cliente usando o AWS SDK para .NET

O C# de exemplo a seguir mostra como funciona a criptografia de servidor com chaves fornecidas pelo cliente (SSE-C). O exemplo executa as seguintes operações. Cada operação mostra como especificar cabeçalhos relacionados a SSE-C na solicitação.

- Objeto PUT—faz upload de um objeto e solicita a criptografia de servidor, usando chaves de criptografia fornecidas pelo cliente.
- Objeto GET—faz download do objeto que foi carregado na etapa anterior. A solicitação fornece as mesmas informações de criptografia fornecidas no upload do objeto. O Amazon S3 precisa dessas informações para descriptografar o objeto e devolvê-lo para você.

- Metadados do objeto GET—fornecerá as mesmas informações de criptografia usadas ao criar o objeto para recuperar os metadados do objeto.
- Copiar objeto—faz uma cópia de um objeto carregado. Como o objeto de origem é armazenado usando SSE-C, a solicitação de cópia deve fornecer as informações de criptografia. Por padrão, o Amazon S3 não criptografa a cópia de um objeto. O código instrui o Amazon S3 a criptografar o objeto copiado que usa SSE-C fornecendo informações relacionadas à criptografia para o destino. Ele também armazena o destino.

#### Note

Para exemplos de upload de objetos grandes usando a API multipart upload, consulte [Usar o AWS SDK para .NET para multipart upload \(API de alto nível\) \(p. 197\)](#) e [Usar o AWS SDK para .NET para multipart upload \(API de nível baixo\) \(p. 204\)](#).

Para obter informações sobre SSE-C, consulte [Proteção de dados usando criptografia de servidor com chaves de criptografia fornecidas pelo cliente \(SSE-C\) \(p. 425\)](#). Para obter informações sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

#### Example

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.IO;
using System.Security.Cryptography;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class SSEClientEncryptionKeyObjectOperationsTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string keyName = "*** key name for new object created ***";
        private const string copyTargetKeyName = "*** key name for object copy ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            ObjectOpsUsingClientEncryptionKeyAsync().Wait();
        }

        private static async Task ObjectOpsUsingClientEncryptionKeyAsync()
        {
            try
            {
                // Create an encryption key.
                Aes aesEncryption = Aes.Create();
                aesEncryption.KeySize = 256;
                aesEncryption.GenerateKey();
                string base64Key = Convert.ToBase64String(aesEncryption.Key);

                // 1. Upload the object.
                PutObjectRequest putObjectRequest = await UploadObjectAsync(base64Key);
                // 2. Download the object and verify that its contents matches what you
                uploaded.
            }
        }
    }
}
```

```
        await DownloadObjectAsync(base64Key, putObjectRequest);
        // 3. Get object metadata and verify that the object uses AES-256
encryption.
        await GetObjectMetadataAsync(base64Key);
        // 4. Copy both the source and target objects using server-side encryption
with
        //    a customer-provided encryption key.
        await CopyObjectAsync(aesEncryption, base64Key);
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine("Error encountered ***. Message:'{0}' when writing an
object", e.Message);
    }
    catch (Exception e)
    {
        Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
    }
}

private static async Task<PutObjectRequest> UploadObjectAsync(string base64Key)
{
    PutObjectRequest putObjectRequest = new PutObjectRequest
    {
        BucketName = bucketName,
        Key = keyName,
        ContentBody = "sample text",
        ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
        ServerSideEncryptionCustomerProvidedKey = base64Key
    };
    PutObjectResponse putObjectResponse = await
client.PutObjectAsync(putObjectRequest);
    return putObjectRequest;
}
private static async Task DownloadObjectAsync(string base64Key, PutObjectRequest
putObjectRequest)
{
    GetObjectRequest getObjectRequest = new GetObjectRequest
    {
        BucketName = bucketName,
        Key = keyName,
        // Provide encryption information for the object stored in Amazon S3.
        ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
        ServerSideEncryptionCustomerProvidedKey = base64Key
    };

    using (GetObjectResponse getResponse = await
client.GetObjectAsync(getObjectRequest))
    using (StreamReader reader = new StreamReader(getResponse.ResponseStream))
    {
        string content = reader.ReadToEnd();
        if (String.Compare(putObjectRequest.ContentBody, content) == 0)
            Console.WriteLine("Object content is same as we uploaded");
        else
            Console.WriteLine("Error...Object content is not same.");

        if (getResponse.ServerSideEncryptionCustomerMethod ==
ServerSideEncryptionCustomerMethod.AES256)
            Console.WriteLine("Object encryption method is AES256, same as we
set");
        else
            Console.WriteLine("Error...Object encryption method is not the same as
AES256 we set");
    }
}
```

```
        // Assert.AreEqual(putObjectRequest.ContentBody, content);
        // Assert.AreEqual(ServerSideEncryptionCustomerMethod.AES256,
getResponse.ServerSideEncryptionCustomerMethod);
    }
}
private static async Task GetObjectMetadataAsync(string base64Key)
{
    GetObjectMetadataRequest getObjectMetadataRequest = new
GetObjectMetadataRequest
{
    BucketName = bucketName,
    Key = keyName,

    // The object stored in Amazon S3 is encrypted, so provide the necessary
encryption information.
    ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
    ServerSideEncryptionCustomerProvidedKey = base64Key
};

    GetObjectMetadataResponse getObjectMetadataResponse = await
client.GetObjectMetadataAsync(getObjectMetadataRequest);
    Console.WriteLine("The object metadata show encryption method used is: {0}",
getObjectMetadataResponse.ServerSideEncryptionCustomerMethod);
    // Assert.AreEqual(ServerSideEncryptionCustomerMethod.AES256,
getObjectMetadataResponse.ServerSideEncryptionCustomerMethod);
}
private static async Task CopyObjectAsync(Aes aesEncryption, string base64Key)
{
    aesEncryption.GenerateKey();
    string copyBase64Key = Convert.ToBase64String(aesEncryption.Key);

    CopyObjectRequest copyRequest = new CopyObjectRequest
{
    SourceBucket = bucketName,
    SourceKey = keyName,
    DestinationBucket = bucketName,
    DestinationKey = copyTargetKeyName,
    // Information about the source object's encryption.
    CopySourceServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
    CopySourceServerSideEncryptionCustomerProvidedKey = base64Key,
    // Information about the target object's encryption.
    ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
    ServerSideEncryptionCustomerProvidedKey = copyBase64Key
};
    await client.CopyObjectAsync(copyRequest);
}
}
```

## Outras operações do Amazon S3 e SSE-C

O exemplo da seção anterior mostra como solicitar a criptografia de servidor com a chave fornecida pelo cliente (SSE-C) nas operações PUT, GET, Head e Copy. Esta seção descreve outras APIs do Amazon S3 que oferecem suporte para SSE-C.

Para fazer upload de objetos grandes, você pode usar a API de multipart upload (consulte [Upload de objetos usando a API de multipart upload \(p. 181\)](#)). O AWS SDK para .NET fornece APIs de alto ou baixo nível para fazer upload de grandes objetos. Essas APIs oferecem suporte para cabeçalhos relacionados à criptografia na solicitação.

- Ao usar a API do Transfer-Utility de alto nível, você fornece os cabeçalhos específicos de criptografia na `TransferUtilityUploadRequest`, conforme mostrado. Para obter exemplos de código, consulte [Usar o AWS SDK para .NET para multipart upload \(API de alto nível\) \(p. 197\)](#).

```
TransferUtilityUploadRequest request = new TransferUtilityUploadRequest()
{
    FilePath = filePath,
    BucketName = existingBucketName,
    Key = keyName,
    // Provide encryption information.
    ServerSideEncryptionCustomerMethod = ServerSideEncryptionCustomerMethod.AES256,
    ServerSideEncryptionCustomerProvidedKey = base64Key,
};
```

- Ao usar a API de baixo nível, você fornece informações relacionadas à criptografia na solicitação para iniciar o multipart upload, seguidas por informações de criptografia idênticas nas solicitações subsequentes de upload de parte. Você não precisa fornecer cabeçalhos específicos de criptografia na solicitação de multipart upload completo. Para ver exemplos, consulte [Usar o AWS SDK para .NET para multipart upload \(API de nível baixo\) \(p. 204\)](#).

O seguinte é um exemplo de multipart upload de baixo nível que faz uma cópia de um objeto grande existente. No exemplo, o objeto a ser copiado é armazenado no Amazon S3 usando o SSE-C, e você deseja salvar o objeto de destino também usando o SSE-C. No exemplo, faça o seguinte:

- Inicie uma solicitação de multipart upload fornecendo uma chave de criptografia e as informações relacionadas.
- Forneça as chaves de criptografia de objeto de origem e de destino e as informações relacionadas na `CopyPartRequest`.
- Obtenha o tamanho do objeto de origem a ser copiado recuperando os metadados do objeto.
- Faça upload dos objetos em partes de 5 MB.

### Example

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.IO;
using System.Security.Cryptography;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class SSECLowLevelMPUcopyObjectTest
    {
        private const string existingBucketName = "*** bucket name ***";
        private const string sourceKeyName      = "*** source object key name ***";
        private const string targetKeyName     = "*** key name for the target object ***";
        private const string filePath          = @ "*** file path ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
        static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            CopyObjClientEncryptionKeyAsync().Wait();
        }
    }
}
```

```
private static async Task CopyObjClientEncryptionKeyAsync()
{
    Aes aesEncryption = Aes.Create();
    aesEncryption.KeySize = 256;
    aesEncryption.GenerateKey();
    string base64Key = Convert.ToBase64String(aesEncryption.Key);

    await CreateSampleObjUsingClientEncryptionKeyAsync(base64Key, s3Client);

    await CopyObjectAsync(s3Client, base64Key);
}

private static async Task CopyObjectAsync(IAmazonS3 s3Client, string base64Key)
{
    List<CopyPartResponse> uploadResponses = new List<CopyPartResponse>();

    // 1. Initialize.
    InitiateMultipartUploadRequest initiateRequest = new
InitiateMultipartUploadRequest
    {
        BucketName = existingBucketName,
        Key = targetKeyName,
        ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
        ServerSideEncryptionCustomerProvidedKey = base64Key,
    };

    InitiateMultipartUploadResponse initResponse =
        await s3Client.InitiateMultipartUploadAsync(initiateRequest);

    // 2. Upload Parts.
    long partSize = 5 * (long)Math.Pow(2, 20); // 5 MB
    long firstByte = 0;
    long lastByte = partSize;

    try
    {
        // First find source object size. Because object is stored encrypted with
        // customer provided key you need to provide encryption information in
your request.
        GetObjectMetadataRequest getObjectMetadataRequest = new
GetObjectMetadataRequest()
        {
            BucketName = existingBucketName,
            Key = sourceKeyName,
            ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
            ServerSideEncryptionCustomerProvidedKey = base64Key // " * **source
object encryption key ***"
        };

        GetObjectMetadataResponse getObjectMetadataResponse = await
s3Client.GetObjectMetadataAsync(getObjectMetadataRequest);

        long filePosition = 0;
        for (int i = 1; filePosition < getObjectMetadataResponse.ContentLength; i
++)
    {
        CopyPartRequest copyPartRequest = new CopyPartRequest
        {
            UploadId = initResponse.UploadId,
            // Source.
            SourceBucket = existingBucketName,
            SourceKey = sourceKeyName,
            // Source object is stored using SSE-C. Provide encryption
information.
    }
}
}
```

```
        CopySourceServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
        CopySourceServerSideEncryptionCustomerProvidedKey =
base64Key, //***source object encryption key ***",
        FirstByte = firstByte,
        // If the last part is smaller then our normal part size then use
the remaining size.
        LastByte = lastByte > getObjectMetadataResponse.ContentLength ?
            getObjectMetadataResponse.ContentLength - 1 : lastByte,

        // Target.
        DestinationBucket = existingBucketName,
        DestinationKey = targetKeyName,
        PartNumber = i,
        // Encryption information for the target object.
        ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
        ServerSideEncryptionCustomerProvidedKey = base64Key
    };
    uploadResponses.Add(await s3Client.CopyPartAsync(copyPartRequest));
    filePosition += partSize;
    firstByte += partSize;
    lastByte += partSize;
}

// Step 3: complete.
CompleteMultipartUploadRequest completeRequest = new
CompleteMultipartUploadRequest
{
    BucketName = existingBucketName,
    Key = targetKeyName,
    UploadId = initResponse.UploadId,
};
completeRequest.AddPartETags(uploadResponses);

CompleteMultipartUploadResponse completeUploadResponse =
    await s3Client.CompleteMultipartUploadAsync(completeRequest);
}
catch (Exception exception)
{
    Console.WriteLine("Exception occurred: {0}", exception.Message);
    AbortMultipartUploadRequest abortMPURequest = new
AbortMultipartUploadRequest
{
    BucketName = existingBucketName,
    Key = targetKeyName,
    UploadId = initResponse.UploadId
};
s3Client.AbortMultipartUpload(abortMPURequest);
}
}
private static async Task CreateSampleObjUsingClientEncryptionKeyAsync(string
base64Key, IAmazonS3 s3Client)
{
    // List to store upload part responses.
    List<UploadPartResponse> uploadResponses = new List<UploadPartResponse>();

    // 1. Initialize.
    InitiateMultipartUploadRequest initiateRequest = new
InitiateMultipartUploadRequest
{
    BucketName = existingBucketName,
    Key = sourceKeyName,
    ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
    ServerSideEncryptionCustomerProvidedKey = base64Key
```

```
};

InitiateMultipartUploadResponse initResponse =
    await s3Client.InitiateMultipartUploadAsync(initiateRequest);

// 2. Upload Parts.
long contentLength = new FileInfo(filePath).Length;
long partSize = 5 * (long)Math.Pow(2, 20); // 5 MB

try
{
    long filePosition = 0;
    for (int i = 1; filePosition < contentLength; i++)
    {
        UploadPartRequest uploadRequest = new UploadPartRequest
        {
            BucketName = existingBucketName,
            Key = sourceKeyName,
            UploadId = initResponse.UploadId,
            PartNumber = i,
            PartSize = partSize,
            FilePosition = filePosition,
            FilePath = filePath,
            ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
            ServerSideEncryptionCustomerProvidedKey = base64Key
        };

        // Upload part and add response to our list.
        uploadResponses.Add(await s3Client.UploadPartAsync(uploadRequest));

        filePosition += partSize;
    }

    // Step 3: complete.
    CompleteMultipartUploadRequest completeRequest = new
CompleteMultipartUploadRequest
{
    BucketName = existingBucketName,
    Key = sourceKeyName,
    UploadId = initResponse.UploadId,
    //PartETags = new List<PartETag>(uploadResponses)

};

completeRequest.AddPartETags(uploadResponses);

CompleteMultipartUploadResponse completeUploadResponse =
    await s3Client.CompleteMultipartUploadAsync(completeRequest);

}

catch (Exception exception)
{
    Console.WriteLine("Exception occurred: {0}", exception.Message);
    AbortMultipartUploadRequest abortMPURequest = new
AbortMultipartUploadRequest
{
    BucketName = existingBucketName,
    Key = sourceKeyName,
    UploadId = initResponse.UploadId
};
    await s3Client.AbortMultipartUploadAsync(abortMPURequest);
}
}
```

## Especificação de criptografia de servidor com chaves de criptografia fornecidas pelo cliente usando a API REST

As APIs REST do Amazon S3 a seguir oferecem suporte aos cabeçalhos, com relação à criptografia de servidor com chaves de criptografia fornecidas pelo cliente. Para obter mais informações sobre esses cabeçalhos, consulte [Uso de SSE-C \(p. 426\)](#).

- [Objeto GET](#)
- [Objeto HEAD](#)
- [Objeto PUT](#)
- [Objeto PUT - Copiar](#)
- [Objeto POST](#)
- [Iniciar multipart upload](#)
- [Upload de parte](#)
- [Upload de parte - Copiar](#)

## Proteger dados usando criptografia no lado do cliente

Criptografia no lado do cliente é o ato de criptografar os dados antes de enviá-los para o Amazon S3. Para habilitar a criptografia no lado do cliente, você tem as seguintes opções:

- Usar uma chave mestra do AWS KMS gerenciada pelo cliente
- Usar uma chave mestra no lado do cliente

Os AWS SDKs a seguir dão suporte à criptografia no lado do cliente:

- [AWS SDK para .NET](#)
- [AWS SDK para Go](#)
- [AWS SDK for Java](#)
- [AWS SDK para PHP](#)
- [AWS SDK para Ruby](#)

## Opção 1: Usar uma chave mestra do KMS da AWS gerenciada pelo cliente (CMK)

Ao usar uma chave mestra do AWS KMS gerenciada pelo cliente para habilitar a criptografia de dados no lado do cliente, você fornece um ID de chave mestra de cliente da KMS da AWS (ID da CMK).

- Ao fazer upload de um objeto—Usando o ID do CMK, o cliente primeiro envia uma solicitação para o AWS Key Management Service (AWS KMS) de uma chave que possa ser usada para criptografar os dados do objeto. O AWS KMS retorna duas versões de uma chave de criptografia de dados gerada de maneira aleatória:
  - Uma versão em texto simples que o cliente usa para criptografar os dados de objeto
  - Um blob de criptografia da mesma chave de criptografia de dados que o cliente faz upload para o Amazon S3 como metadados de objeto

### Note

O cliente obtém uma chave de criptografia de dados exclusiva para cada objeto cujo upload é feito.

- Para fazer download de um objeto—O cliente faz download do objeto criptografado do Amazon S3 junto com a versão do blob de criptografia da chave de criptografia de dados armazenada como metadados de objeto. Em seguida, o cliente envia o blob de criptografia para o AWS KMS para obter a versão em texto simples da chave para poder descriptografar os dados de objeto.

Para obter mais informações sobre o AWS KMS, consulte [O que é o AWS Key Management Service?](#) no AWS Key Management Service Developer Guide.

#### Example

O exemplo a seguir faz upload de um objeto para o Amazon S3 usando o AWS KMS com o AWS SDK for Java. O exemplo usa uma chave mestra do cliente (CMK) gerenciada pelo KMS para criptografar dados no lado do cliente antes de fazer upload para o Amazon S3. Se já tiver um CMK, você poderá usá-lo especificando o valor da variável `kms_cmk_id` no código de exemplo. Se não tiver um CMK, ou precisar de outro, você poderá gerar um por meio da API Java. O exemplo mostra como gerar um CMK.

Para obter mais informações sobre o material de chaves, consulte [Importar material de chaves no AWS Key Management Service \(AWS KMS\)](#). Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

import java.io.ByteArrayOutputStream;
import java.io.IOException;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.RegionUtils;
import com.amazonaws.services.kms.AWSKMS;
import com.amazonaws.services.kms.AWSKMSClientBuilder;
import com.amazonaws.services.kms.model.CreateKeyResult;
import com.amazonaws.services.s3.AmazonS3Encryption;
import com.amazonaws.services.s3.AmazonS3EncryptionClientBuilder;
import com.amazonaws.services.s3.model.CryptoConfiguration;
import com.amazonaws.services.s3.model.KMSEncryptionMaterialsProvider;
import com.amazonaws.services.s3.model.S3Object;
import com.amazonaws.services.s3.model.S3ObjectInputStream;

public class UploadObjectKMSKey {

    public static void main(String[] args) throws IOException {
        String bucketName = "*** Bucket name ***";
        String keyName = "*** Object key name ***";
        String clientRegion = "*** Client region ***";
        String kms_cmk_id = "***AWS KMS customer master key ID***";
        int readChunkSize = 4096;

        try {
            // Optional: If you don't have a KMS key (or need another one),
            // create one. This example creates a key with AWS-created
            // key material.
            AWSKMS kmsClient = AWSKMSClientBuilder.standard()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(clientRegion)
                .build();
            CreateKeyResult keyResult = kmsClient.createKey();
            kms_cmk_id = keyResult.getKeyMetadata().getKeyId();

            // Create the encryption client.
```

```
KMSEncryptionMaterialsProvider materialProvider = new
KMSEncryptionMaterialsProvider(kms_cmk_id);
CryptoConfiguration cryptoConfig = new CryptoConfiguration()
    .withAwsKmsRegion(RegionUtils.getRegion(clientRegion));
AmazonS3Encryption encryptionClient =
AmazonS3EncryptionClientBuilder.standard()
    .withCredentials(new ProfileCredentialsProvider())
    .withEncryptionMaterials(materialProvider)
    .withCryptoConfiguration(cryptoConfig)
    .withRegion(clientRegion).build();

// Upload an object using the encryption client.
String origContent = "S3 Encrypted Object Using KMS-Managed Customer Master
Key.";
int origContentLength = origContent.length();
encryptionClient.putObject(bucketName, keyName, origContent);

// Download the object. The downloaded object is still encrypted.
S3Object downloadedObject = encryptionClient.getObject(bucketName, keyName);
S3ObjectInputStream input = downloadedObject.getObjectContent();

// Decrypt and read the object and close the input stream.
byte[] readBuffer = new byte[readChunkSize];
ByteArrayOutputStream baos = new ByteArrayOutputStream(readChunkSize);
int bytesRead = 0;
int decryptedContentLength = 0;

while ((bytesRead = input.read(readBuffer)) != -1) {
    baos.write(readBuffer, 0, bytesRead);
    decryptedContentLength += bytesRead;
}
input.close();

// Verify that the original and decrypted contents are the same size.
System.out.println("Original content length: " + origContentLength);
System.out.println("Decrypted content length: " + decryptedContentLength);
}
catch(AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
}
catch(SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

## Opção 2: Usar uma chave mestra no lado do cliente

Esta seção mostra como usar uma chave mestra no lado do cliente para criptografia de dados no lado do cliente.

### Important

As chaves mestras no lado do cliente e seus dados não criptografados nunca são enviados para a AWS. É importante que você gerencie com segurança suas chaves de criptografia. Se perde-las, você não poderá descriptografar os seus dados.

Como funciona:

- Ao fazer upload de um objeto—você fornece uma chave mestra no lado do cliente para o cliente de criptografia do Amazon S3. O cliente usa a chave mestra apenas para criptografar a chave de criptografia de dados que gera aleatoriamente. O processo funciona deste modo:
  1. O cliente de criptografia do Amazon S3 gera localmente uma chave simétrica de uso único (também conhecida como uma chave de criptografia de dados ou chave de dados). Ele usa a chave de dados para criptografar os dados de um único objeto do Amazon S3. O cliente gera uma chave de dados diferente para cada objeto.
  2. O cliente criptografa a chave de criptografia de dados usando a chave mestra que você forneceu. O cliente faz upload da chave de dados criptografados e de sua descrição do material como parte dos metadados de objeto. O cliente usa a descrição material para determinar qual chave mestra no lado do cliente usar para a descriptografia.
  3. O cliente faz upload dos dados criptografados para o Amazon S3 e salva a chave dos dados criptografados como metadados de objeto (`x-amz-meta-x-amz-key`) no Amazon S3.
- Ao fazer download de um objeto—O cliente faz download do objeto criptografado do Amazon S3. Usando a descrição do material nos metadados do objeto, o cliente determina qual chave mestra usar para descriptografar a chave dos dados. O cliente usa essa chave mestra para descriptografar a chave de dados e, em seguida, usa a chave de dados para descriptografar o objeto.

A chave mestra no lado do cliente que você fornece pode ser uma chave simétrica ou um par de chaves pública/privada. Os exemplos a seguir mostram como usar os tipos de chaves.

Para obter mais informações, consulte [Criptografia de dados no lado do cliente com o AWS SDK for Java e o Amazon S3](#).

#### Note

Se você receber uma mensagem de erro de criptografia quando usar a API de criptografia pela primeira vez, sua versão do JDK pode ter um arquivo de políticas de jurisdição JCE (Java Cryptography Extension) que limita o tamanho máximo da chave para transformações de criptografia e descriptografia para 128 bits. O AWS SDK requer uma chave de tamanho máximo de 256 bits. Para verificar o tamanho de chave máximo, use o método `getMaxAllowedKeyLength()` da classe `javax.crypto.Cipher`. Para remover a restrição de tamanho da chave, instale os arquivos de política de jurisdição de força ilimitada JCE (Java Cryptography Extension) na [página de download de Java SE](#).

#### Example

O exemplo a seguir mostra como realizar essas tarefas:

- Gerar uma chave AES de 256 bits
- Salve e carregue a chave AES para e do sistema de arquivo
- Use a chave AES para criptografar dados no lado do cliente antes de enviá-los para o Amazon S3
- Use a chave AES para descriptografar dados recebidos do Amazon S3
- Verifique se os dados descriptografados são os mesmos que os dados originais

Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.  
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)  
  
import java.io.ByteArrayInputStream;  
import java.io.File;  
import java.io.FileInputStream;  
import java.io.FileOutputStream;  
import java.io.IOException;
```

```
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.security.spec.InvalidKeySpecException;
import java.security.spec.X509EncodedKeySpec;

import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3EncryptionClientBuilder;
import com.amazonaws.services.s3.model.EncryptionMaterials;
import com.amazonaws.services.s3.model.ObjectMetadata;
import com.amazonaws.services.s3.model.PutObjectRequest;
import com.amazonaws.services.s3.model.S3Object;
import com.amazonaws.services.s3.model.StaticEncryptionMaterialsProvider;

public class S3ClientSideEncryptionSymMasterKey {

    public static void main(String[] args) throws Exception {
        String clientRegion = "**** Client region ****";
        String bucketName = "**** Bucket name ****";
        String objectKeyName = "**** Object key name ****";
        String masterKeyDir = System.getProperty("java.io.tmpdir");
        String masterKeyName = "secret.key";

        // Generate a symmetric 256-bit AES key.
        KeyGenerator symKeyGenerator = KeyGenerator.getInstance("AES");
        symKeyGenerator.init(256);
        SecretKey symKey = symKeyGenerator.generateKey();

        // To see how it works, save and load the key to and from the file system.
        saveSymmetricKey(masterKeyDir, masterKeyName, symKey);
        symKey = loadSymmetricAESKey(masterKeyDir, masterKeyName, "AES");

        try {
            // Create the Amazon S3 encryption client.
            EncryptionMaterials encryptionMaterials = new EncryptionMaterials(symKey);
            AmazonS3 s3EncryptionClient = AmazonS3EncryptionClientBuilder.standard()
                .withCredentials(new ProfileCredentialsProvider())
                .withEncryptionMaterials(new
            StaticEncryptionMaterialsProvider(encryptionMaterials))
                .withRegion(clientRegion)
                .build();

            // Upload a new object. The encryption client automatically encrypts it.
            byte[] plaintext = "S3 Object Encrypted Using Client-Side Symmetric Master
Key.".getBytes();
            s3EncryptionClient.putObject(new PutObjectRequest(bucketName,
                                                                objectKeyName,
                                                                new
            ByteArrayInputStream(plaintext),
                                                                new ObjectMetadata()));

            // Download and decrypt the object.
            S3Object downloadedObject = s3EncryptionClient.getObject(bucketName,
                                                                    objectKeyName);
            byte[] decrypted =
            com.amazonaws.util.IOUtils.toByteArray(downloadedObject.getObjectContent());

            // Verify that the data that you downloaded is the same as the original data.
            System.out.println("Plaintext: " + new String(plaintext));
            System.out.println("Decrypted text: " + new String(decrypted));
        }
    }
}
```

```
        }
        catch(AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        }
        catch(SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }

    private static void saveSymmetricKey(String masterKeyDir, String masterKeyName,
    SecretKey secretKey) throws IOException {
        X509EncodedKeySpec x509EncodedKeySpec = new
        X509EncodedKeySpec(secretKey.getEncoded());
        FileOutputStream keyOutputStream = new FileOutputStream(masterKeyDir +
        File.separator + masterKeyName);
        keyOutputStream.write(x509EncodedKeySpec.getEncoded());
        keyOutputStream.close();
    }

    private static SecretKey loadSymmetricAESKey(String masterKeyDir, String masterKeyName,
    String algorithm)
        throws IOException, NoSuchAlgorithmException, InvalidKeySpecException,
    InvalidKeyException {
        // Read the key from the specified file.
        File keyFile = new File(masterKeyDir + File.separator + masterKeyName);
        FileInputStream keyInputStream = new FileInputStream(keyFile);
        byte[] encodedPrivateKey = new byte[(int) keyFile.length()];
        keyInputStream.read(encodedPrivateKey);
        keyInputStream.close();

        // Reconstruct and return the master key.
        return new SecretKeySpec(encodedPrivateKey, "AES");
    }
}
```

O exemplo a seguir mostra como realizar essas tarefas:

- Gerar um par de chaves RSA de 1024 bits
- Salve e carregue as chaves RSA para o sistema de arquivo
- Use as chaves RSA para criptografar dados no lado do cliente antes de enviá-los para o Amazon S3
- Use as chaves RSA para descriptografar dados recebidos do Amazon S3
- Verifique se os dados descriptografados são os mesmos que os dados originais

Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-
developer-guide/blob/master/LICENSE-SAMPLECODE.)

import java.io.ByteArrayInputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.security.KeyFactory;
import java.security.KeyPair;
```

```
import java.security.KeyPairGenerator;
import java.security.NoSuchAlgorithmException;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.SecureRandom;
import java.security.spec.InvalidKeySpecException;
import java.security.spec.PKCS8EncodedKeySpec;
import java.security.spec.X509EncodedKeySpec;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3EncryptionClientBuilder;
import com.amazonaws.services.s3.model.EncryptionMaterials;
import com.amazonaws.services.s3.model.ObjectMetadata;
import com.amazonaws.services.s3.model.PutObjectRequest;
import com.amazonaws.services.s3.model.S3Object;
import com.amazonaws.services.s3.model.StaticEncryptionMaterialsProvider;
import com.amazonaws.util.IOUtils;

public class S3ClientSideEncryptionAsymmetricMasterKey {

    public static void main(String[] args) throws Exception {
        String clientRegion = "**** Client region ****";
        String bucketName = "**** Bucket name ****";
        String objectKeyName = "**** Key name ****";
        String rsaKeyDir = System.getProperty("java.io.tmpdir");
        String publicKeyName = "public.key";
        String privateKeyName = "private.key";

        // Generate a 1024-bit RSA key pair.
        KeyPairGenerator keyGenerator = KeyPairGenerator.getInstance("RSA");
        keyGenerator.initialize(1024, new SecureRandom());
        KeyPair origKeyPair = keyGenerator.generateKeyPair();

        // To see how it works, save and load the key pair to and from the file system.
        saveKeyPair(rsaKeyDir, publicKeyName, privateKeyName, origKeyPair);
        KeyPair keyPair = loadKeyPair(rsaKeyDir, publicKeyName, privateKeyName, "RSA");

        try {
            // Create the encryption client.
            EncryptionMaterials encryptionMaterials = new EncryptionMaterials(keyPair);
            AmazonS3 s3EncryptionClient = AmazonS3EncryptionClientBuilder.standard()
                .withCredentials(new ProfileCredentialsProvider())
                .withEncryptionMaterials(new
                    StaticEncryptionMaterialsProvider(encryptionMaterials))
                .withRegion(clientRegion)
                .build();

            // Create a new object.
            byte[] plaintext = "S3 Object Encrypted Using Client-Side Asymmetric Master
Key.".getBytes();
            S3Object object = new S3Object();
            object.setKey(objectKeyName);
            object.setObjectContent(new ByteArrayInputStream(plaintext));
            ObjectMetadata metadata = new ObjectMetadata();
            metadata.setContentLength(plaintext.length);

            // Upload the object. The encryption client automatically encrypts it.
            PutObjectRequest putRequest = new PutObjectRequest(bucketName,
                object.getKey(),
                object.getObjectContent(),
                metadata);
            s3EncryptionClient.putObject(putRequest);
        }
    }
}
```

```
// Download and decrypt the object.
S3Object downloadedObject = s3EncryptionClient.getObject(bucketName,
object.getKey());
byte[] decrypted = IOUtils.toByteArray(downloadedObject.getObjectContent());

// Verify that the data that you downloaded is the same as the original data.
System.out.println("Plaintext: " + new String(plaintext));
System.out.println("Decrypted text: " + new String(decrypted));
}

catch(AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
}

catch(SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}

private static void saveKeyPair(String dir,
                                String publicKeyName,
                                String privateKeyName,
                                KeyPair keyPair) throws IOException {
    PrivateKey privateKey = keyPair.getPrivate();
    PublicKey publicKey = keyPair.getPublic();

    // Write the public key to the specified file.
    X509EncodedKeySpec x509EncodedKeySpec = new
X509EncodedKeySpec(publicKey.getEncoded());
    FileOutputStream publicKeyOutputStream = new FileOutputStream(dir + File.separator
+ publicKeyName);
    publicKeyOutputStream.write(x509EncodedKeySpec.getEncoded());
    publicKeyOutputStream.close();

    // Write the private key to the specified file.
    PKCS8EncodedKeySpec pkcs8EncodedKeySpec = new
PKCS8EncodedKeySpec(privateKey.getEncoded());
    FileOutputStream privateKeyOutputStream = new FileOutputStream(dir + File.separator
+ privateKeyName);
    privateKeyOutputStream.write(pkcs8EncodedKeySpec.getEncoded());
    privateKeyOutputStream.close();
}

private static KeyPair loadKeyPair(String dir,
                                    String publicKeyName,
                                    String privateKeyName,
                                    String algorithm)
throws IOException, NoSuchAlgorithmException, InvalidKeySpecException {
    // Read the public key from the specified file.
    File publicKeyFile = new File(dir + File.separator + publicKeyName);
    FileInputStream publicKeyInputStream = new FileInputStream(publicKeyFile);
    byte[] encodedPublicKey = new byte[(int) publicKeyFile.length()];
    publicKeyInputStream.read(encodedPublicKey);
    publicKeyInputStream.close();

    // Read the private key from the specified file.
    File privateKeyFile = new File(dir + File.separator + privateKeyName);
    FileInputStream privateKeyInputStream = new FileInputStream(privateKeyFile);
    byte[] encodedPrivateKey = new byte[(int) privateKeyFile.length()];
    privateKeyInputStream.read(encodedPrivateKey);
    privateKeyInputStream.close();

    // Convert the keys into a key pair.
    KeyFactory keyFactory = KeyFactory.getInstance(algorithm);
```

```
X509EncodedKeySpec publicKeySpec = new X509EncodedKeySpec(encodedPublicKey);
PublicKey publicKey = keyFactory.generatePublic(publicKeySpec);

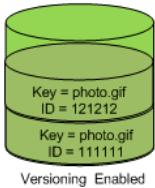
PKCS8EncodedKeySpec privateKeySpec = new PKCS8EncodedKeySpec(encodedPrivateKey);
PrivateKey privateKey = keyFactory.generatePrivate(privateKeySpec);

return new KeyPair(publicKey, privateKey);
}
}
```

## Usar versionamento

Versionamento é um meio de manter diversas variantes de um objeto no mesmo bucket. O versionamento pode ser usado para preservar, recuperar e restaurar todas as versões de cada objeto armazenado no bucket do Amazon S3. Com o versionamento, você pode se recuperar, facilmente, de ações não intencionais do usuário e de falhas de aplicativo.

Em um bucket, por exemplo, você pode ter dois objetos com a mesma chave, mas diferentes IDs de versão, como `photo.gif` (versão 111111) e `photo.gif` (versão 121212).



Buckets com versionamento habilitado permitem que você recupere objetos de uma exclusão ou substituição acidental. Por exemplo:

- Se você excluir um objeto em vez de removê-lo permanentemente, o Amazon S3 inserirá um marcador de exclusão, tornando o objeto a versão atual. Você sempre pode restaurar a versão anterior. Para obter mais informações, consulte [Exclusão de versões de objeto \(p. 461\)](#).
- Se você substituir um objeto, isso criará uma nova versão do objeto no bucket. Você sempre pode restaurar a versão anterior.

### Important

Se você tem uma política de ciclo de vida de expiração do objeto em seu bucket sem versão e quer manter o mesmo comportamento de exclusão permanente quando ativar o controle de versão, precisará adicionar uma política de expiração de versão desatualizada. A política de expiração do ciclo de vida gerenciará as exclusões de versões desatualizadas de objeto no bucket habilitado para versão. (Um bucket habilitado para versão mantém uma versão atual e zero ou mais versões desatualizadas de objeto.) Para mais informações, consulte [Como faço para criar uma política de ciclo de vida para um bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

Os buckets podem estar em um de três estados: sem versionamento (o padrão), habilitado para versionamento ou com versionamento suspenso.

### Important

Depois que um bucket é habilitado para versionamento, ele nunca pode voltar a um estado sem versionamento. Você pode, contudo, suspender o versionamento nesse bucket.

O estado de versionamento aplica-se a todos (nunca alguns) os objetos nesse bucket. Na primeira vez que você habilita o versionamento de um bucket, os objetos nele passam a ter versões e recebem um ID de versão único. Observe o seguinte:

- Os objetos armazenados em seu bucket antes da habilitação do versionamento têm um ID de versão `null`. Quando você habilita o versionamento, os objetos existentes em seu bucket não são alterados. O que muda é como o Amazon S3 trata os objetos em solicitações futuras. Para obter mais informações, consulte [Gerenciamento de objetos em um bucket com versionamento ativado \(p. 453\)](#).
- O proprietário do bucket (ou qualquer usuário com as devidas permissões) pode suspender o versionamento para interromper o acúmulo de versões de objetos. Quando você suspende o versionamento, os objetos existentes em seu bucket não são alterados. O que muda é como o Amazon S3 trata os objetos em solicitações futuras. Para obter mais informações, consulte [Gerenciamento de objetos em um bucket com versionamento suspenso \(p. 467\)](#).

## Como configurar o versionamento de um bucket

Você pode configurar o versionamento do bucket usando qualquer um dos seguintes métodos:

- Configure o versionamento usando o console do Amazon S3.
- Configure o versionamento com programação usando os AWS SDKs.

O console e os SDKs chamam a API REST que o Amazon S3 oferece para gerenciar o versionamento.

### Note

Se necessário, você também pode chamar a API REST do Amazon S3 diretamente do seu código. Contudo, isso pode ser complicado, porque exige que você grave código para autenticar suas solicitações.

Cada bucket que você cria tem um sub-recurso de versionamento (consulte [Opções de configuração de bucket \(p. 57\)](#)) associado a ele. Por padrão, seu bucket não tem versões e, consequentemente, o sub-recurso de versionamento armazena uma configuração vazia de versionamento.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
</VersioningConfiguration>
```

Para habilitar o versionamento, você envia uma solicitação ao Amazon S3 com uma configuração de versionamento que inclui um status.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>Enabled</Status>
</VersioningConfiguration>
```

Para suspender o versionamento, você define o valor do status como `Suspended`.

O proprietário do bucket, uma conta da AWS que criou o bucket (conta raiz) e usuários autorizados podem configurar o estado de versionamento de um bucket. Para obter mais informações sobre permissões, consulte [Gerenciamento de permissões de acesso aos recursos do Amazon S3 \(p. 282\)](#).

Para ver um exemplo de configuração de versionamento, consulte [Exemplos de ativação de versionamento de bucket \(p. 451\)](#).

## Exclusão de MFA

Se desejar, você pode adicionar outra camada de segurança, configurando um bucket para habilitar a exclusão de MFA (autenticação multifator), o que requer autenticação adicional para qualquer uma das seguintes operações:

- Alterar o estado de versionamento de seu bucket

- Excluir, permanentemente, uma versão de objeto

A exclusão de MFA exige duas formas de autenticação:

- Suas credenciais de segurança
- A concatenação de um número serial válido, um espaço e o código de seis dígitos exibidos em um dispositivo de autenticação aprovado

A exclusão de MFA oferece segurança adicional, por exemplo, caso suas credenciais de segurança sejam comprometidas.

Para habilitar ou desabilitar a Exclusão de MFA, use a mesma API utilizada para configurar o versionamento em um bucket. O Amazon S3 armazena a configuração de Exclusão de MFA no mesmo sub-recurso de versionamento que armazena o status de versionamento do bucket.

A exclusão de MFA pode ajudar a evitar exclusões acidentais de bucket já que a pessoa que inicia a ação de exclusão deve:

- Provar a posse física de um dispositivo MFA com um código MFA e
- adicionar uma camada extra de fricção e segurança à ação de exclusão

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>VersioningState</Status>
  <MfaDelete>MfaDeleteState</MfaDelete>
</VersioningConfiguration>
```

Para usar a exclusão de MFA, você pode usar um dispositivo de hardware ou MFA virtual para gerar um código de autenticação. O exemplo a seguir mostra um código de autenticação gerado exibido em um dispositivo de hardware.



#### Note

A exclusão de MFA e o acesso à API protegido por MFA são recursos que visam fornecer proteção para diferentes cenários. Você configura a exclusão de MFA em um bucket para garantir que os dados em seu bucket não sejam excluídos acidentalmente. O acesso à API protegido por MFA é usado para impor outro fator de autenticação (código MFA) ao acessar recursos confidenciais do Amazon S3. Você pode solicitar que quaisquer operações nesses recursos do Amazon S3 sejam feitas com credenciais temporárias criadas com MFA. Para ver um exemplo, consulte [Adição de uma política de bucket para exigir MFA \(p. 362\)](#).

Para obter mais informações sobre como adquirir e ativar um dispositivo de autenticação, consulte <https://aws.amazon.com/iam/details/mfa>.

#### Note

O proprietário do bucket, a conta da AWS que criou o bucket (conta raiz) e todos os usuários do IAM autorizados podem habilitar o versionamento, mas apenas o proprietário do bucket (conta raiz) pode habilitar a exclusão de MFA.

## Tópicos relacionados

Para obter mais informações, consulte os tópicos a seguir:

- [Exemplos de ativação de versionamento de bucket \(p. 451\)](#)

- Gerenciamento de objetos em um bucket com versionamento ativado (p. 453)
- Gerenciamento de objetos em um bucket com versionamento suspenso (p. 467)
- Aumentos significativos em respostas HTTP 503 para solicitações do Amazon S3 para buckets com o versionamento habilitado (p. 621)

## Exemplos de ativação de versionamento de bucket

### Tópicos

- [Usar o console do Amazon S3 \(p. 451\)](#)
- [Using the AWS SDK for Java \(p. 451\)](#)
- [Using the AWS SDK for .NET \(p. 452\)](#)
- [Uso de outros AWS SDKs \(p. 453\)](#)

Esta seção fornece exemplos de como habilitar o versionamento de um bucket. Os exemplos primeiro ativam o versionamento de um bucket e depois recuperam o status de versionamento. Para obter uma introdução, consulte [Usar versionamento \(p. 448\)](#).

### Usar o console do Amazon S3

Para obter mais informações sobre ativação de versionamento em um bucket usando o console do Amazon S3, consulte [Como faço para habilitar ou suspender o versionamento em um bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

### Using the AWS SDK for Java

#### Example

Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

```
import java.io.IOException;

import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Region;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.model.AmazonS3Exception;
import com.amazonaws.services.s3.model.BucketVersioningConfiguration;
import com.amazonaws.services.s3.model.SetBucketVersioningConfigurationRequest;

public class BucketVersioningConfigurationExample {
    public static String bucketName = "*** bucket name ***";
    public static AmazonS3Client s3Client;

    public static void main(String[] args) throws IOException {
        s3Client = new AmazonS3Client(new ProfileCredentialsProvider());
        s3Client.setRegion(Region.getRegion(Regions.US_EAST_1));
        try {

            // 1. Enable versioning on the bucket.
            BucketVersioningConfiguration configuration =
                new BucketVersioningConfiguration().withStatus("Enabled");

            SetBucketVersioningConfigurationRequest setBucketVersioningConfigurationRequest =
                new SetBucketVersioningConfigurationRequest(bucketName, configuration);

            s3Client.setBucketVersioningConfiguration(setBucketVersioningConfigurationRequest);
        }
    }
}
```

```
// 2. Get bucket versioning configuration information.  
BucketVersioningConfiguration conf =  
s3Client.getBucketVersioningConfiguration(bucketName);  
System.out.println("bucket versioning configuration status: " + conf.getStatus());  
  
} catch (AmazonS3Exception amazonS3Exception) {  
    System.out.format("An Amazon S3 error occurred. Exception: %s",  
amazonS3Exception.toString());  
} catch (Exception ex) {  
    System.out.format("Exception: %s", ex.toString());  
}  
}  
}  
}
```

## Using the AWS SDK for .NET

Para obter informações sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

### Example

```
using System;  
using Amazon.S3;  
using Amazon.S3.Model;  
  
namespace s3.amazon.com.docsamples  
{  
    class BucketVersioningConfiguration  
    {  
        static string bucketName = "*** bucket name ***";  
  
        public static void Main(string[] args)  
        {  
            using (var client = new AmazonS3Client(Amazon.RegionEndpoint.USEast1))  
            {  
                try  
                {  
                    EnableVersioningOnBucket(client);  
                    string bucketVersioningStatus =  
RetrieveBucketVersioningConfiguration(client);  
                }  
                catch (AmazonS3Exception amazonS3Exception)  
                {  
                    if (amazonS3Exception.ErrorCode != null &&  
                        (amazonS3Exception.ErrorCode.Equals("InvalidAccessKeyId")  
                        ||  
                        amazonS3Exception.ErrorCode.Equals("InvalidSecurity")))  
                    {  
                        Console.WriteLine("Check the provided AWS Credentials.");  
                        Console.WriteLine(  
                            "To sign up for service, go to http://aws.amazon.com/s3");  
                    }  
                    else  
                    {  
                        Console.WriteLine(  
                            "Error occurred. Message:{0} when listing objects",  
                            amazonS3Exception.Message);  
                    }  
                }  
            }  
            Console.WriteLine("Press any key to continue...");  
            Console.ReadKey();  
        }  
    }  
}
```

```
}

static void EnableVersioningOnBucket(IAmazonS3 client)
{
    PutBucketVersioningRequest request = new PutBucketVersioningRequest
    {
        BucketName = bucketName,
        VersioningConfig = new S3BucketVersioningConfig
        {
            Status = VersionStatus.Enabled
        }
    };

    PutBucketVersioningResponse response = client.PutBucketVersioning(request);
}

static string RetrieveBucketVersioningConfiguration(IAmazonS3 client)
{
    GetBucketVersioningRequest request = new GetBucketVersioningRequest
    {
        BucketName = bucketName
    };

    GetBucketVersioningResponse response = client.GetBucketVersioning(request);
    return response.VersioningConfig.Status;
}
}
```

## Uso de outros AWS SDKs

Para obter informações sobre como usar outros AWS SDKs, consulte [Código de exemplo e bibliotecas](#).

## Gerenciamento de objetos em um bucket com versionamento ativado

### Tópicos

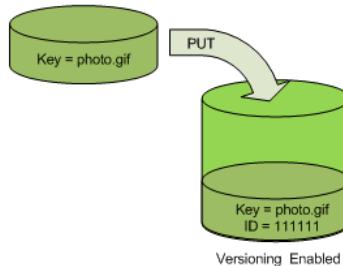
- [Adição de objetos a buckets com versionamento ativado \(p. 453\)](#)
- [Listagem de objetos em um bucket com versionamento ativado \(p. 455\)](#)
- [Recuperação de versões de objeto \(p. 459\)](#)
- [Exclusão de versões de objeto \(p. 461\)](#)
- [Transição de versões de objeto \(p. 466\)](#)
- [Restauração de versões anteriores \(p. 466\)](#)
- [Permissões de objeto com versões \(p. 467\)](#)

Os objetos que são armazenados em seu bucket antes da habilitação do versionamento têm um ID de versão null. Quando você habilita o versionamento, os objetos existentes em seu bucket não são alterados. O que muda é como o Amazon S3 trata os objetos em solicitações futuras. Os tópicos nesta seção explicam várias operações de objeto em um bucket com versionamento ativado.

## Adição de objetos a buckets com versionamento ativado

Assim que você ativa o versionamento em um bucket, o Amazon S3 adiciona, automaticamente, um ID de versão único a cada objeto armazenado (usando PUT, POST ou COPY) no bucket.

A figura a seguir mostra que o Amazon S3 adiciona um ID de versão único a um objeto quando este é adicionado a um bucket com versionamento ativado.



## Tópicos

- [Usar o console \(p. 454\)](#)
- [Uso dos AWS SDKs \(p. 454\)](#)
- [Uso dos REST API \(p. 454\)](#)

## Usar o console

Para obter instruções, consulte [Como faço upload de um objeto em um bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

## Uso dos AWS SDKs

Para ver exemplos de upload de objetos usando os AWS SDKs para Java, .NET e PHP, consulte [Upload de objetos \(p. 175\)](#). Os exemplos de upload de objetos em buckets com e sem versionamento ativado são os mesmos, embora o Amazon S3 atribua um número de versão para buckets com versionamento ativado. Caso contrário, o número de versão é nulo.

Para obter informações sobre como usar outros AWS SDKs, consulte [Código de exemplo e bibliotecas](#).

## Uso dos REST API

### Adição de objetos a buckets com versionamento ativado

1	Ative o versionamento de um bucket usando uma solicitação <code>PUT Bucket versioning</code> . Para obter mais informações, consulte <a href="#">PUT em bucket com versionamento</a> .
2	Envie uma solicitação <code>PUT</code> , <code>POST</code> ou <code>COPY</code> para armazenar um objeto no bucket.

Quando você adiciona um objeto a um bucket com versionamento ativado, o Amazon S3 retorna o ID de versão do objeto no cabeçalho de resposta `x-amz-version-id`, por exemplo:

```
x-amz-version-id: 3/L4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY
```

### Note

As taxas normais do Amazon S3 aplicam-se a cada versão de um objeto armazenado e transferido. Cada versão de um objeto é um objeto inteiro; não é apenas um diff da versão anterior. Assim, se você tiver três versões de um objeto armazenado, será cobrado pelos três objetos.

### Note

Os valores de ID de versão que o Amazon S3 atribui são seguros para URL (podem ser incluídos como parte de um URI).

## Listagem de objetos em um bucket com versionamento ativado

### Tópicos

- [Usar o console \(p. 455\)](#)
- [Uso dos AWS SDKs \(p. 455\)](#)
- [Uso dos REST API \(p. 458\)](#)

Esta seção fornece um exemplo de como listar versões de objeto a partir de um bucket habilitado para versionamento. O Amazon S3 armazena informações de versão de objeto no sub-recurso `versões` (consulte [Opções de configuração de bucket \(p. 57\)](#)) que está associado ao bucket.

### Usar o console

Para obter informações sobre listagem de versões de objeto usando o console do Amazon S3, consulte [Como visualizo as versões de um objeto do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

### Uso dos AWS SDKs

Os exemplos nesta seção mostram como recuperar uma listagem de objetos de um bucket com versionamento ativado. Cada solicitação retorna até 1.000 versões, a menos que você especifique um número menor. Se o bucket contiver mais versões que esse limite, envie uma série de solicitações para recuperar a lista de todas as versões. Esse processo de retornar resultados em “páginas” é chamado paginação. Para mostrar como a paginação funciona, os exemplos limitam cada resposta a duas versões de objeto. Depois de recuperar a primeira página de resultados, cada exemplo verifica se a lista de versões foi truncada. Em caso afirmativo, o exemplo continua a recuperar páginas até que todas as versões tenham sido recuperadas.

#### Note

Os exemplos a seguir também funcionam com bucket sem versionamento, ou para objetos sem versões individuais. Nesses casos, o Amazon S3 retorna a listagem de objetos com um ID de versão `null`.

Para obter informações sobre como usar outros AWS SDKs, consulte [Código de exemplo e bibliotecas](#).

### Usar o AWS SDK for Java

Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

#### Example

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.  
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)  
  
import java.io.IOException;  
  
import com.amazonaws.AmazonServiceException;  
import com.amazonaws.SdkClientException;  
import com.amazonaws.auth.profile.ProfileCredentialsProvider;  
import com.amazonaws.services.s3.AmazonS3;  
import com.amazonaws.services.s3.AmazonS3ClientBuilder;  
import com.amazonaws.services.s3.model.ListVersionsRequest;  
import com.amazonaws.services.s3.model.S3VersionSummary;  
import com.amazonaws.services.s3.model.VersionListing;  
  
public class ListKeysVersioningEnabledBucket {
```

```
public static void main(String[] args) throws IOException {
    String clientRegion = "**** Client region ****";
    String bucketName = "**** Bucket name ****";

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

        // Retrieve the list of versions. If the bucket contains more versions
        // than the specified maximum number of results, Amazon S3 returns
        // one page of results per request.
        ListVersionsRequest request = new ListVersionsRequest()
            .withBucketName(bucketName)
            .withMaxResults(2);
        VersionListing versionListing = s3Client.listVersions(request);
        int numVersions = 0, numPages = 0;
        while(true) {
            numPages++;
            for (S3VersionSummary objectSummary :
                versionListing.getVersionSummaries()) {
                System.out.printf("Retrieved object %s, version %s\n",
                    objectSummary.getKey(),
                    objectSummary.getVersionId());
                numVersions++;
            }
            // Check whether there are more pages of versions to retrieve. If
            // there are, retrieve them. Otherwise, exit the loop.
            if(versionListing.isTruncated()) {
                versionListing = s3Client.listNextBatchOfVersions(versionListing);
            }
            else {
                break;
            }
        }
        System.out.println(numVersions + " object versions retrieved in " + numPages +
" pages");
    }
    catch(AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    }
    catch(SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}
```

## Using the AWS SDK for .NET

Para obter informações sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

### Example

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-
developer-guide/blob/master/LICENSE-SAMPLECODE.)
```

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

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

        public static void Main(string[] args)
        {
            s3Client = new AmazonS3Client(bucketRegion);
            GetObjectListWithAllVersionsAsync().Wait();
        }

        static async Task GetObjectListWithAllVersionsAsync()
        {
            try
            {
                ListVersionsRequest request = new ListVersionsRequest()
                {
                    BucketName = bucketName,
                    // You can optionally specify key name prefix in the request
                    // if you want list of object versions of a specific object.

                    // For this example we limit response to return list of 2 versions.
                    MaxKeys = 2
                };
                do
                {
                    ListVersionsResponse response = await
s3Client.ListVersionsAsync(request);
                    // Process response.
                    foreach (S3ObjectVersion entry in response.Versions)
                    {
                        Console.WriteLine("key = {0} size = {1}",
                            entry.Key, entry.Size);
                    }

                    // If response is truncated, set the marker to get the next
                    // set of keys.
                    if (response.IsTruncated)
                    {
                        request.KeyMarker = response.NextKeyMarker;
                        request.VersionIdMarker = response.NextVersionIdMarker;
                    }
                    else
                    {
                        request = null;
                    }
                } while (request != null);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when writing
an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
        }
    }
}
```

```
    }
}
}
```

## Uso dos REST API

Para listar todas as versões de todos os objetos em um bucket, use o sub-recurso `versions` em uma solicitação GET Bucket. O Amazon S3 pode recuperar até 1.000 objetos no máximo, e cada versão de objeto conta totalmente com um objeto. Portanto, se um bucket contiver duas chaves (por exemplo, `photo.gif` e `picture.jpg`) e a primeira chave tiver 990 versões e a segunda chave tiver 400 versões, uma solicitação única recuperará as 990 versões de `photo.gif` e apenas as 10 versões mais recentes de `picture.jpg`.

O Amazon S3 retorna as versões de objeto na ordem em que foram armazenadas, com as últimas armazenadas sendo retornadas primeiro.

Para listar todas as versões de objeto em um bucket

- Em uma solicitação GET Bucket, inclua o sub-recurso `versions`.

```
GET /?versions HTTP/1.1
Host: bucketName.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 +0000
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

## Recuperação de um subconjunto de objetos em um bucket

Esta seção discute os seguintes dois cenários de exemplo:

- Você deseja recuperar um subconjunto de todas as versões de objeto em um bucket, por exemplo, recuperar todas as versões de um objeto específico.
- O número de versões de objeto na resposta excede o valor de `max-key` (1.000 por padrão), de modo que você tem que enviar uma segunda solicitação para recuperar as versões de objeto remanescentes.

Para recuperar um subconjunto de versões de objeto, você usa os parâmetros de solicitação para GET Bucket. Para obter mais informações, consulte [GET Bucket](#).

### Exemplo 1: recuperação de todas as versões de apenas um objeto específico

Você pode recuperar todas as versões de um objeto, usando o sub-recurso `versions` e o parâmetro de solicitação `prefix` usando o processo a seguir. Para obter mais informações sobre `prefix`, consulte [GET Bucket](#).

#### Recuperação de todas as versões de uma chave

1	Defina o parâmetro <code>prefix</code> como a chave do objeto que você deseja recuperar.
2	Envie uma solicitação GET Bucket usando o sub-recurso <code>versions</code> e <code>prefix</code> . <code>GET /?versions&amp;prefix=objectName HTTP/1.1</code>

#### Example Recuperação de objetos usando um prefixo

O exemplo a seguir recupera objetos cuja chave é ou começa com `myObject`.

```
GET /?versions&prefix=myObject HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Você pode usar os outros parâmetros de solicitação para recuperar um subconjunto de todas as versões do objeto. Para obter mais informações, consulte [GET Bucket](#).

#### Exemplo 2: recuperação de uma listagem de objetos adicionais se a resposta estiver truncada

Se o número de objetos que podem ser retornados em uma solicitação GET exceder o valor de `max-keys`, a resposta conterá `<isTruncated>true</isTruncated>` e incluirá a primeira chave (em `NextKeyMarker`) e o primeiro ID de versão (em `NextVersionIdMarker`) que satisfazem a solicitação, mas que não foram retornados. Você usa esses valores retornados como a posição de início em uma solicitação subsequente para recuperar os objetos adicionais que satisfazem a solicitação GET.

Use o seguinte processo para recuperar os objetos adicionais que satisfazem a solicitação `GET Bucket versions` original de um bucket. Para obter mais informações sobre `key-marker`, `version-id-marker`, `NextKeyMarker` e `NextVersionIdMarker`, consulte [GET Bucket](#).

#### Recuperação de respostas adicionais que satisfazem a solicitação GET original

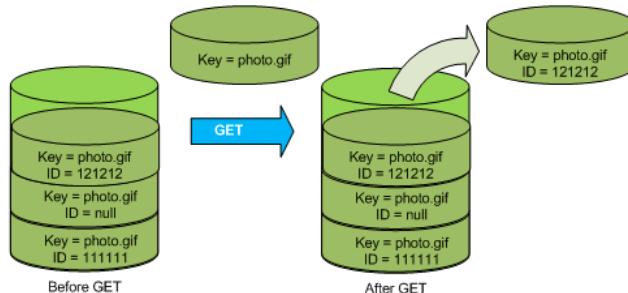
1	Defina o valor de <code>key-marker</code> como a chave retornada em <code>NextKeyMarker</code> na resposta anterior.
2	Defina o valor de <code>version-id-marker</code> como o ID de versão retornado em <code>NextVersionIdMarker</code> na resposta anterior.
3	Envie uma solicitação <code>GET Bucket versions</code> usando <code>key-marker</code> e <code>version-id-marker</code> .

#### Example Recuperação de objetos começando com a chave e o ID de versão especificados

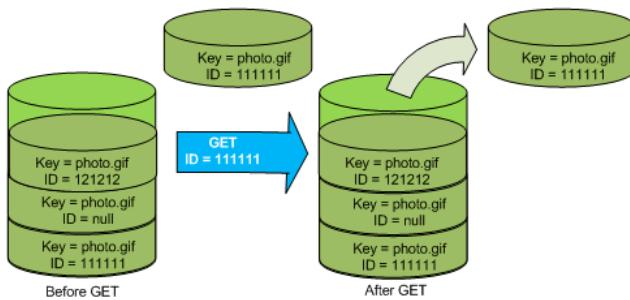
```
GET /?versions&key-marker=myObject&version-id-marker=298459348571 HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

## Recuperação de versões de objeto

Uma solicitação GET simples recupera a versão atual de um objeto. A figura a seguir mostra como o GET retorna a versão atual do objeto, `photo.gif`.



Para recuperar uma versão específica, você tem que especificar seu ID de versão. A figura a seguir mostra que a solicitação `GET versionId` recupera a versão especificada do objeto (não necessariamente a versão atual).



## Usar o console

Para obter instruções, consulte [Como visualizo as versões de um objeto do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

## Uso dos AWS SDKs

Para ver exemplos de upload de objetos usando os AWS SDKs para Java, .NET e PHP, consulte [Obtenção de objetos \(p. 166\)](#). Os exemplos de upload de objetos em buckets com e sem versionamento ativado são os mesmos, embora o Amazon S3 atribua um número de versão para buckets com versionamento ativado. Caso contrário, o número de versão é nulo.

Para obter informações sobre como usar outros AWS SDKs, consulte [Código de exemplo e bibliotecas](#).

## Uso de REST

Para recuperar uma versão específica do objeto

1. Defina `versionId` como o ID da versão do objeto que você deseja recuperar.
2. Envie uma solicitação `GET Object versionId`.

### Example Recuperação de um objeto com versões

A solicitação a seguir recupera a versão L4kqtJlcpXroDTDmpUMLUo de `my-image.jpg`.

```
GET /my-image.jpg?versionId=L4kqtJlcpXroDTDmpUMLUo HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

## Tópicos relacionados

[Recuperação de metadados de uma versão de objeto \(p. 460\)](#)

### Recuperação de metadados de uma versão de objeto

Se você quiser recuperar apenas os metadados (e não o conteúdo) de um objeto, use a operação `HEAD`. Por padrão, você obtém os metadados da versão mais recente. Para recuperar os metadados de um objeto específico, você especifica seu ID de versão.

Para recuperar os metadados de uma versão do objeto

1. Defina `versionId` como o ID da versão do objeto cujos metadados você deseja recuperar.
2. Envie uma solicitação `HEAD Object versionId`.

### Example Recuperação de metadados de um objeto com versões

A solicitação a seguir recupera os metadados da versão 3HL4kqCxf3vjVBH40Nrjfkd de my-image.jpg.

```
HEAD /my-image.jpg?versionId=3HL4kqCxf3vjVBH40Nrjfkd HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

A seguir, um exemplo de resposta.

```
HTTP/1.1 200 OK
x-amz-id-2: ef8yU9AS1ed4OpIszj7UDNEHGran
x-amz-request-id: 318BC8BC143432E5
x-amz-version-id: 3HL4kqtJlcpXroDTDmjVBH40Nrjfkd
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2006 12:00:00 GMT
ETag: "fba9dede5f27731c9771645a39863328"
Content-Length: 434234
Content-Type: text/plain
Connection: close
Server: AmazonS3
```

## Exclusão de versões de objeto

Você pode excluir versões de objeto sempre que quiser. Além disso, você também pode definir regras de configuração de ciclo de vida de objetos que têm um ciclo de vida bem definido para solicitar que o Amazon S3 expire as versões de objeto atuais ou que remova permanentemente versões anteriores do objeto. Quando seu bucket tem versionamento ativado ou suspenso, as ações da configuração do ciclo de vida funcionam do seguinte modo:

- A ação `Expiration` aplica-se à versão atual do objeto e, em vez de excluir a versão atual do objeto, o Amazon S3 a retém como uma versão antiga por adicionar um marcador de exclusão, que se torna a versão atual.
- A ação `NoncurrentVersionExpiration` aplica-se a versões antigas do objeto, e o Amazon S3 remove permanentemente essas versões do objeto. Você não pode recuperar objetos removidos permanentemente.

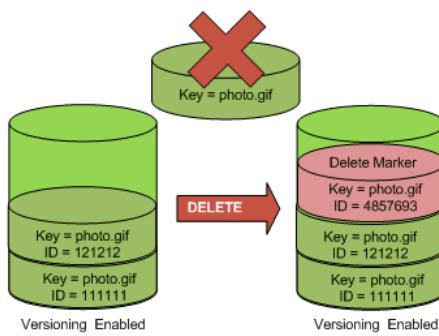
Para obter mais informações, consulte [Gerenciamento do ciclo de vida de objetos \(p. 122\)](#).

Uma solicitação `DELETE` tem os seguintes casos de uso:

- Quando o versionamento está habilitado, um `DELETE` simples não pode excluir permanentemente um objeto.

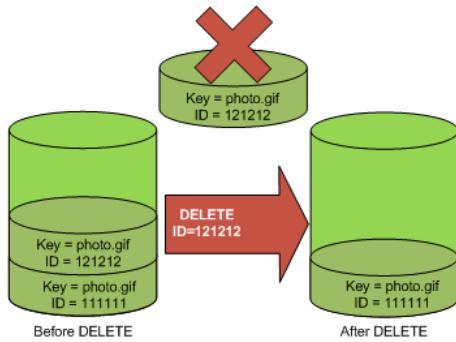
Em vez disso, o Amazon S3 insere um marcador de exclusão no bucket, que se torna a versão atual do objeto com um novo ID. Quando você tenta uma solicitação `GET` em um objeto cuja versão atual é um marcador de exclusão, o Amazon S3 se comporta como se o objeto tivesse sido excluído (mesmo que não tenha sido apagado) e retorna um erro 404.

A figura a seguir mostra que uma simples solicitação `DELETE` não remove de fato o objeto especificado. Em vez disso, o Amazon S3 insere um marcador de exclusão.



- Para excluir permanentemente objetos com versões, você deve usar `DELETE Object versionId`.

A figura a seguir mostra que excluir uma versão do objeto especificada remove permanentemente esse objeto.



## Usar o console

Para obter instruções, consulte [Como visualizo as versões de um objeto do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

## Uso dos AWS SDKs

Para ver exemplos de upload de objetos usando os AWS SDKs para Java, .NET e PHP, consulte [Excluir objetos \(p. 237\)](#). Os exemplos de upload de objetos em buckets com e sem versionamento ativado são os mesmos, embora o Amazon S3 atribua um número de versão para buckets com versionamento ativado. Caso contrário, o número de versão é nulo.

Para obter informações sobre como usar outros AWS SDKs, consulte [Código de exemplo e bibliotecas](#).

## Uso de REST

Para excluir uma versão específica de um objeto

- Em `DELETE`, especifique o ID da versão.

### Example Exclusão de uma versão específica

Os exemplos a seguir mostram como excluir a versão UIORUnfnd89493jJFJ de `photo.gif`.

```
DELETE /photo.gif?versionId=UIORUnfnd89493jJFJ HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
```

```
Authorization: AWS AKIAIOSFODNN7EXAMPLE:xQE0diMbLRepdf3YB+FIEEXAMPLE=
Content-Type: text/plain
Content-Length: 0
```

## Tópicos relacionados

[Uso de exclusão de MFA \(p. 463\)](#)

[Trabalho com marcadores de exclusão \(p. 463\)](#)

[Remoção de marcadores de exclusão \(p. 464\)](#)

[Usar versionamento \(p. 448\)](#)

## Uso de exclusão de MFA

Se a configuração de versionamento do bucket tiver a exclusão de MFA ativada, o proprietário do bucket deverá incluir o cabeçalho de solicitação `x-amz-mfa` nas solicitações para excluir permanentemente uma versão de objeto ou alterar o estado de versionamento do bucket. As solicitações que incluem `x-amz-mfa` devem usar HTTPS. O valor do cabeçalho é uma concatenação do número de série do seu dispositivo de autenticação, um espaço e o código de autenticação exibido nele. Se você não incluir esse cabeçalho, a solicitação falhará.

Para obter mais informações sobre dispositivos de autenticação, consulte <https://aws.amazon.com/iam/details/mfa/>.

Example Exclusão de um objeto de um bucket com exclusão de MFA ativada

O exemplo a seguir mostra como excluir `my-image.jpg` (com a versão especificada), de um bucket configurado com exclusão de MFA ativada. Note o espaço entre `[SerialNumber]` e `[AuthenticationCode]`. Para obter mais informações, consulte [Objeto DELETE](#).

```
DELETE /my-image.jpg?versionId=3HL4kqCxf3vjVBH40Nrjfkd HTTPS/1.1
Host: bucketName.s3.amazonaws.com
x-amz-mfa: 20899872 301749
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Para obter mais informações sobre ativação de exclusão de MFA, consulte [Exclusão de MFA \(p. 449\)](#).

## Trabalho com marcadores de exclusão

Um marcador de exclusão é um espaço reservado (marcador) para um objeto com versões que foi nomeado em uma solicitação `DELETE` simples. Como o objeto estava em um bucket com versionamento ativado, o objeto não foi excluído. O marcador de exclusão, contudo, faz com que o Amazon S3 se comporte como se este tivesse sido excluído.

Um marcador de exclusão tem um nome de chave (ou chave) e um ID de versão como qualquer outro objeto. Contudo, um marcador de exclusão difere de outros objetos nas seguintes maneiras:

- Ele não tem dados associados.
- Ele não é associado a um valor de lista de controle de acesso (ACL).
- Ele não recupera nada em uma solicitação `GET` porque não tem dados; você recebe um erro 404.
- A única operação que você pode usar em um marcador de exclusão é `DELETE` e apenas o proprietário do bucket pode emitir tal solicitação.

Marcadores de exclusão incorrem em uma cobrança nominal de armazenamento no Amazon S3. O tamanho de armazenamento de um marcador de exclusão é igual ao tamanho do nome da chave do

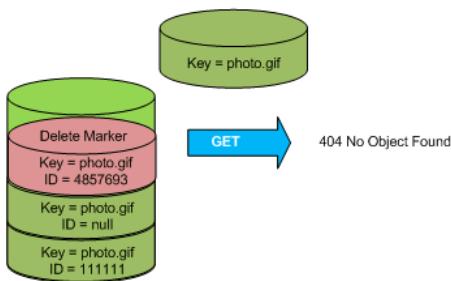
marcador de exclusão. Um nome de chave é uma sequência de caracteres Unicode. A codificação UTF-8 adiciona de 1 a 4 bytes de armazenamento ao seu bucket para cada caractere no nome. Para obter mais informações sobre nomes de chave, consulte [Chaves de objeto \(p. 102\)](#). Para obter mais informações sobre exclusão de um marcador de exclusão, consulte [Remoção de marcadores de exclusão \(p. 464\)](#).

Somente o Amazon S3 pode criar e excluir um marcador de exclusão, e ele faz isso sempre que você envia uma solicitação `DELETE Object` em um objeto em um bucket com versionamento ativado ou suspenso. O objeto nomeado na solicitação `DELETE` não é de fato excluído. Em vez disso, o marcador de exclusão torna-se a versão atual do objeto. O nome de chave (ou chave) do objeto torna-se a chave do marcador de exclusão. Se você tentar obter um objeto e sua versão atual for um marcador de exclusão, o Amazon S3 responderá com:

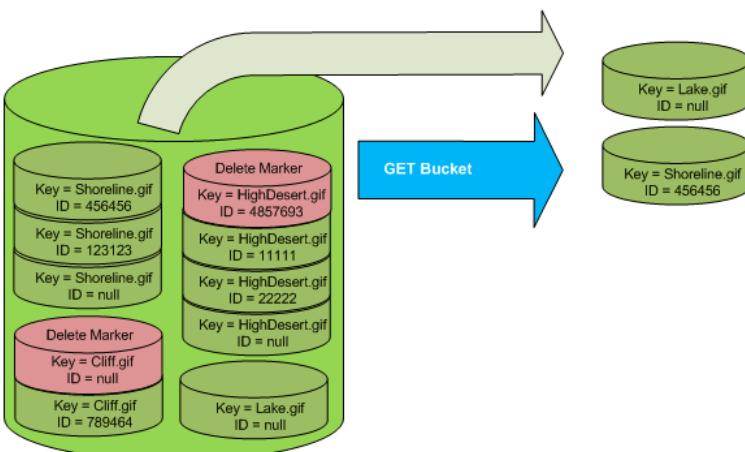
- Um erro 404 (objeto não encontrado)
- Um cabeçalho de resposta, `x-amz-delete-marker: true`

O cabeçalho de resposta mostra que o objeto acessado era um marcador de exclusão. Esse cabeçalho de resposta nunca retorna `false`; se o valor for `false`, o Amazon S3 não incluirá esse cabeçalho na resposta.

A figura a seguir mostra como um `GET` simples em um objeto, cuja versão atual é um marcador de exclusão, retorna um erro 404 Nenhum objeto encontrado.



O único modo de listar marcadores de exclusão (e outras versões de um objeto) é usando o sub-recurso `versions` em uma solicitação `GET Bucket versions`. Um `GET` simples não recupera objetos com marcadores de exclusão. A figura a seguir mostra como uma solicitação `GET Bucket` simples não retorna objetos cuja versão atual é um marcador de exclusão.

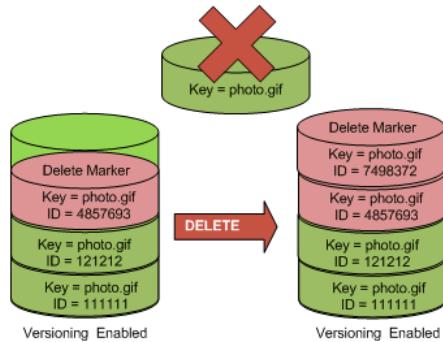


## Remoção de marcadores de exclusão

Para excluir um marcador de exclusão, você deve especificar seu ID de versão em uma solicitação `DELETE Object versionId`. Se você usar uma solicitação `DELETE` para excluir um marcador de

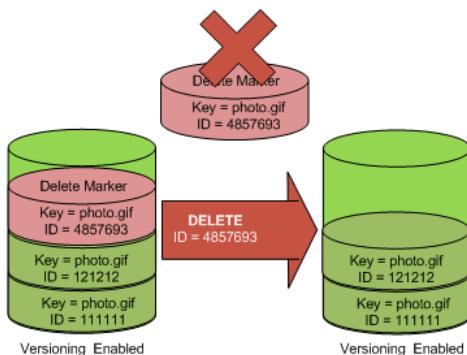
exclusão (sem especificar o ID da versão do marcador de exclusão), o Amazon S3 não excluirá o marcador de exclusão, mas, em vez disso, inserirá outro marcador de exclusão.

A figura a seguir mostra como um **DELETE** simples em um marcador de exclusão não remove nada, mas adiciona um novo marcador de exclusão a um bucket.



Em um bucket com versionamento ativado, esse novo marcador de exclusão deve ter um ID de versão único. Assim, é possível ter vários marcadores de exclusão do mesmo objeto em um bucket.

Para excluir permanentemente um marcador de exclusão, você deve incluir seu ID de versão em uma solicitação **DELETE Object versionId**. A figura a seguir mostra como uma solicitação **DELETE Object versionId** simples remove, permanentemente, um marcador de exclusão. Apenas o proprietário de um bucket pode remover permanentemente um marcador de exclusão.



O efeito da remoção do marcador de exclusão é que uma simples solicitação **GET** não irá recuperar a versão atual (121212) do objeto.

Para remover permanentemente um marcador de exclusão

1. Defina **versionId** como o ID da versão do marcador de exclusão que você deseja remover.
2. Envie uma solicitação **DELETE Object versionId**.

#### Example Remoção de um marcador de exclusão

O exemplo a seguir remove o marcador de exclusão de `photo.gif` com versão 4857693.

```
DELETE /photo.gif?versionId=4857693 HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Quando você exclui um marcador de exclusão, o Amazon S3 inclui na resposta:

```
204 NoContent
x-amz-version-id: versionID
x-amz-delete-marker: true
```

## Transição de versões de objeto

Você pode definir as regras de configuração de ciclo de vida de objetos que têm ciclo de vida bem definido para fazer a transição de versões de objeto para a classe de armazenamento GLACIER em um momento específico no ciclo de vida do objeto. Para obter mais informações, consulte [Gerenciamento do ciclo de vida de objetos \(p. 122\)](#).

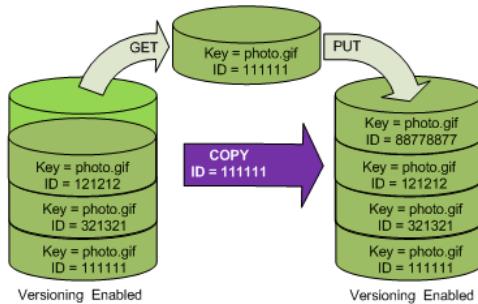
## Restauração de versões anteriores

Uma das proposições de valor do versionamento é a habilidade de recuperar versões anteriores de um objeto. Existem duas abordagens para se fazer isso:

- Copiar uma versão anterior do objeto para o mesmo bucket
  - O objeto copiado torna-se a versão atual desse objeto e todas as versões são preservadas.
- Excluir, permanentemente, a versão atual do objeto

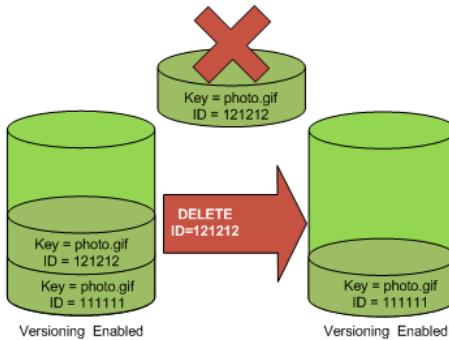
Quando você exclui a versão atual do objeto, torna a versão anterior a versão atual desse objeto.

Como todas as versões do objeto são preservadas, você pode fazer de qualquer versão anterior a versão atual copiando uma versão específica do objeto para o mesmo bucket. Na figura a seguir, o objeto de origem (ID = 111111) é copiado no mesmo bucket. O Amazon S3 fornece um novo ID (88778877) e torna-se a versão atual do objeto. Assim, o bucket tem tanto a versão original (111111) quanto a cópia (88778877) do objeto.



Um GET subsequente recuperará a versão 88778877.

A figura a seguir mostra como excluir a versão atual (121212) de um objeto, o que deixa a versão anterior (111111) como a atual do objeto.



Um GET subsequente recuperará a versão 111111.

## Permissões de objeto com versões

As permissões são definidas no nível da versão. Cada versão tem seu próprio proprietário de objeto; uma conta da AWS que cria a versão do objeto é o proprietário. Assim, você pode definir diferentes permissões para diferentes versões do mesmo objeto. Para fazer isso, você deve especificar o ID da versão do objeto cujas permissões deseja definir em uma solicitação PUT Object `versionId acl`. Para uma descrição detalhada e instruções de uso de ACLs, consulte [Gerenciamento de permissões de acesso aos recursos do Amazon S3 \(p. 282\)](#).

Example Definição de permissões para uma versão de objeto

A solicitação a seguir define a permissão do beneficiário, `BucketOwner@amazon.com`, como `FULL_CONTROL` na chave, `my-image.jpg`, ID de versão, `3HL4kqtJvjVBH40Nrjfk`.

```
PUT /my-image.jpg?acl&versionId=3HL4kqtJvjVBH40Nrjfk HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
Content-Length: 124

<AccessControlPolicy>
  <Owner>
    <ID>75cc57f09aa0c8caeab4f8c24e99d10f8e7faeefbf76c078efc7c6caea54ba06a</ID>
    <DisplayName>mtd@amazon.com</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
        <ID>a9a7b886d6fd24a52fe8ca5bef65f89a64e0193f23000e241bf9b1c61be666e9</ID>
        <DisplayName>BucketOwner@amazon.com</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

Da mesma forma, para obter permissões para uma versão específica do objeto, você deve especificar seu ID de versão em uma solicitação GET Object `versionId acl`. Você precisa incluir o ID da versão porque, por padrão, o GET Object `acl` retorna as permissões da versão atual do objeto.

Example Recuperação das permissões para uma versão especificada de objeto

No exemplo a seguir, o Amazon S3 retorna as permissões da chave, `my-image.jpg`, ID da versão, `DVBH40Nr8X8gUMLUo`.

```
GET /my-image.jpg?versionId=DVBH40Nr8X8gUMLUo&acl HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU
```

Para obter mais informações, consulte [Objeto GET acl](#).

## Gerenciamento de objetos em um bucket com versionamento suspenso

Tópicos

- [Adição de objetos a buckets com versionamento suspenso \(p. 468\)](#)
- [Recuperação de objetos de buckets com versionamento suspenso \(p. 469\)](#)
- [Exclusão de objetos de buckets com versionamento suspenso \(p. 469\)](#)

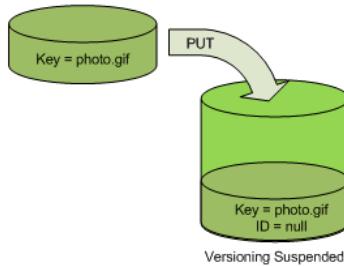
Você suspende o versionamento para parar de ter novas versões do mesmo objeto em um bucket. Você pode querer fazer isso porque deseja ter apenas uma única versão de um objeto no bucket, ou você pode querer não incorrer em cobrança por múltiplas versões.

Quando você suspende o versionamento, os objetos existentes em seu bucket não são alterados. O que muda é como o Amazon S3 trata os objetos em solicitações futuras. Os tópicos nesta seção explicam várias operações de objeto em um bucket com versionamento suspenso.

## Adição de objetos a buckets com versionamento suspenso

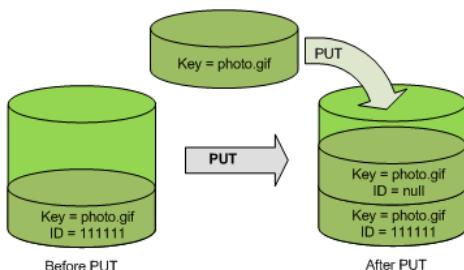
Assim que você suspende versionamento em um bucket, o Amazon S3 adiciona automaticamente um ID de versão `null` a cada objeto subsequente armazenado a partir dali (usando `PUT`, `POST` ou `COPY`) nesse bucket.

A figura a seguir mostra como o Amazon S3 adiciona um ID de versão `null` a cada objeto quando este é adicionado a um bucket com versionamento suspenso.

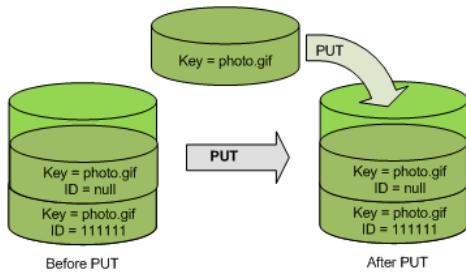


Se um versão nula já existir no bucket e você adicionar outro objeto com a mesma chave, o objeto adicionado substituirá a versão original nula.

Se existirem objetos com versões no bucket, a versão que você usa no `PUT` torna-se a versão atual do objeto. A figura a seguir mostra como a adição de um objeto a um bucket que contém objetos com versões não substitui o objeto já existente no bucket. Nesse caso, a versão 111111 já estava no bucket. O Amazon S3 anexa um ID de versão de nulo ao objeto que está sendo adicionado e o armazena no bucket. A versão 111111 não é substituída.



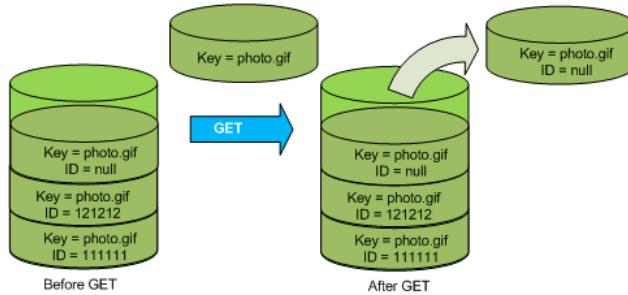
Se uma versão nula já existir em um bucket, a versão nula será substituída, como mostrado na figura a seguir.



Embora a chave e o ID (null) da versão nula sejam iguais antes e após o PUT, o conteúdo da versão nula originalmente armazenada no bucket é substituído pelo conteúdo do objeto PUT no bucket.

## Recuperação de objetos de buckets com versionamento suspenso

Uma solicitação GET Object retorna a versão atual de um objeto sempre independentemente de você ter ou não ativado o versionamento de um bucket. A figura a seguir mostra como um GET simples retorna a versão atual de um objeto.



## Exclusão de objetos de buckets com versionamento suspenso

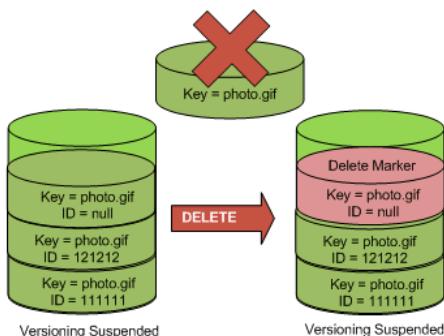
Se o versionamento estiver suspenso, uma solicitação DELETE:

- Pode remover apenas um objeto cujo ID de versão seja null

Não removerá nada se não existir uma versão nula do objeto no bucket.

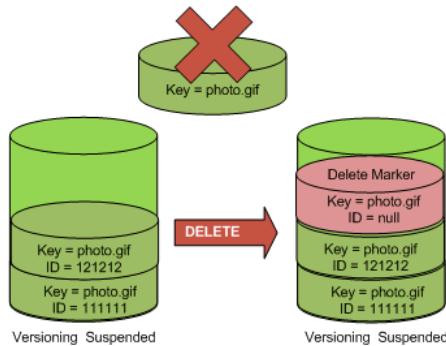
- Insere um marcador de exclusão no bucket.

A figura a seguir mostra como um DELETE simples remove uma versão nula e o Amazon S3 insere um marcador de exclusão em seu lugar com o ID de versão null.

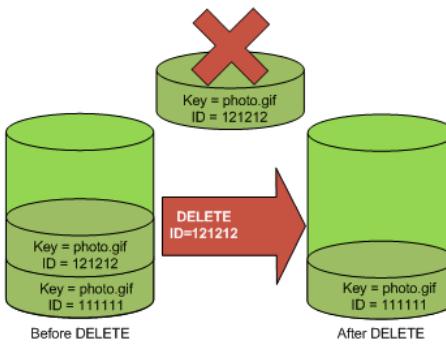


Lembre-se de que um marcador de exclusão não tem conteúdo, de modo que você perde o conteúdo da versão nula quando um marcador de exclusão a substitui.

A figura a seguir mostra um bucket que não tem uma versão nula. Nesse caso, o **DELETE** não remove nada; o Amazon S3 apenas insere o marcador de exclusão.



Mesmo em um bucket com versionamento suspenso, o proprietário do bucket pode excluir permanentemente uma versão especificada. A figura a seguir mostra que excluir uma versão do objeto especificada remove permanentemente esse objeto. Apenas o proprietário do bucket pode excluir uma versão de objeto especificada.



## Introdução ao Amazon S3 Object Lock

O Amazon S3 Object Lock permite que você armazene objetos usando um modelo "Write Once Read Many" (WORM). Usando o S3 Object Lock, você pode evitar que um objeto seja excluído ou substituído por um tempo fixo ou indefinido. O S3 Object Lock permite atender aos requisitos regulatórios que exigem armazenamento WORM ou simplesmente adicionar uma camada adicional de proteção em relação a alterações e exclusões de objetos. O Amazon S3 Object Lock foi avaliado por Cohasset Associates para ser usado em ambientes sujeitos a regulamentações SEC 17a-4, CTCC e FINRA. Para obter mais informações sobre como o S3 Object Lock se relaciona a essas regulamentações, consulte o [Cohasset Associates Compliance Assessment](#).

O S3 Object Lock fornece duas maneiras de gerenciar a retenção de objetos: períodos de retenção e retenções legais. Um período de retenção especifica um período fixo durante o qual um objeto permanece bloqueado. Durante esse período, o objeto será protegido por WORM e não poderá ser substituído nem excluído.

Uma retenção legal oferece a mesma proteção de um período de retenção, mas sem data de expiração. Em vez disso, uma retenção legal permanecerá em vigor até você removê-la explicitamente. As retenções legais independem de períodos de retenção: uma versão do objeto pode ter um período de retenção e uma retenção legal, um, mas não o outro ou nenhum dos dois.

O S3 Object Lock só funciona em buckets com versão, e os períodos de retenção e as retenções legais se aplicam a versões de objeto individuais. Quando você bloqueia uma versão do objeto, o Amazon S3 armazena as informações do bloqueio nos metadados dessa versão do objeto. A colocação de um período de retenção ou uma retenção legal em um objeto só protege a versão especificada na solicitação e não evita a criação de novas versões do objeto. Se você colocar um objeto em um bucket que tenha o mesmo nome da chave de um objeto existente protegido, o Amazon S3 criará uma nova versão desse objeto, a armazenará no bucket conforme solicitado e relatará a solicitação como concluída com êxito. A versão existente, protegida, do objeto permanece bloqueada de acordo com a configuração da retenção.

Para usar Amazon S3 Object Lock, você segue estas etapas:

1. Crie um novo bucket com S3 Object Lock habilitado.
2. (Opcional) Configure um período de retenção padrão para objetos colocados no bucket.
3. Coloque os objetos que você deseja bloquear no bucket.
4. Aplique um período de retenção, uma retenção legal, ou ambos, aos objetos que você deseja proteger.

Os tópicos a seguir descrevem como usar Amazon S3 Object Lock.

#### Tópicos

- [Visão geral do Amazon S3 Object Lock \(p. 471\)](#)
- [Gerenciar bloqueios de objeto \(p. 474\)](#)

## Visão geral do Amazon S3 Object Lock

### Modos de retenção

O Amazon S3 Object Lock fornece dois modos de retenção: governança e conformidade. Esses modos de retenção aplicam níveis diferentes de proteção aos objetos. Aplique um dos modos de retenção a qualquer versão do objeto protegida por S3 Object Lock.

No modo de governança, os usuários não podem substituir nem excluir uma versão do objeto ou alterar as configurações de bloqueio, a menos que tenham permissões especiais. O modo de governança permite proteger objetos contra a exclusão da maioria dos usuários, ao mesmo tempo em que permite conceder a alguns usuários permissão para alterar as configurações de retenção ou excluir o objeto, caso necessário. Também é possível usar o modo de governança para testar as configurações do período de retenção antes de criar um período de retenção do modo de conformidade. Para substituir ou remover as configurações de retenção do modo de governança, um usuário deve ter a permissão `s3:BypassGovernanceMode` e incluir explicitamente `x-amz-bypass-governance-retention:true` como um cabeçalho de solicitação com qualquer solicitação que exija a substituição do modo de governança.

No modo de conformidade, uma versão do objeto protegida não pode ser substituída nem excluída por qualquer usuário, inclusive o usuário raiz na conta da AWS. Depois que um objeto estiver bloqueado no modo de conformidade, o modo de retenção não poderá ser alterado nem o período de retenção poderá ser encurtado. O modo de conformidade garante que uma versão do objeto protegida não possa ser substituída nem excluída durante o período de retenção.

#### Note

A atualização dos metadados de uma versão do objeto, ocorrida quando você faz ou altera um bloqueio de objeto, não substitui a versão do objeto nem redefine o time stamp `Last-Modified`.

## Períodos de retenção

Um período de retenção protege uma versão do objeto por um período fixo. Quando você coloca um período de retenção em uma versão do objeto, o Amazon S3 armazena um time stamp nos metadados da versão do objeto para indicar quando o período de retenção expira. Depois que o período de retenção expirar, a versão do objeto não poderá ser substituída nem excluída, a menos que você tenha feito uma retenção legal na versão do objeto.

Um período de retenção pode ser colocado em uma versão do objeto explicitamente ou por meio de uma configuração padrão do bucket. Ao aplicar um período de retenção a uma versão do objeto explicitamente, você especifica uma retenção até uma determinada data para a versão do objeto. O Amazon S3 armazenará a data nos metadados da versão do objeto e protegerá a versão do objeto até o período de retenção.

Ao usar as configurações padrão do bucket, você não especifica uma retenção até uma determinada data. Em vez disso, especifique uma duração, em dias ou anos, pela qual a versão do objeto colocada no bucket deve ser protegida. Quando você coloca um objeto no bucket, o Amazon S3 calcula uma retenção até uma determinada data para a versão do objeto adicionando a duração especificada para o time stamp da criação da versão do objeto e armazena a data nos metadados da versão do objeto. A versão do objeto acaba sendo protegida exatamente, ainda que você tenha colocado explicitamente um bloqueio com esse período de retenção na versão do objeto.

### Note

Caso a solicitação para colocar uma versão do objeto em um bucket contenha um modo de retenção explícito e um período, essas configurações substituem todas as padrão do bucket dessa versão do objeto.

Assim como acontece com todas as outras configurações do S3 Object Lock, os períodos de retenção se aplicam a versões de objeto individuais. As versões diferentes de um único objeto podem ter modos e períodos de retenção diferentes.

Por exemplo, se você tiver um objeto por 15 dias em um período de retenção de 30 dias e usar PUT em um objeto no S3 com o mesmo nome e um período de retenção de 60 dias, PUT será bem-sucedido e o Amazon S3 criará uma nova versão do objeto com um período de retenção de 60 dias. A versão anterior mantém o período de retenção original e se torna exclusiva em 15 dias.

Prolongue um período de retenção depois que você tiver aplicado uma configuração de retenção a uma versão do objeto. Para isso, envie uma nova solicitação de bloqueio para a versão do objeto com uma retenção até uma data específica posterior a uma configurada atualmente para a versão do objeto. O Amazon S3 substitui o período de retenção existente pelo período novo, mais longo. Qualquer usuário com permissões para colocar um período de retenção do objeto pode prolongar um período de retenção para uma versão do objeto bloqueada em qualquer modo.

## Retenções legais

O S3 Object Lock também permite colocar uma retenção legal em uma versão do objeto. Assim como um período de retenção, uma retenção legal evita que uma versão do objeto seja substituída ou excluída. Porém, uma retenção legal não tem um período de retenção associado e permanecerá em vigor até ser removida. As retenções legais podem ser feitas e removidas livremente por qualquer usuário com a permissão `s3:PutObjectLegalHold`.

As retenções legais independentes dos períodos de retenção. Desde que o bucket que contém o objeto tenha S3 Object Lock habilitado, é possível fazer e remover retenções legais, independentemente da versão do objeto especificada ter ou não um período de retenção definido. Fazer uma retenção legal em uma versão do objeto não afeta o modo de retenção ou o período de retenção dessa versão do objeto. Por exemplo, se você fizer uma retenção legal em uma versão do objeto ainda com a versão do objeto também protegida por um período de retenção e o período de retenção expirar, o objeto não perderá a

proteção WORM. Em vez disso, a retenção legal continuará protegendo o objeto até um usuário autorizado removê-la explicitamente. Da mesma maneira, se você remover uma retenção legal enquanto uma versão do objeto tiver um período de retenção em vigor, a versão do objeto continuará protegida até o período de retenção expirar.

## Configuração do bucket

Para usar o S3 Object Lock, você primeiro habilita o bloqueio do objeto para um bucket. Também é possível configurar um modo e um período de retenção padrão que se aplicarão a novos objetos colocados no bucket.

### Habilitar bloqueio do objeto

Para bloquear todos os objetos, você precisa configurar um bucket para usar Amazon S3 Object Lock. Para configurar um bucket para o S3 Object Lock, você especifica quando cria o bucket que deseja habilitar S3 Object Lock. Depois de configurar um bucket para S3 Object Lock, você poderá bloquear objetos nesse bucket com períodos de retenção, retenções legais ou ambos.

#### Note

- Só é possível habilitar S3 Object Lock para novos buckets. Se você precisar ativar S3 Object Lock para um bucket existente, entre em contato com o AWS Support.
- Ao criar um bucket com S3 Object Lock habilitado, o Amazon S3 habilita automaticamente o versionamento para o bucket.
- Assim que criar um bucket com S3 Object Lock habilitado, você não poderá desabilitar o Object Lock nem suspender o versionamento para o bucket.

### Configurações de retenção padrão

A habilitação do S3 Object Lock para um bucket permite que o bucket armazene objetos protegidos, mas não protege automaticamente objetos colocados no bucket. Se quiser proteger automaticamente versões de objeto colocadas no bucket, você poderá configurar um período de retenção padrão. As configurações padrão se aplicam a todos os novos objetos no bucket, a menos que você especifique explicitamente um modo de retenção e um período diferente para um objeto ao criá-lo.

#### Tip

Se quiser impor o modo de retenção padrão do bucket e um período para todas as novas versões do objeto colocadas em um bucket, você poderá definir os padrões do bucket e negar aos usuários a permissão para fazer configurações de retenção do objeto. O Amazon S3 acaba aplicando o modo de retenção padrão e o período a novas versões do objeto colocadas no bucket e rejeita todas as solicitações feitas para colocar um objeto que inclua um modo de retenção e uma configuração.

As configurações padrão do bucket exigem um modo e um período. Um modo padrão do bucket é de governança ou de conformidade, conforme descrito em [Modos de retenção \(p. 471\)](#). Um período de retenção padrão é descrito não como um time stamp, mas como um período em dias ou em anos. Quando você coloca uma versão do objeto em um bucket com um período de retenção, o S3 Object Lock calcula uma retenção até uma data específica adicionando o período de retenção padrão ao carimbo de data e hora de criação da versão do objeto. O Amazon S3 armazena o time stamp resultante como a retenção até uma data específica da versão, ainda que você tenha calculado o carimbo de data e hora manualmente ou colocado na versão do objeto por conta própria.

As configurações padrão só se aplicam a novos objetos colocados no bucket. A colocação de uma configuração de retenção padrão em um bucket não coloca configurações de retenção em objetos já existentes no bucket.

### Important

Os bloqueios de objeto se aplicam apenas a versões de objeto individuais. Caso você coloque um objeto em um bucket que tenha um período de retenção padrão e não especifique explicitamente um período de retenção para esse objeto, o Amazon S3 cria o objeto com um período de retenção correspondente ao padrão do bucket. Depois que o objeto for criado, o período de retenção será independente do período de retenção padrão do bucket. A alteração do período de retenção padrão de um bucket não vai alterar o período de retenção existente para nenhum objeto nesse bucket.

## Permissões obrigatórias

As operações do S3 Object Lock exigem as permissões listadas na tabela a seguir.

### Permissões de bloqueio do objeto do S3

Operação	Permissões obrigatórias
Criar ou modificar o modo e o período de retenção de uma versão do objeto	s3:PutObjectRetention
Criar ou modificar uma retenção legal para uma versão do objeto	s3:PutObjectLegalHold
Obter o modo e o período de retenção de uma versão do objeto	s3:GetObjectRetention
Obter o status da retenção legal de uma versão do objeto	s3:GetObjectLegalHold
Ignorar modo de retenção de governança	s3:BypassGovernanceRetention
Obter configuração do Object Lock de um bucket	s3:GetBucketObjectLockConfiguration
Criar ou modificar uma configuração do Object Lock de um bucket	s3:PutBucketObjectLockConfiguration

## Restrições e limitações

Não é possível copiar de um bucket que tenha o S3 Object Lock habilitado usando Cross-Region Replication (CRR – Replicação entre regiões). Se você tentar configurar uma regra CRR usando um bucket de origem configurado para S3 Object Lock, a solicitação falhará. Porém, use um bucket com S3 Object Lock habilitado como o destino de uma regra CRR. Isso permite aplicar a proteção WORM aos objetos replicados. Para obter mais informações sobre CRR, consulte [Replicação entre regiões \(p. 544\)](#).

## Recursos relacionados

- [Introdução ao Amazon S3 Object Lock \(p. 470\)](#)
- [Gerenciar bloqueios de objeto \(p. 474\)](#)

## Gerenciar bloqueios de objeto

### Exibir informações do bloqueio

Você pode exibir o status Object Lock da versão de um objeto usando os comandos GET Object ou HEAD Object. Ambos os comandos retornam o modo de retenção, de manutenção até uma data específica, e o

status de retenção legal para a versão do objeto especificada. Para exibir o modo e o período de retenção de uma versão do objeto, você deve ter a permissão `s3:GetObjectRetention`. Para exibir o status de retenção legal de uma versão do objeto, você deve ter a permissão `s3:GetObjectLegalHold`. Se você usar GET ou HEAD em uma versão do objeto, mas não tiver as permissões necessárias para exibir o status de bloqueio, a solicitação será bem-sucedida, mas não retornará as informações sem permissão para exibir.

Exiba a configuração de retenção padrão Object Lock do bucket, caso haja uma, solicitando a configuração Object Lock do bucket. Você deve ter a permissão `s3:GetBucketObjectLockConfiguration` para exibir a configuração de um bucket. Caso você faça uma solicitação para uma configuração de bloqueio do objeto em um bucket que não tenha Object Lock habilitado, o Amazon S3 retorna um erro.

O S3 Inventory Reports pode ser configurado nos buckets para incluir Retain Until Date, Object Lock Mode e Legal Hold Status de todos os objetos em um bucket. Para obter mais informações sobre o S3 Inventory Reports, consulte [Inventário do Amazon S3 \(p. 273\)](#).

## Ignorar modo de governança

Realize operações em versões de objeto bloqueadas em modo de governança, mesmo desprotegidas caso você tenha a permissão `s3:BypassGovernanceRetention`. Isso inclui a exclusão de uma versão do objeto, a redução do período de retenção ou a remoção do bloqueio de objeto com a colocação de um novo bloqueio com parâmetros vazios. Para ignorar o modo de governança, você deve indicar explicitamente na solicitação que você deseja ignorar o modo de governança. Você pode fazer isso incluindo o cabeçalho `x-amz-bypass-governance-retention:true` com a sua solicitação ou usando o parâmetro equivalente com solicitações feitas por meio da AWS CLI ou dos AWS SDKs. O Console de gerenciamento da AWS aplica automaticamente esse cabeçalho para solicitações feitas por meio do console se você tiver a permissão necessária para ignorar o modo de governança.

### Note

Ignorar o modo de governança não afeta o status de retenção legal de uma versão do objeto. Caso uma versão do objeto tenha uma retenção legal habilitada, esta permanece em vigência e evita que solicitações substituam ou excluam a versão do objeto.

## Eventos e notificações

O S3 Events pode ser configurado para operações no nível do objeto em um bucket do Object Lock. Quando as chamadas `PUT Object`, `HEAD Object` e `GET Object` incluírem metadados do Object Lock, os eventos dessas chamadas incluirão esses valores de metadados. Quando os metadados do Object Lock forem adicionados a ou atualizados para um objeto, essas ações também vão disparar eventos. Esses eventos ocorrem sempre que você usa `PUT` ou `GET` na retenção do objeto ou informações de retenção legal.

Use o Amazon S3 Event Notifications para rastrear o acesso e as alterações feitas nas configurações do Object Lock e nos dados que usam o AWS CloudTrail. Use também o Amazon CloudWatch para gerar alertas com base nesses dados. Para obter mais informações sobre o S3 Events, consulte [Configurar notificações de evento do Amazon S3 \(p. 522\)](#). Para obter mais informações sobre o AWS CloudTrail consulte a [Documentação do AWS CloudTrail](#). Para obter mais informações sobre o Amazon CloudWatch consulte a [Documentação do AWS CloudWatch](#).

## Definir limites de retenção

Defina os períodos de retenção mínimo e máximo permitidos para um bucket usando uma política de bucket. Faça isso usando a chave de condição `s3:object-lock-remaining-retention-days`. O seguinte exemplo mostra uma política de bucket que define um período de retenção mínimo de 10 dias:

```
{"Version": "2012-10-17",
```

```
"Id": "<Policy1436912751980>",
"Statement": [
    "Sid": "<Stmt1436912698057>",
    "Effect": "Deny",
    "Principal": "*",
    "Action": ["s3:PutObjectRetention"],
    "Resource": "arn:aws:s3:::<example-bucket>/*",
    "Condition": {"NumericGreaterThan": {"s3:object-lock-remaining-retention-days": "10"}}
]}
```

Para obter mais informações sobre como usar políticas de bucket, consulte [Uso de políticas de bucket e políticas de usuário \(p. 326\)](#).

## Marcadores de exclusão e gerenciamento do ciclo de vida

Embora possa excluir uma versão do objeto protegida, você ainda pode criar um marcador de exclusão para esse objeto. A colocação de um marcador de exclusão em um objeto não exclui versão do objeto, mas faz o Amazon S3 se comportar como se o objeto tivesse sido excluído. Para obter mais informações sobre marcadores de exclusão, consulte [Trabalho com marcadores de exclusão \(p. 463\)](#).

### Note

Os marcadores de exclusão não são protegidos por WORM, independentemente de qualquer período de retenção ou da retenção legal no objeto subjacente.

As configurações do Object Lifecycle Management continuam funcionando normalmente em objetos protegidos, inclusive a colocação de marcadores de exclusão. No entanto, as versões de objeto protegidas continuam seguras, evitando a exclusão ou a substituição por uma configuração do ciclo de vida. Para obter mais informações sobre como gerenciar ciclos de vida de objeto, consulte [Gerenciamento do ciclo de vida de objetos \(p. 122\)](#).

## Recursos relacionados

- [Introdução ao Amazon S3 Object Lock \(p. 470\)](#)
- [Visão geral do Amazon S3 Object Lock \(p. 471\)](#)

# Realizar operações em lote

## Cadastre-se na demonstração

É possível usar o Operações em lote do Amazon S3 para realizar operações em lote de larga escala nos objetos do Amazon S3. O Operações em lote do Amazon S3 pode executar uma única operação em lista de objetos do Amazon S3 que você especificar. Um único trabalho pode realizar a operação especificada em bilhões de objetos contendo exabytes de dados. O Amazon S3 monitora o progresso, envia notificações e armazena um relatório detalhado de conclusão de todas as ações, fornecendo uma experiência totalmente gerenciada, auditável e sem servidor. Use Operações em lote do Amazon S3 por meio do Console de gerenciamento da AWS, da AWS CLI, dos SDKs da AWS ou da API REST.

Use o Operações em lote do Amazon S3 para copiar objetos e definir tags de objetos para listas de controle de acesso (ACLs). Também é possível iniciar restaurações de objetos do Amazon S3 Glacier ou invocar uma função do AWS Lambda para realizar ações personalizadas usando seus objetos. Realize essas operações em uma lista personalizada de objetos ou use um relatório de inventário do Amazon S3 para simplificar ainda mais a geração das maiores listas de objetos. O Operações em lote do Amazon S3 usa as mesmas APIs que você usa com o Amazon S3; portanto, a interface parecerá familiar para você.

## Tópicos

- [Terminologia \(p. 477\)](#)
- [Os elementos básicos: trabalhos de operação em lote do Amazon S3 \(p. 478\)](#)
- [Criar um trabalho de operação em lote \(p. 479\)](#)
- [Operações \(p. 483\)](#)
- [Gerenciar operações em lote \(p. 489\)](#)

## Terminologia

Esta seção usa os termos trabalhos, operações e tarefas. Confira suas definições abaixo:

### Trabalho

Uma tarefa é a unidade básica de trabalho para Operações em lote do Amazon S3. Uma tarefa contém todas as informações necessárias para executar a operação especificada nos objetos listados no manifesto. Depois que você tiver fornecido essas informações e solicitado o início do trabalho, ele executará a operação em cada objeto no manifesto.

### Operação

Uma operação é um comando único que você deseja que uma tarefa execute. Cada tarefa contém somente um tipo de operação com um conjunto de parâmetros, que o Operações em lote do Amazon S3 executa em cada objeto.

### Tarefa

Uma tarefa é a unidade de execução para um trabalho. Uma tarefa representa uma única chamada para uma operação de API do Amazon S3 ou AWS Lambda a fim de realizar a operação do trabalho em um único objeto. Ao longo da vida útil de uma tarefa, o Operações em lote do Amazon S3 criará uma tarefa para cada objeto especificado no manifesto.

# Os elementos básicos: trabalhos de operação em lote do Amazon S3

[Cadastre-se na demonstração](#)

Para criar um trabalho, você dá a Operações em lote do Amazon S3 uma lista de objetos e seleciona a ação a ser realizada neles. O Operações em lote do Amazon S3 dá suporte às seguintes operações:

- [PUT copy object \(Copiar objeto PUT\)](#)
- [PUT object tagging \(Marcação de objeto PUT\)](#)
- [PUT object ACL \(ACL de objeto PUT\)](#)
- [Initiate Glacier restore \(Iniciar restauração do Glacier\)](#)

Os objetos em que você deseja que uma tarefa aja estão listados em um objeto de manifesto. Um trabalho realiza a operação especificada em cada objeto incluído no manifesto. Use um relatório [Inventário do Amazon S3 \(p. 273\)](#) como um manifesto, o que facilita a criação de listas grandes de objetos localizadas em um bucket. Também é possível especificar um manifesto em um formato CSV simples que permite realizar operações em lotem em uma lista personalizada de objetos contidos em um único bucket.

Depois que você tiver criado um trabalho, o Amazon S3 processará a lista de objetos no manifesto e executará a operação especificada em cada objeto. Enquanto uma tarefa está em execução, monitore o andamento de maneira programática ou por meio do console do Amazon S3. Também é possível configurar uma tarefa para gerar um relatório de conclusão quando ele termina. O relatório de conclusão descreve os resultados de cada tarefa executada pelo trabalho. Para obter mais informações sobre como monitorar tarefas, consulte [Gerenciar operações em lote \(p. 489\)](#).

## Como especificar um manifesto

Manifesto é um objeto do Amazon S3 que lista as chaves de objeto sobre o qual você deseja que o Amazon S3 aja. Para especificar um manifesto para um trabalho, especifique a chave de objeto, a ETag e o ID da versão opcional do manifesto. É possível especificar um manifesto na solicitação de criação de um trabalho usando um dos seguintes formatos:

- Relatório de inventário do Amazon S3 — Deve ser exibido em formato CSV do Amazon S3. É necessário especificar o arquivo `manifest.json` associado ao relatório de inventário. Para obter mais informações sobre relatórios de inventário, consulte [Inventário do Amazon S3 \(p. 273\)](#). Se o relatório de inventário incluir IDs de versões, o Operações em lote do Amazon S3 operará nas versões especificadas do objeto.
- Arquivo CSV — Cada linha no arquivo deve incluir o nome do bucket, a chave do objeto e, opcionalmente, a versão do objeto. É possível especificar IDs de versão para todos os objetos ou pular essa etapa. Para obter mais informações sobre o formato de manifesto CSV, consulte [JobManifestSpec](#) em Amazon Simple Storage Service API Reference.

Veja um exemplo a seguir:

```
Examplebucket,objectkey1,PZ9ibn9D5lP6p298B7S9_ceqx1n5EJ0p
Examplebucket,objectkey2,PZ9ibn9D5lP6p298B7S9_ceqx1n5EJ0p
Examplebucket,objectkey3,jbo9_jhdPEyB4RrmOxWS0kU0EoNrU_oI
```

### Important

Se os objetos no manifesto estiverem em um bucket com versão, você deverá especificar os IDs de versão dos objetos. Ao criar um trabalho, o Operações em lote do Amazon S3 analisa todo o manifesto antes de executá-lo. No entanto, ele não tira snapshots do estado do bucket. Como os manifestos podem conter bilhões de objetos, as tarefas podem demorar muito para serem executadas. Se você substituir um objeto por uma nova versão enquanto uma tarefa estiver em execução e não especificar o ID da versão desse objeto, o Amazon S3 realizará a operação na versão mais recente, e não na versão que existia quando você criou a tarefa. A única maneira de evitar esse comportamento é especificar o ID de versão do objeto listado no manifesto.

## Criar um trabalho de operação em lote

### Cadastre-se na demonstração

Esta seção dá as informações necessárias para criar um trabalho do Operações em lote do Amazon S3. Ela também descreve os resultados de uma solicitação `Create Job`.

### Criar solicitação de tarefa

Para criar uma tarefa, você deve fornecer as seguintes informações:

#### Operação

Especifique a operação que deseja que o Operações em lote do Amazon S3 execute nos objetos no manifesto. Cada tipo de operação aceita parâmetros específicos dessa operação, o que permite realizar as mesmas tarefas como se tivesse realizado a operação uma por uma em cada objeto.

#### Manifesto

O manifesto é uma lista de todos os objetos em que você deseja que o Operações em lote do Amazon S3 execute a ação especificada. Use um relatório [Inventário do Amazon S3 \(p. 273\)](#) como manifesto ou sua própria lista CSV personalizada de objetos.

#### Priority

Use prioridades de trabalho para indicar a prioridade relativa desse trabalho em relação a outros em execução na conta. Um número maior indica uma prioridade mais alta.

As prioridades de trabalho não têm significado intrínseco para o Operações em lote do Amazon S3, de maneira que é possível usá-las para priorizar trabalhos da maneira que você desejar. Por exemplo, talvez você queira atribuir prioridade 1 todos os trabalhos `Initiate Restore Object`, prioridade 2 a todos os trabalhos `PUT Object Copy` e prioridade 3 a todos os trabalhos `Put Object ACL`. As operações em lote priorizam trabalhos em ordem numérica, mas não é garantida rigidez a essa ordem. Por isso, você não deve usar prioridades de tarefa para garantir que nenhuma comece ou termine antes de qualquer outra. Caso precise garantir uma ordem rígida, aguarde a conclusão de uma tarefa para iniciar a próxima.

#### RoleArn

Você deve especificar a função do IAM que executará a tarefa. A função do IAM usada para executar a tarefa deve ter permissões suficientes para realizar a operação especificada no trabalho. Por exemplo, para executar um trabalho `PUT Object Copy`, a função do IAM deve ter permissões `s3:GetObject` para o bucket de origem e permissões `s3:PutObject` para o bucket de destino. A função também precisa de permissões para ler o manifesto e gravar o relatório de conclusão do trabalho. Para obter mais informações sobre as funções do , consulte Funções do . Para obter mais informações sobre permissões do Amazon S3, consulte [Especificação de permissões em uma política \(p. 330\)](#).

#### Relatório

Especifique se você deseja que o Operações em lote do Amazon S3 gere um relatório de conclusão. Caso solicite um relatório de conclusão da tarefa, você deve fornecer os parâmetros para o relatório neste elemento. As informações necessárias incluem o bucket onde você deseja armazenar o relatório, o formato do relatório, se deseja que o relatório inclua os detalhes de todas as tarefas ou apenas tarefas com falha e uma string de prefixo opcional.

#### Descrição (opcional)

Também é possível fornecer uma descrição de até 256 caracteres que possa ajudar a rastrear e monitorar o trabalho. O Amazon S3 inclui essa descrição sempre que retorna informações sobre um trabalho ou exibe detalhes dele no console do Amazon S3. É possível classificar e filtrar os trabalhos com facilidade de acordo com as descrições atribuídas. As descrições não precisam ser exclusivas, de maneira que você possa usar descrições como categorias (por exemplo, "Tarefas de cópia de log semanais") para ajudar a rastrear grupos de tarefas semelhantes.

## Criar resposta da tarefa

Se a solicitação do `Create Job` for bem-sucedida, o Amazon S3 retornará um ID de trabalho. O ID de trabalho é um identificador exclusivo gerado pelo Amazon S3 automaticamente, de maneira que possa identificar as operações em lote e monitorar o status.

Quando você cria um trabalho pela AWS CLI, SDKs do AWS ou API REST, é possível definir operações em lote do Amazon S3 para que o trabalho seja processado automaticamente. Ele é executado assim que fica pronto e não aguarda o processamento de trabalhos de prioridade mais alta. Ao criar um trabalho por meio do Console de gerenciamento da AWS, você deve examinar os detalhes dele e confirmar se deseja executá-lo antes do Operações em lote começar a processá-lo. Depois que você confirma que deseja executar a tarefa, ela avança como se tivesse sido criada por meio de um dos outros métodos.

## Conceder permissões para operações em lote

O Amazon S3 deve ter permissão para executar operações em lote em seu nome. Conceda essas permissões por meio de uma função do IAM. Para obter mais informações, consulte [Funções do IAM](#). Quando você cria uma função do IAM, você anexa as seguintes políticas de confiança e permissões.

### Política de confiança

Anexe a seguinte política de confiança à função do IAM para permitir que o serviço principal do Amazon S3 assuma a função.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "Service": "batchoperations.s3.amazonaws.com"  
            },  
            "Action": "sts:AssumeRole"  
        }  
    ]  
}
```

### Políticas de permissões

Dependendo do tipo de operação, é possível anexar uma das seguintes políticas:

### Note

Independentemente da operação, o Amazon S3 precisa de permissão para ler o objeto do manifesto no seu bucket do S3 e, opcionalmente, gerar um relatório para o bucket. Portanto, todas as políticas a seguir incluem as seguintes permissões:

- PUT copy object (Copiar objeto PUT)

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Action": [  
                "s3:PutObject",  
                "s3:PutObjectAcl",  
                "s3:PutObjectTagging"  
            ],  
            "Effect": "Allow",  
            "Resource": "arn:aws:s3:::{DestinationBucket}/*"  
        },  
        {  
            "Action": [  
                "s3:GetObject"  
            ],  
            "Effect": "Allow",  
            "Resource": "arn:aws:s3:::{SourceBucket}/*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:GetObject",  
                "s3:GetObjectVersion",  
                "s3:GetBucketLocation"  
            ],  
            "Resource": [  
                "arn:aws:s3:::{ManifestBucket}/*"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:PutObject",  
                "s3:GetBucketLocation"  
            ],  
            "Resource": [  
                "arn:aws:s3:::{ReportBucket}/*"  
            ]  
        }  
    ]  
}
```

- PUT object tagging (Marcação de objetos do PUT)

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:PutObjectTagging",  
                "s3:PutObjectVersionTagging"  
            ],  
            "Resource": "arn:aws:s3:::{TargetResource}/*"  
        },  
    ]  
}
```

```
{  
    "Effect": "Allow",  
    "Action": [  
        "s3:GetObject",  
        "s3:GetObjectVersion",  
        "s3:GetBucketLocation"  
    ],  
    "Resource": [  
        "arn:aws:s3:::{ManifestBucket}/*"  
    ]  
},  
{  
    "Effect": "Allow",  
    "Action": [  
        "s3:PutObject",  
        "s3:GetBucketLocation"  
    ],  
    "Resource": [  
        "arn:aws:s3:::{ReportBucket}/*"  
    ]  
}  
]
```

- PUT object ACL (ACL de objetos do PUT)

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:PutObjectAcl",  
                "s3:PutObjectVersionAcl"  
            ],  
            "Resource": "arn:aws:s3:::{TargetResource}/*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:GetObject",  
                "s3:GetObjectVersion",  
                "s3:GetBucketLocation"  
            ],  
            "Resource": [  
                "arn:aws:s3:::{ManifestBucket}/*"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:PutObject",  
                "s3:GetBucketLocation"  
            ],  
            "Resource": [  
                "arn:aws:s3:::{ReportBucket}/*"  
            ]  
        }  
    ]  
}
```

- Initiate Glacier restore (Iniciar restauração do Glacier)

```
{
```

```
"Version": "2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "s3:RestoreObject"
        ],
        "Resource": "arn:aws:s3:::{${TargetResource}}/*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:GetBucketLocation"
        ],
        "Resource": [
            "arn:aws:s3:::{${ManifestBucket}}/*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "s3:PutObject",
            "s3:GetBucketLocation"
        ],
        "Resource": [
            "arn:aws:s3:::{${ReportBucket}}/*"
        ]
    }
]
```

## Recursos relacionados

- Os elementos básicos: trabalhos de operação em lote do Amazon S3 (p. 478)
- Operações (p. 483)
- Gerenciar operações em lote (p. 489)

## Operações

O Operações em lote do Amazon S3 dá suporte a cinco operações diferentes. As seções neste tópico descrevem essas operações.

Siga os seguintes comando da AWS Command Line Interface (AWS CLI) para testar o Operações em lote do Amazon S3.

1. Crie uma função do IAM e atribua permissões. A função dá permissão ao Amazon S3 para adicionar tags de objeto. Você vai criar esse trabalho na próxima etapa.
  - a. Crie uma função do IAM da seguinte maneira:

```
aws iam create-role \
--role-name S3BatchJobRole \
--assume-role-policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::{${ReportBucket}}/*"
            ]
        }
    ]
}'
```

```
        "Effect": "Allow",
        "Principal": {
            "Service": "batchoperations.s3.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
    }
]
}'
```

Registre o nome de recurso da Amazon (ARN) da função. Você precisará dele ao criar trabalhos.

- b. Crie uma política do IAM com permissões e anexe-a à função do IAM que você criou na etapa anterior. Para obter mais informações sobre permissões, consulte [Conceder permissões para operações em lote \(p. 480\)](#).

```
aws iam put-role-policy \
--role-name S3BatchJobRole \
--policy-name PutObjectTaggingBatchJobPolicy \
--policy-document '{
"Version": "2012-10-17",
"Statement": [
{
    "Effect": "Allow",
    "Action": [
        "s3:PutObjectTagging",
        "s3:PutObjectVersionTagging"
    ],
    "Resource": "arn:aws:s3:::{${TargetResource}}/*"
},
{
    "Effect": "Allow",
    "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetBucketLocation"
    ],
    "Resource": [
        "arn:aws:s3:::{${ManifestBucket}}/*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "s3:PutObject",
        "s3:GetBucketLocation"
    ],
    "Resource": [
        "arn:aws:s3:::{${ReportBucket}}/*"
    ]
}
]
}'
```

2. Crie um trabalho do S3PutObjectTagging. O arquivo manifest.csv fornece uma lista de buckets e valores de chave de objeto. O trabalho aplica as tags específicas aos objetos identificados no manifesto. O ETag é o ETag do objeto manifest.csv, que você pode obter no console da Amazon S3. A solicitação especifica o parâmetro do no-confirmation-required. Portanto, o Amazon S3 qualifica o trabalho para ser executado sem a necessidade de confirmá-lo usando o comando update-job-status.

```
aws s3control create-job \
--region us-west-2 \
--account-id acct-id \
```

```
--operation '{"S3PutObjectTagging": { "TagSet": [{"Key":"keyOne", "Value":"ValueOne"}] }}' \
--manifest '{"Spec":{"Format":"S3BatchOperations_CSV_20180820","Fields": \
["Bucket","Key"]},"Location":{"ObjectArn":"arn:aws:s3:::my_manifests/ \
manifest.csv","ETag":"60e460c9d1046e73f7dde5043ac3ae85"}}' \
--report '{"Bucket":"arn:aws:s3:::bucket-where-completion-report-goes","Prefix":"final-reports", \
"Format":"Report_CSV_20180820","Enabled":true,"ReportScope":"AllTasks"}' \
--priority 42 \
--role-arn IAM-role \
--client-request-token $(uuidgen) \
--description "job Description" \
--no-confirmation-required
```

Em resposta, o Amazon S3 retorna um ID de trabalho (por exemplo, 00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c). Você vai precisar do ID nos próximos comandos.

3. Obtenha a descrição do trabalho.

```
aws s3control describe-job \
--region us-west-2 \
--account-id acct-id \
--job-id 00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c
```

4. Obtenha a lista de trabalhos Active e Complete.

```
aws s3control list-jobs \
--region us-west-2 \
--account-id acct-id \
--job-statuses '["Active","Complete"]' \
--max-results 20
```

5. Atualize a prioridade do trabalho (quanto mais alto o número, mais alta a prioridade de execução).

```
aws s3control update-job-priority \
--region us-west-2 \
--account-id acct-id \
--priority 98 \
--job-id 00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c
```

6. Se você não tiver especificado um parâmetro --no-confirmation-required no create-job, o trabalho permanecerá em estado suspenso até você confirmar o trabalho, definindo o status como Ready. Então, o Amazon S3 qualificará o trabalho para execução.

```
aws s3control update-job-status \
--region us-west-2 \
--account-id 181572960644 \
--job-id 00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c \
--requested-job-status 'Ready'
```

7. Para cancelar o trabalho, defina o status do trabalho como Cancelled.

```
aws s3control update-job-status \
--region us-west-2 \
--account-id 181572960644 \
--job-id 00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c \
--status-update-reason "No longer needed" \
--requested-job-status Cancelled
```

## Tópicos

- [PUT Object Copy \(Copiar Objeto PUT\) \(p. 486\)](#)
- [Iniciar a restauração de um objeto \(p. 486\)](#)
- [Invocar uma função do Lambda \(p. 487\)](#)
- [Put Object ACL \(ACL de objeto PUT\) \(p. 488\)](#)
- [Put Object Tagging \(Marcação de objeto PUT\) \(p. 488\)](#)

## PUT Object Copy (Copiar Objeto PUT)

A operação PUT Object Copy (Copiar objeto PUT) copia cada objeto especificado no manifesto. É possível copiar objetos no mesmo bucket com novos nomes de chave, para um bucket diferente na mesma região da AWS, ou a um bucket em uma região diferente. O Operações em lote do Amazon S3 oferece suporte à maioria das opções disponíveis para copiar objetos no Amazon S3. Essas opções incluem a definição de metadados de objetos e permissões e a alteração da classe de armazenamento de um objeto. Para obter mais informações sobre a funcionalidade disponível no Amazon S3 para copiar objetos, consulte [Cópia de objetos \(p. 219\)](#).

### Restrições e limitações

- Todos os objetos de origem devem estar em um só bucket.
- Todos os objetos de destino devem estar em um só bucket.
- Você deve ler todas as permissões para o bucket de origem e gravar permissões para o bucket de destino.
- Os objetos a serem copiados devem ter até 5 GB.
- Todas as opções para PUT Object Copy (Copiar Object PUT) são compatíveis, exceto as verificações condicionais de ETags e server-side encryption with customer-provided encryption keys.
- Se os buckets não tiverem versões, você deverá substituir os objetos pelos mesmos nomes de chave.

## Iniciar a restauração de um objeto

A operação do `InitiateRestore` envia uma solicitação de restauração para o Amazon S3 Glacier para cada objeto especificado no manifesto. Para criar um trabalho de restauração de objeto, é necessário incluir dois elementos na solicitação:

- `ExpirationInDays`

Ao restaurar um objeto do S3 Glacier, o objeto restaurante é somente uma cópia temporária, que será excluída pelo Amazon S3 depois de um período predeterminado. Esse elemento especifica quanto tempo a cópia temporária ficará disponível no Amazon S3. Depois que ela expirar, só será possível recuperar o objeto fazendo sua restauração no S3 Glacier novamente. Para obter mais informações sobre a restauração de objetos, consulte [Restaurar objetos arquivados \(p. 259\)](#).

- `GlacierJobTier`

O Amazon S3 pode restaurar objetos do S3 Glacier de acordo com três níveis de recuperação: Expedited (Acelerada), Standard (Padrão) e Bulk (Lote). O Operações em lote do Amazon S3 oferece suporte somente aos níveis Standard e Bulk. Para obter mais informações sobre os níveis de recuperação do S3 Glacier, consulte [Opções de recuperação de arquivos \(p. 259\)](#). Para obter mais informações sobre a definição de preço de cada nível, consulte a seção “Definição de preço de recuperações” em [Amazon S3 GlacierDefinição de preço](#).

#### Important

O trabalho de restauração de objetos só dá início à solicitação. O Operações em lote do Amazon S3 relata o trabalho como concluído para cada objeto depois que a solicitação é iniciada para

cada objeto. O Amazon S3 não atualiza o trabalho, nem notifica você sobre a conclusão da restauração. No entanto, é possível usar notificações de eventos para receber alertas quando os objetos estiverem disponíveis no Amazon S3. Para obter mais informações, consulte [Configurar notificações de evento do Amazon S3 \(p. 522\)](#).

## Restauração de sobreposições

Se o trabalho de restauração tentar restaurar um objeto que já esteja no processo de ser restaurado, o Operações em lote do Amazon S3 agirá da seguinte forma:

A operação será bem-sucedida para o objeto se uma das seguintes condições for verdadeira:

- Quando comparado com a solicitação de restauração já em andamento, o `ExpirationInDays` deste trabalho é o mesmo e `GlacierJobTier` é mais rápido.
- A solicitação anterior já foi concluída e o objeto já está disponível no modo Reduced Redundancy Storage (armazenamento de redundância reduzida). Nesse caso, o Operações em lote do Amazon S3 atualiza a data de vencimento do objeto restaurado para corresponder à `ExpirationInDays` especificada no trabalho.

A operação apresenta falha para o objeto se uma das seguintes condições for verdadeira:

- A solicitação em andamento ainda não foi concluída e a duração da restauração desse trabalho (especificada por `ExpirationInDays`) é diferente da especificada na solicitação de restauração já em andamento.
- O nível de restauração para este trabalho (especificado por `GlacierJobTier`) é o mesmo ou é mais lento do que o nível especificado na solicitação em andamento.

## Limitações

Os trabalhos de restauração de objeto têm as seguintes limitações:

- É necessário criar um trabalho de restauração de objeto na mesma região em que estão os objetos arquivados.
- O Operações em lote do Amazon S3 não é compatível com o S3 Glacier SELECT.
- Operações em lote do Amazon S3 não oferece suporte ao nível de recuperação Expedited (Acelerada).

## Invocar uma função do Lambda

A operação do `LambdaInvoke` executa uma função do AWS Lambda em cada objeto do manifesto. É possível criar funções personalizadas do Lambda para executar as funções existentes fornecidas pelo Modelo de aplicativo sem servidor da AWS. Você pode fornecer argumentos personalizados à função e registrar a saída dela no relatório de conclusão do trabalho. Para obter mais informações sobre como usar o Lambda, consulte [O que é o AWS Lambda?](#) no AWS Lambda Developer Guide.

## Restrições e limitações

- Você não pode usar as funções do Lambda existentes (que não sejam as compatíveis com o Modelo de aplicativo sem servidor da AWS).
- Use a mesma função para todos os objetos.

## Put Object ACL (ACL de objeto PUT)

A operação Put Object ACL (ACL de objeto PUT) substitui as listas de controle de acesso (ACLs) do Amazon S3 para cada objeto listado no manifesto. Com as ACLs, é possível definir quem pode acessar o objeto e quais ações podem ser executadas.

O Operações em lote do Amazon S3 é compatível com ACLs personalizadas e pré-configuradas fornecidas pelo Amazon S3 com um conjunto predefinido de permissões de acesso.

Se os objetos no manifesto estiverem em um bucket com versão, você poderá aplicar as ACLs a versões específicas de cada objeto. Para fazer isso, especifique um ID de versão para cada objeto no manifesto. Se você não incluir o ID de versão para um objeto, o Operações em lote do Amazon S3 aplicará a ACL à versão mais recente dele.

### Note

Se quiser limitar o acesso público a todos os objetos de um bucket, você deverá usar o bloqueio de acesso público do Amazon S3 em vez de Operações em lote do Amazon S3. O bloqueio do acesso público pode limitar o acesso público a cada bucket ou à conta com uma única operação simples, que entra em vigor rapidamente. Essa é uma melhor opção caso seu objetivo seja controlar o acesso público a todos os objetos em um bucket ou uma conta. Use o Operações em lote do Amazon S3 quando precisar aplicar uma ACL personalizada a cada objeto no manifesto. Para obter mais informações sobre bloqueio de acesso público no Amazon S3, consulte [Usar o Amazon S3 Block Public Access \(p. 402\)](#).

## Restrições e limitações

- A função especificada para executar o trabalho PUT Object ACL (ACL de objeto PUT) deve ter permissões para realizar a operação do Amazon S3 subsequente. Para mais informações sobre as permissões necessárias, consulte [PUT Object ACL \(ACL de objeto PUT\)](#) no Amazon Simple Storage Service API Reference.
- O Operações em lote do Amazon S3 usa a operação PUT Object ACL (ACL de objeto PUT) do Amazon S3 para aplicar a ACL especificada a cada objeto no manifesto. Portanto, todas as restrições e limitações aplicadas a uma operação PUT Object ACL (ACL de objeto PUT) subsequente também serão aplicadas a esses trabalhos do Operações em lote do Amazon S3. Para obter mais informações, consulte a seção [Recursos relacionados \(p. 488\)](#) desta página.

## Recursos relacionados

- [Gerenciar o acesso com ACLs \(p. 390\)](#)
- [GET Object ACL \(ACL de objeto GET\)](#) no Amazon Simple Storage Service API Reference

## Put Object Tagging (Marcação de objeto PUT)

A operação Put Object Tagging (Marcação de objeto PUT) substitui as tags de objeto do Amazon S3 de cada objeto listado no manifesto. A tag de objeto do Amazon S3 é um par de strings chave-valor que pode ser usado para armazenar metadados sobre um objeto.

Para criar um trabalho PUT Object Tagging (Marcação de objeto PUT), forneça um conjunto de tags que deseja aplicar. O Operações em lote do Amazon S3 vai aplicá-lo a cada objeto. O conjunto de tags fornecido substitui o conjunto atualmente associado aos objetos no manifesto. O Operações em lote do Amazon S3 não oferece suporte para a manutenção do conjunto existente no momento da edição de um conjunto novo.

Se os objetos no manifesto estiverem em um bucket com versão, você poderá aplicar o conjunto de tags a versões específicas de cada objeto. Para fazer isso, especifique um ID de versão para cada objeto no manifesto. Se você não incluir o ID de versão para um objeto, o Operações em lote do Amazon S3 aplicará o conjunto de tag à versão mais recente dele.

## Restrições e limitações

- A função especificada para executar o trabalho Put Object Tagging (Marcação de objeto PUT) deve ter permissões para realizar a operação do Amazon S3 subsequente. Para mais informações sobre as permissões necessárias, consulte [PUT Object Tagging \(Marcação de objeto PUT\)](#) no Amazon Simple Storage Service API Reference.
- O Operações em lote do Amazon S3 usa a operação PUT Object Tagging (Marcação de objeto PUT) do Amazon S3 para aplicar tags a cada objeto no manifesto. Portanto, todas as restrições e limitações aplicadas a uma operação PUT Object Tagging (Marcação de objeto PUT) subsequente também serão aplicadas a esses trabalhos do Operações em lote do Amazon S3. Para obter mais informações, consulte a seção [Recursos relacionados \(p. 489\)](#) desta página.

## Recursos relacionados

- [Marcação de objetos \(p. 114\)](#)
- [GET Object Tagging \(Marcação de objeto GET\)](#) no Amazon Simple Storage Service API Reference
- [PUT Object Tagging\(Marcação de objeto PUT\)](#) no Amazon Simple Storage Service API Reference

# Gerenciar operações em lote

[Cadastre-se na demonstração](#)

O Amazon S3 oferece um conjunto eficiente de ferramentas para ajudar a gerenciar os trabalhos do Operações em lote após a criação deles. Esta seção descreve as operações que é possível usar para gerenciar os trabalhos. Realize todas as operações listadas nesta seção usando o Console de gerenciamento da AWS, a AWS CLI, os SDKs da AWS ou as APIs REST.

### Tópicos

- [Listar os trabalhos \(p. 489\)](#)
- [Visualizar detalhes do trabalho \(p. 490\)](#)
- [Como atribuir prioridade aos trabalhos \(p. 490\)](#)
- [Status do trabalho \(p. 490\)](#)
- [Como monitorar falhas nos trabalhos \(p. 492\)](#)
- [Notificações e registro em log \(p. 493\)](#)
- [Relatórios de conclusão \(p. 493\)](#)

## Listar os trabalhos

Você pode recuperar uma lista dos trabalhos de operação em lote. Ela inclui os trabalhos ainda não concluídos, bem como os concluídos nos últimos 90 dias. A lista inclui informações para cada trabalho, como ID, descrição, prioridade, status atual e número de tarefas que foram bem-sucedidas e que apresentaram falha. Você pode filtrar a lista por status. Ao recuperar uma lista pelo console, você também pode pesquisar os trabalhos por descrição ou ID e filtrá-los por região da AWS.

## Visualizar detalhes do trabalho

Se quiser mais informações sobre um trabalho do que puder recuperar listando trabalhos, você poderá exibir todos os detalhes de um único trabalho. Além das informações exibidas na lista, os detalhes de cada trabalho trazem mais detalhes. Entre eles estão os parâmetros da operação, os detalhes sobre o manifesto, as informações sobre o relatório de conclusão (se você tiver configurado um quando criou o trabalho) e o nome de recurso da Amazon (ARN) da função do usuário atribuído para executar o trabalho. Exibindo os detalhes de um trabalho individual, você acessa toda a configuração de um trabalho.

## Como atribuir prioridade aos trabalhos

Você pode atribuir uma prioridade numérica a cada trabalho. Qualquer número inteiro positivo pode ser usado. O Operações em lote do Amazon S3 prioriza os trabalhos de acordo com a ordem atribuída. Os trabalhos com prioridade mais alta (ou um valor inteiro mais alto para o parâmetro de prioridade) são avaliados primeiro. A prioridade é determinada em ordem decrescente. Por exemplo, uma fila de trabalhos com um valor de prioridade 10 tem preferência de programação com relação a uma fila de trabalhos com um valor de prioridade 1.

Você pode alterar a prioridade do trabalho enquanto ele está sendo executado. Se você enviar um novo trabalho com uma prioridade mais alta enquanto um trabalho estiver em execução, o trabalho de menor prioridade poderá ser pausado para permitir a execução do trabalho de maior prioridade.

### Note

O Operações em lote do Amazon S3 cumpre prioridades de trabalho com base em melhor esforço. Embora os trabalhos com prioridades mais altas normalmente tenham precedência sobre os de prioridades mais baixas, o Amazon S3 não garante a classificação rígida dos trabalhos.

## Status do trabalho

Depois de criar um trabalho, ele passa por diversos status. A tabela a seguir descreve os status existentes e as transições possíveis entre eles.

Status	Descrição	Transições
New	Ao ser criado, o trabalho ganha o status New.	Ele passa automaticamente para o estado Preparing quando o Amazon S3 começa a processar o objeto do manifesto.
Preparing	O Amazon S3 está processando o objeto do manifesto e outros parâmetros do trabalho para configurar e executar o trabalho.	O trabalho passa automaticamente para o estado Ready depois que o Amazon S3 termina de processar o manifesto e outros parâmetros. Então, está tudo pronto para começar a executar a operação nos objetos listados no manifesto.  Se o trabalho exigir confirmação antes da execução, como ao criar um trabalho usando o console do Amazon S3, ele passará do estado Preparing para Suspended. Ele permanece no estado Suspended até

Status	Descrição	Transições
		que você confirme que deseja executá-lo.
Suspended	O trabalho exige confirmação, mas você ainda não confirmou que deseja executá-lo. Somente os trabalhos criados pelo console do Amazon S3 exigem confirmação. Os trabalhos criados pelo console entram no estado <code>Suspended</code> imediatamente depois de <code>Preparing</code> . Depois de confirmar que deseja executar o trabalho e ele entra no estado <code>Ready</code> , ele não retorna mais a <code>Suspended</code> .	Depois de confirmar que deseja executar o trabalho, o status muda para <code>Ready</code> .
Ready	O Amazon S3 está pronto para executar as operações do objeto solicitadas.	Quando começa a ser executado, o trabalho passa automaticamente de <code>Active</code> para Amazon S3. O tempo que o trabalho permanece no estado <code>Ready</code> depende do fato de outros trabalhos de maior prioridade estarem sendo executados e do tempo necessários para conclui-los.
Active	O Amazon S3 está executando a operação solicitada nos objetos listados no manifesto. Enquanto o trabalho está em <code>Active</code> , é possível monitorar seu progresso usando o console do Amazon S3 ou a operação do <code>DescribeJob</code> por meio da API REST, AWS CLI ou SDKs da AWS.	O trabalho deixa o estado <code>Active</code> quando para de executar operações nos objetos. Isso pode acontecer automaticamente, como quando um trabalho é concluído com êxito ou apresenta falha. Também pode ocorrer como resultado da ação do usuário, por exemplo, quando ele cancela o trabalho. O estado que o trabalho passa a apresentar depende do motivo da transição.
Pausing	O trabalho está passando de outro estado para <code>Paused</code> .	O trabalho passa automaticamente para <code>Paused</code> quando o estado <code>Pausing</code> é concluído.
Paused	O trabalho pode assumir o estado <code>Paused</code> se você enviar outro trabalho de maior prioridade enquanto o atual está sendo executado.	O trabalho <code>Paused</code> volta automaticamente para <code>Active</code> depois que os trabalhos de prioridade mais alta que estavam bloqueando sua execução são concluídos, suspensos ou falham.

Status	Descrição	Transições
Complete	O trabalho terminou a execução da operação solicitada em todos os objetos do manifesto. A operação pode ter sido concluída com êxito ou ter apresentado falha para cada objeto. Se você configurar a criação do relatório de conclusão, o relatório estará disponível assim que o trabalho estiver em Complete.	Complete é um estado terminal. Quando o trabalho atinge Complete, ele não muda mais de estado.
Cancelling	O trabalho está passando para o estado Cancelled.	O trabalho passa automaticamente para Cancelled quando o estado Cancelling é concluído.
Cancelled	Você solicitou que o trabalho fosse cancelado e o Operações em lote do Amazon S3 fez o cancelamento com sucesso. O trabalho não enviará novas solicitações ao Amazon S3.	Cancelled é um estado terminal. Quando o trabalho atinge Cancelled, ele não muda mais de estado.
Failing	O trabalho está passando para o estado Failed.	O trabalho passa automaticamente para Failed quando o estado Failing é concluído.
Failed	O trabalho apresentou falha e não está mais em execução. Para obter mais informações sobre falhas em trabalhos, consulte <a href="#">Como monitorar falhas nos trabalhos (p. 492)</a> .	Failed é um estado terminal. Quando o trabalho atinge Failed, ele não muda mais de estado.

## Como monitorar falhas nos trabalhos

Se uma operação em lote encontrar um problema que o impeça de ser executado com êxito, como não ser capaz de ler o manifesto especificado, ela falha. Quando falha, um trabalho gera um ou mais códigos ou motivos de falha. O Operações em lote do Amazon S3 armazena esses códigos e motivos com o trabalho, de maneira que seja possível exibi-los solicitando os detalhes do trabalho. Caso você tenha solicitado um relatório de conclusão para o trabalho, os códigos e os motivos de falha também são exibidos.

Para evitar que trabalhos executem um grande número de operações malsucedidas, o Amazon S3 impõe um limite de falhas por tarefa em cada trabalho do Operações em lote. O Amazon S3 monitora a taxa de falhas das tarefas depois que o trabalho executa pelo menos 1.000 tarefas. Se, a qualquer momento, a taxa de falha (o número de tarefas que falharam em proporção ao número total de tarefas executadas) exceder 50%, o trabalho vai falhar. Se o trabalho falhar porque excedeu o limite de falhas da tarefa, você poderá identificar a causa das falhas. Por exemplo, você pode ter incluído por acidente alguns objetos no manifesto que não existem no bucket especificado. Depois de corrigir os erros, você pode reenviar o trabalho.

#### Note

O Operações em lote do Amazon S3 funciona de maneira assíncrona e não necessariamente executa tarefas na ordem em que os objetos estão listados no manifesto. Portanto, não é possível usar a ordem do manifesto para determinar quais tarefas dos objetos foram bem-sucedidas e quais falharam. Em vez disso, examine o relatório de conclusão do trabalho (caso você tenha solicitado um) ou exiba os logs de evento do AWS CloudTrail para ajudar a determinar a origem das falhas.

## Notificações e registro em log

Além de solicitar relatórios de conclusão, você pode também registrar, examinar e auditar a atividade do Operações em lote usando eventos do Amazon S3. À medida que avança, o trabalho emite eventos que você pode registrar usando o AWS CloudTrail, Amazon Simple Notification Service (Amazon SNS) e Amazon Simple Queue Service (Amazon SQS). Como o Batch Operations usa APIs do Amazon S3 existentes para realizar tarefas, essas tarefas também emitem os mesmos eventos que emitiriam se você as chamasse diretamente. Por isso, você pode rastrear e registrar o andamento do trabalho e todas as tarefas usando as mesmas ferramentas de notificação, registro em log e auditoria, além dos processos já usados com o Amazon S3. Para obter mais informações sobre eventos do Amazon S3, consulte [Configurar notificações de evento do Amazon S3 \(p. 522\)](#).

## Relatórios de conclusão

Ao criar um trabalho, você solicita um relatório de conclusão. Desde que o Operações em lote do Amazon S3 invoque com êxito pelo menos uma tarefa, ele vai gerar um relatório de conclusão depois de concluir a execução de tarefas, falhar ou ser cancelado. Configure o relatório de conclusão para incluir todas as tarefas ou somente tarefas com falha.

O relatório de conclusão inclui a configuração do trabalho, o status e as informações de cada tarefa, inclusive a chave e a versão do objeto, status, códigos de erro e descrições de eventuais erros. Mesmo que você não configure o relatório de conclusão, ainda poderá monitorar e auditar o trabalho e suas tarefas usando CloudTrail, Amazon CloudWatch, Amazon SNS e Amazon SQS. No entanto, os relatórios de conclusão oferecem uma maneira fácil de visualizar os resultados das tarefas em um formato consolidado, sem a necessidade de configurações adicionais.

# Hospedagem de um site estático no Amazon S3

Você pode hospedar um site estático no Amazon Simple Storage Service (Amazon S3). Em um site estático, as páginas da Web individuais incluem conteúdo estático. Elas também podem conter scripts do lado do cliente. Em contrapartida, um site dinâmico conta com o processamento no lado do servidor, incluindo scripts como PHP, JSP ou ASP.NET. O Amazon S3 não oferece suporte a scripts no lado do servidor. A Amazon Web Services (AWS) também tem recursos para hospedagem de sites dinâmicos. Para saber mais sobre hospedagem de sites na AWS, acesse [Sites e hospedagem de sites](#).

## Tópicos

- [Endpoints de site \(p. 495\)](#)
- [Configuração de bucket para hospedagem de site \(p. 496\)](#)
- [Demonstrações de exemplo - Hospedagem de sites no Amazon S3 \(p. 509\)](#)

Para hospedar um site estático, você configura um bucket do Amazon S3 para hospedagem de sites e faz upload do conteúdo do seu site no bucket. Esse bucket deve ter acesso público de leitura. É intencional que todos tenham acesso de leitura a esse bucket. O site fica disponível no endpoint do site da região da AWS específica do bucket, que está em um dos seguintes formatos:

```
<bucket-name>.s3-website-<AWS-region>.amazonaws.com
```

```
<bucket-name>.s3-website.<AWS-region>.amazonaws.com
```

Para obter uma lista de endpoints de site específicos da região da AWS para o Amazon S3, consulte [Endpoints de site \(p. 495\)](#). Por exemplo, suponha que você crie um bucket chamado `examplebucket` em Região Oeste dos EUA (Oregon) e o configure como um site. Os seguintes URLs de exemplo fornecem acesso ao conteúdo de seu site:

- Este URL retorna um documento de indexação padrão que você configurou para o site.

```
http://examplebucket.s3-website-us-west-2.amazonaws.com/
```

- Este URL solicita o objeto `photo.jpg`, que está armazenado no nível raiz do bucket.

```
http://examplebucket.s3-website-us-west-2.amazonaws.com/photo.jpg
```

- Este URL solicita o objeto `docs/doc1.html` em seu bucket.

```
http://examplebucket.s3-website-us-west-2.amazonaws.com/docs/doc1.html
```

## Usar seu próprio domínio

Em vez de acessar usando um endpoint de site do Amazon S3, você pode usar seu próprio domínio, como `example.com` para atender ao seu conteúdo. O Amazon S3, junto com o Amazon Route 53, oferece suporte à hospedagem de um site no domínio raiz. Por exemplo, se você tiver o domínio raiz `example.com` e quiser hospedar seu site no Amazon S3, os visitantes do site poderão acessá-lo no navegador digitando `http://www.example.com` ou `http://example.com`. Para ver uma demonstração de exemplo, consulte [Exemplo: configurar um site estático usando um domínio personalizado \(p. 511\)](#).

Para configurar um bucket para hospedagem de sites, você adiciona a configuração de site ao bucket. Para obter mais informações, consulte [Configuração de bucket para hospedagem de site \(p. 496\)](#).

## Endpoints de site

Quando você configura um bucket para hospedagem de sites, o site é disponibilizado pelo endpoint de site específico da região. Os endpoints de site são diferentes dos endpoints para onde você envia solicitações de API REST. Para obter mais informações sobre as diferenças entre os endpoints, consulte [Principais diferenças entre o site da Amazon e o endpoint de API REST \(p. 495\)](#).

As duas formas gerais de um endpoint de site do Amazon S3 são:

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

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

A forma usada para o endpoint depende da região em que o bucket reside. Por exemplo, se o seu bucket chama `example-bucket` e reside na região Oeste dos EUA (Oregon), o site está disponível no seguinte endpoint de site do Amazon S3:

`http://example-bucket.s3-website-us-west-2.amazonaws.com/`

Ou, se o seu bucket chama `example-bucket` e reside na região UE (Frankfurt), o site está disponível no seguinte endpoint de site do Amazon S3:

`http://example-bucket.s3-website.eu-central-1.amazonaws.com/`

Para obter uma lista dos endpoints de site do Amazon S3 por região, consulte [Endpoints de site do Amazon Simple Storage Service](#) na Referência geral da AWS.

Para que seus clientes acessem o conteúdo no endpoint de site, você deve fazer com que seu conteúdo seja publicamente legível. Para fazer isso, você pode usar uma política de bucket ou uma ACL em um objeto para conceder as permissões necessárias.

### Note

Os buckets de Pagamento pelo solicitante não permitem acesso pelo endpoint do site. Qualquer solicitação para tal bucket recebe uma resposta 403 Access Denied. Para obter mais informações, consulte [Buckets de Pagamento pelo solicitante \(p. 83\)](#).

Se você tiver um domínio registrado, poderá adicionar uma entrada DNS CNAME para apontar para o endpoint de site do Amazon S3. Por exemplo, se você registrou o domínio `www.example-bucket.com`, pode criar um bucket `www.example-bucket.com` e adicionar um registro DNS CNAME que aponte para `www.example-bucket.com.s3-website-<region>.amazonaws.com`. Todas as solicitações a `http://www.example-bucket.com` são redirecionadas para `www.example-bucket.com.s3-website-<region>.amazonaws.com`. Para obter mais informações, consulte [Hospedagem virtual de buckets \(p. 46\)](#).

## Principais diferenças entre o site da Amazon e o endpoint de API REST

O endpoint de site é otimizado para acesso de um navegador da web. A tabela a seguir descreve as principais diferenças entre o endpoint de API REST da Amazon e o endpoint de site.

Principal diferença	Endpoint de API REST	Endpoint de site
Controle de acesso	Oferece suporte a conteúdo público e privado.	Oferece suporte apenas a conteúdo publicamente legível.
Manuseio de mensagens de erro	Retorna uma resposta de erro formatada em XML.	Retorna um documento HTML.
Suporte a redirecionamento	Não aplicável	Oferece suporte a redirecionamentos no nível do objeto e do bucket.
Solicitações com suporte	Oferece suporte a todas as operações de bucket e de objeto	Oferece suporte apenas a solicitações GET e HEAD em objetos.
Responde a solicitações GET e HEAD na raiz de um bucket	Retorna uma lista de chaves de objeto no bucket.	Retorna o documento de índice especificado na configuração de site.
Suporte a Secure Sockets Layer (SSL)	Oferece suporte a conexões SSL.	Não oferece suporte a conexões SSL.

Para obter uma lista dos endpoints do Amazon S3, consulte [Endpoints de solicitações \(p. 11\)](#).

## Configuração de bucket para hospedagem de site

Você pode hospedar um site estático em um bucket do Amazon Simple Storage Service (Amazon S3). Contudo, fazer isso requer algumas configurações. Algumas configurações opcionais também estão disponíveis, dependendo dos requisitos do seu site.

Configurações obrigatórias:

- [Habilitar a hospedagem de sites \(p. 496\)](#)
- [Configuração de suporte a documento de índice \(p. 497\)](#)
- [Permissões necessárias para acesso ao site \(p. 499\)](#)

Configurações opcionais:

- [\(Opcional\) Configurar o registro em log de tráfego da web \(p. 500\)](#)
- [\(Opcional\) Suporte a documento de erro personalizado \(p. 500\)](#)
- [\(Opcional\) Configuração de um redirecionamento de uma página da web \(p. 502\)](#)

## Habilitar a hospedagem de sites

Siga estas etapas para habilitar a hospedagem de sites para seus buckets do Amazon S3 usando o console do [Amazon S3](#):

Para habilitar a hospedagem de sites para um bucket do Amazon S3

1. Faça login no Console de gerenciamento da AWS e abra o console da Amazon S3 em <https://console.aws.amazon.com/s3/>.

2. Na lista, escolha o bucket que você deseja usar para seu site hospedado.
3. Escolha a guia Properties (Propriedades).
4. Escolha Static website hosting (Hospedagem de site estático) e selecione Use this bucket to host a website (Usar este bucket para hospedar um site).
5. Você é solicitado a fornecer o documento de índice, além de quaisquer documentos de erro opcionais e regras de redirecionamento que sejam necessários.

Para obter informações sobre o que é um documento de índice, consulte [Configuração de suporte a documento de índice \(p. 497\)](#).

## Configuração de suporte a documento de índice

Um documento de índice é uma página da web que o Amazon S3 retorna quando uma solicitação é feita para a raiz de um site ou para qualquer subpasta. Por exemplo, se um usuário insere `http://www.example.com` no navegador, ele não está solicitando nenhuma página específica. Nesse caso, o Amazon S3 exibe o documento de índice, que às vezes é referido como a página padrão.

Ao configurar seu bucket como um site, forneça o nome do documento de índice. Você pode fazer upload de um objeto com esse nome e configura-lo para ser publicamente legível.

A barra no final do URL no nível raiz é opcional. Por exemplo, se você configurar seu site com `index.html` como o documento de índice, qualquer um dos dois URLs a seguir retornará `index.html`.

```
http://example-bucket.s3-website-region.amazonaws.com/  
http://example-bucket.s3-website-region.amazonaws.com
```

Para obter mais informações sobre endpoints de site do Amazon S3, consulte [Endpoints de site \(p. 495\)](#).

## Documentos de índice e pastas

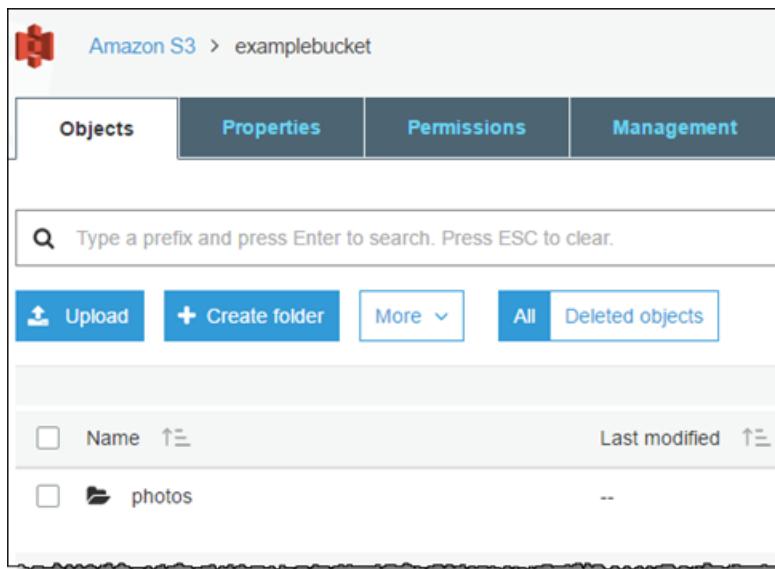
No Amazon S3, um bucket é um contêiner de objetos simples; ele não fornece qualquer organização hierárquica, pois o sistema de arquivos em seu computador faz isso. Você pode criar uma hierarquia lógica usando nomes de chave de objeto que sugerem uma estrutura de pastas. Por exemplo, considere um bucket com três objetos e os seguintes nomes de chave.

- `sample1.jpg`
- `photos/2006/Jan/sample2.jpg`
- `photos/2006/Feb/sample3.jpg`

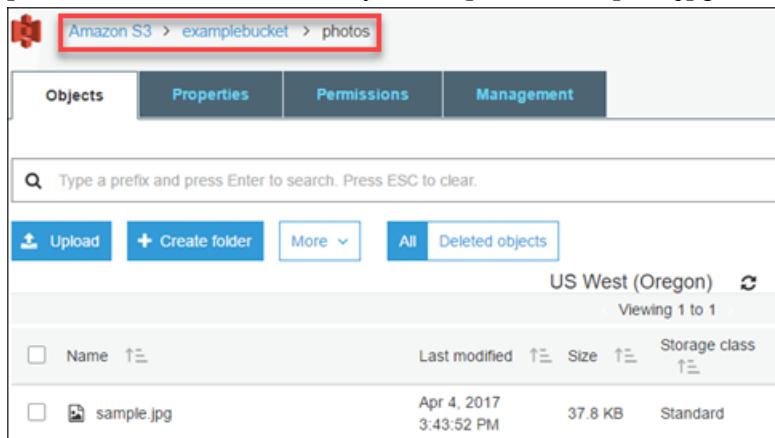
Embora esses objetos sejam armazenados sem uma organização hierárquica física, você pode pressupor a seguinte estrutura de pastas lógica com base nos nomes de chave.

- O objeto `sample1.jpg` está na raiz do bucket.
- O objeto `sample2.jpg` está na subpasta `photos/2006/Jan`.
- O objeto `sample3.jpg` está na subpasta `photos/2006/Feb`.

O conceito de pasta para o qual o console do Amazon S3 oferece suporte é baseado em nomes de chave de objeto. Para continuar o exemplo anterior, o console exibe o `examplebucket` como uma pasta `photos`.



Você pode fazer upload de objetos no bucket ou na pasta `photos` no bucket. Se você adicionar o objeto `sample.jpg` ao bucket, o nome da chave será `sample.jpg`. Se você fizer upload do objeto na pasta `photos`, o nome da chave de objeto será `photos/sample.jpg`.



Se você criar essa estrutura de pastas em seu bucket, deverá ter um documento de índice em cada nível. Quando um usuário especificar um URL que se assemelhe a uma consulta de pasta, a presença ou a ausência de uma barra no final determinará o comportamento do site. Por exemplo, o URL a seguir, com uma barra no final, retorna o documento de índice `photos/index.html`.

```
http://example-bucket.s3-website-region.amazonaws.com/photos/
```

Contudo, se você excluir a barra no final do URL anterior, o Amazon S3 procurará primeiro um objeto `photos` no bucket. Se o objeto `photos` não for encontrado, ele procurará um documento de índice, `photos/index.html`. Se esse documento for encontrado, o Amazon S3 retornará uma mensagem 302 Found e apontará para a chave `photos/`. Para solicitações subsequentes a `photos/`, o Amazon S3 retorna `photos/index.html`. Se o documento de índice não for encontrado, o Amazon S3 retornará um erro.

## Permissões necessárias para acesso ao site

Quando você configura um bucket como um site, deve definir os objetos que deseja exibir como publicamente legíveis. Para fazer isso, grave uma política de bucket que conceda a todos a permissão s3:GetObject. No endpoint de site, se um usuário solicitar um objeto que não existe, o Amazon S3 retornará um código de resposta HTTP 404 (Not Found). Se o objeto existir, mas não você não tiver permissão de leitura nele, o endpoint de site retornará o código de resposta HTTP 403 (Access Denied). O usuário pode usar o código de resposta para inferir se um objeto específico existe. Se você não quiser esse comportamento, não ative o suporte de site para seu bucket.

O exemplo de política de bucket a seguir garante a qualquer um acesso aos objetos na pasta especificada. Para obter mais informações sobre políticas de bucket, consulte [Uso de políticas de bucket e políticas de usuário \(p. 326\)](#).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "PublicReadGetObject",  
            "Effect": "Allow",  
            "Principal": "*",  
            "Action": ["s3:GetObject"],  
            "Resource": ["arn:aws:s3:::example-bucket/*"]  
        }  
    ]  
}
```

### Note

Mantenha o seguinte em mente:

- Para hospedar um site, seu bucket deve ter acesso público de leitura. É intencional que todos tenham acesso de leitura a esse bucket.
- A política de bucket se aplica somente a objetos que pertencem ao proprietário do bucket. Se o seu bucket contiver objetos que não pertençam ao proprietário do bucket, a permissão READ pública nesses objetos deverá ser concedida usando a lista de controle de acesso (ACL) do objeto.

Você pode conceder permissão de leitura pública aos seus objetos usando uma política de bucket ou uma ACL do objeto. Para tornar um objeto publicamente legível usando uma ACL, conceda a permissão READ ao grupo AllUsers, como mostrado no elemento de concessão a seguir. Adicione esse elemento de concessão à ACL do objeto. Para obter informações sobre o gerenciamento de ACLs, consulte [Gerenciar o acesso com ACLs \(p. 390\)](#).

```
<Grant>  
    <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
             xsi:type="Group">  
        <URI>http://acs.amazonaws.com/groups/global/AllUsers</URI>  
    </Grantee>  
    <Permission>READ</Permission>  
</Grant>
```

## (Opcional) Configurar o registro em log de tráfego da web

Se você deseja rastrear o número de visitantes que acessam seu site, ative o registro em log para o bucket de domínio raiz. A habilitação do registro em log é opcional.

Para habilitar o registro em log para o bucket do domínio raiz

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Crie um bucket para registro em log chamado logs.*example*.com na mesma região da AWS em que os buckets example.com e www.example.com foram criados.
3. Crie duas pastas no bucket logs.example.com; uma chamada root e a outra chamada cdn. Se você configurar o Amazon CloudFront para agilizar seu site, usará a pasta cdn.
4. Na lista Bucket name (Nome do bucket), escolha o bucket do domínio raiz, Properties (Propriedades) e Server access logging (Registro de acesso ao servidor).
5. Selecione Enable logging (Habilitar registro).
6. Em Target bucket (Bucket de destino), escolha o bucket criado por você para os arquivos de log, logs.example.com.
7. Para Target prefix (Prefixo de destino), digite **root/**. Essa configuração agrupa os arquivos de dados de log no bucket em uma pasta chamada root, de maneira que seja fácil localizá-los depois.
8. Escolha Save (Salvar).

Agora é possível revisar seus logs no bucket logs.*example*.com, em root e cdn folders.

## (Opcional) Suporte a documento de erro personalizado

A tabela a seguir lista o subconjunto de códigos de resposta HTTP que o Amazon S3 retorna quando ocorre um erro.

Código de erro HTTP	Descrição
301 Moved Permanently (301 movido permanentemente)	Quando um usuário enviar uma solicitação diretamente a endpoints de site do Amazon S3 ( <code>http://s3-website-&lt;region&gt;.amazonaws.com/</code> ), o Amazon S3 retornará uma resposta 301 Moved Permanently (301 movido permanentemente) e redirecionará essas solicitações para <code>https://aws.amazon.com/s3/</code> .
302 Found (302 Encontrado)	Quando o Amazon S3 recebe uma solicitação para uma chave x, <code>http://&lt;bucket&gt;.s3-website-&lt;region&gt;.amazonaws.com/x</code> , sem uma barra no final, ele tenta localizar o objeto com o nome de chave x. Se o objeto não for encontrado, o Amazon S3 determinará que a solicitação é para a subpasta x e redirecionará a solicitação adicionando uma barra no final, e retornará 302 Found (302 Encontrado).
304 Not Modified (304 Não modificado)	Os usuários do Amazon S3 solicitam os cabeçalhos If-Modified-Since, If-Unmodified-Since, If-Match e/ou If-None-Match para determinar se o objeto solicitado é igual ao da cópia em cache guardada pelo cliente. Se o objeto for o mesmo, o endpoint do site retornará uma resposta 304 Not Modified (304 não modificados).

Código de erro HTTP	Descrição
400 Malformed Request (400 Solicitação malformada)	O endpoint de site responde com 400 Malformed Request (400 Solicitação malformada) quando um usuário tenta acessar um bucket pelo endpoint regional incorreto.
403 Forbidden (403 Proibido)	O endpoint de site responde com 403 Forbidden (403 Proibido) quando uma solicitação de usuário se traduz em um objeto que não é publicamente legível. O proprietário do objeto deve tornar o objeto publicamente legível usando uma política de bucket ou uma ACL.
404 Not Found (404 Não encontrado)	O endpoint de site responde com 404 Not Found (404 Não encontrado) pelos seguintes motivos: <ul style="list-style-type: none"> <li>• O Amazon S3 determina que o URL do site refere-se a uma chave de objeto que não existe.</li> <li>• A Amazon pressupõe que a solicitação é para um documento de índice que não existe.</li> <li>• Um bucket especificado no URL não existe.</li> <li>• Um bucket especificado no URL existe, mas não é configurado como um site.</li> </ul> Você pode criar um documento personalizado que é retornado em caso de 404 Not Found (404 Não encontrado). Certifique-se de que o documento seja carregado no bucket configurado como um site e que a configuração de hospedagem de sites esteja definida para usar o documento. Para obter informações sobre como o Amazon S3 interpreta a URL como uma solicitação de um objeto ou um documento de índice, consulte <a href="#">Configuração de suporte a documento de índice (p. 497)</a> .
500 Service Error (500 Erro de serviço)	O endpoint de site responde com 500 Service Error (500 Erro de serviço) quando ocorre um erro interno de servidor.
503 Service Unavailable (503 Serviço não disponível)	O endpoint de site responde com 503 Service Unavailable (503 Serviço não disponível) quando o Amazon S3 determina que você precisa reduzir sua taxa de solicitações.

Para cada um desses erros, o Amazon S3 retorna uma mensagem HTML predefinida. Veja a seguir uma mensagem HTML de exemplo que é retornada para a resposta 403 Forbidden (403 Proibido).

## 403 Forbidden

- Code: AccessDenied
- Message: Access Denied
- RequestId: 873CA367A51F7EC7
- HostId: DdQezl9vkuw5huD5HKsFaTDm9KH4PZzCPRkW3igimILbTu1DiYlvXjgyd7pVxq32

### An Error Occurred While Attempting to Retrieve a Custom Error Document

- Code: AccessDenied
- Message: Access Denied

#### Documento de erro personalizado

É possível fornecer um documento de erro personalizado que contenha uma mensagem de erro amigável e ajuda adicional. Você fornece esse documento de erro personalizado como parte da inclusão da configuração do site em seu bucket. O Amazon S3 retorna seu documento de erro personalizado apenas para a classe HTTP 4XX de códigos de erro.

#### Documentos de erro e comportamento do navegador

Quando ocorre um erro, o Amazon S3 retorna um documento de erro HTML. Se você configurou seu site com um documento de erro personalizado, o Amazon S3 retornará esse documento de erro. No entanto, quando um erro ocorre, alguns navegadores exibem sua própria mensagem, ignorando o documento de erro que o Amazon S3 retorna. Por exemplo, quando ocorre um erro HTTP 404 Not Found, o Google Chrome pode ignorar o documento de erro que o Amazon S3 retorna e exibir seu próprio erro.

## (Opcional) Configuração de um redirecionamento de uma página da web

Se seu bucket do Amazon S3 estiver configurado para hospedagem de sites, você poderá redirecionar solicitações de um objeto para outro objeto no mesmo bucket ou para um URL externo.

#### Tópicos

- [Suporte a redirecionamento de página no console do Amazon S3 \(p. 502\)](#)
- [Configuração de um redirecionamento de página na API REST \(p. 504\)](#)
- [Redirecionamentos condicionais avançados \(p. 504\)](#)

Você define o redirecionamento adicionando a propriedade `x-amz-website-redirect-location` aos metadados do objeto. O site interpreta o objeto como um redirecionamento 301. Para reorientar uma solicitação para outro objeto, você define o local de redirecionamento como a chave do objeto de destino. Para redirecionar uma solicitação para um URL externo, defina o local de redirecionamento como o URL desejado. Para obter mais informações sobre metadados de objeto, consulte [Metadados definidos por sistema \(p. 105\)](#).

Um bucket configurado para hospedagem de sites tem o endpoint de site e o endpoint REST. Uma solicitação para uma página que é configurada como redirecionamento 301 tem os seguintes resultados possíveis, dependendo do endpoint da solicitação:

- Endpoint de site específico da região— O Amazon S3 redireciona a solicitação da página de acordo com o valor da propriedade `x-amz-website-redirect-location`.
- Endpoint REST – O Amazon S3 não redireciona a solicitação da página. Ele retorna o objeto solicitado.

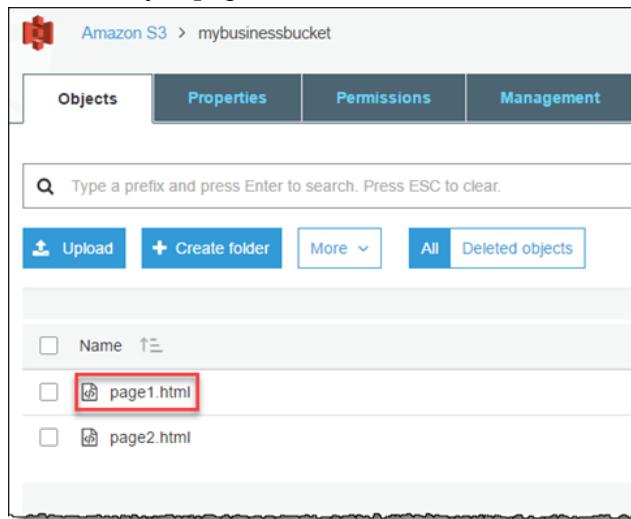
Para obter mais informações sobre os endpoints, consulte [Principais diferenças entre o site da Amazon e o endpoint de API REST \(p. 495\)](#).

Você pode definir um redirecionamento de página no console do Amazon S3 ou usando a API REST do Amazon S3.

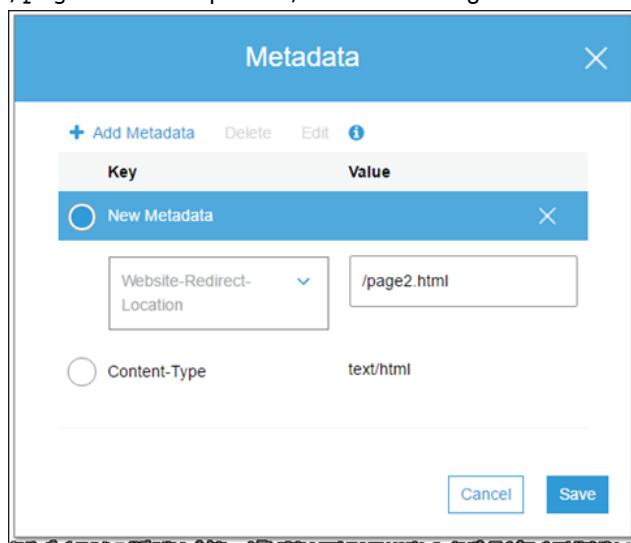
## Suporte a redirecionamento de página no console do Amazon S3

Você pode usar o console do Amazon S3 para definir o local de redirecionamento de site nos metadados do objeto. Quando você define um redirecionamento de página, pode manter ou excluir o conteúdo do objeto de origem. Por exemplo, suponha que você tenha um objeto `page1.html` em seu bucket. Para redirecionar todas as solicitações dessa página para outro objeto, `page2.html`, você pode adotar um destes procedimentos:

- Para manter o conteúdo do objeto page1.html e apenas redirecionar as solicitações de página, escolha o objeto page1.html.



Escolha a guia Propriedades para page1.html e escolha a caixa Metadados. Adicione Website Redirect Location aos metadados, como mostrado no exemplo a seguir, e defina seu valor como /page2.html. O prefixo / no valor é obrigatório.



Você também pode definir o valor como um URL externo, como <http://www.example.com>. Por exemplo, se o seu domínio raiz for `example.com`, e você deseja servir solicitações para `http://example.com` e `http://www.example.com`, será possível criar dois buckets chamados `example.com` e `www.example.com`. Então, mantenha o conteúdo em um dos buckets (digamos `example.com`) e configure o outro bucket para redirecionar todas as solicitações para o bucket `example.com`.

- Para excluir o conteúdo do objeto page1.html e redirecionar as solicitações, você pode fazer upload de um novo objeto com zero byte com a mesma chave page1.html, para substituir o objeto existente. Especifique Website Redirect Location para page1.html no processo de upload. Para obter informações sobre o upload de um objeto, consulte [Fazer upload de objetos do S3](#) no Guia do usuário do console do Amazon Simple Storage Service.

## Configuração de um redirecionamento de página na API REST

As ações de API do Amazon S3 oferecem suporte ao cabeçalho `x-amz-website-redirect-location` na solicitação. O Amazon S3 armazena o valor de cabeçalho nos metadados de objeto como `x-amz-website-redirect-location`.

- [Objeto PUT](#)
- [Iniciar multipart upload](#)
- [Objeto POST](#)
- [Objeto PUT - Copiar](#)

Quando você define um redirecionamento de página, pode manter ou excluir o conteúdo de objeto. Por exemplo, suponha que você tenha um objeto `page1.html` em seu bucket.

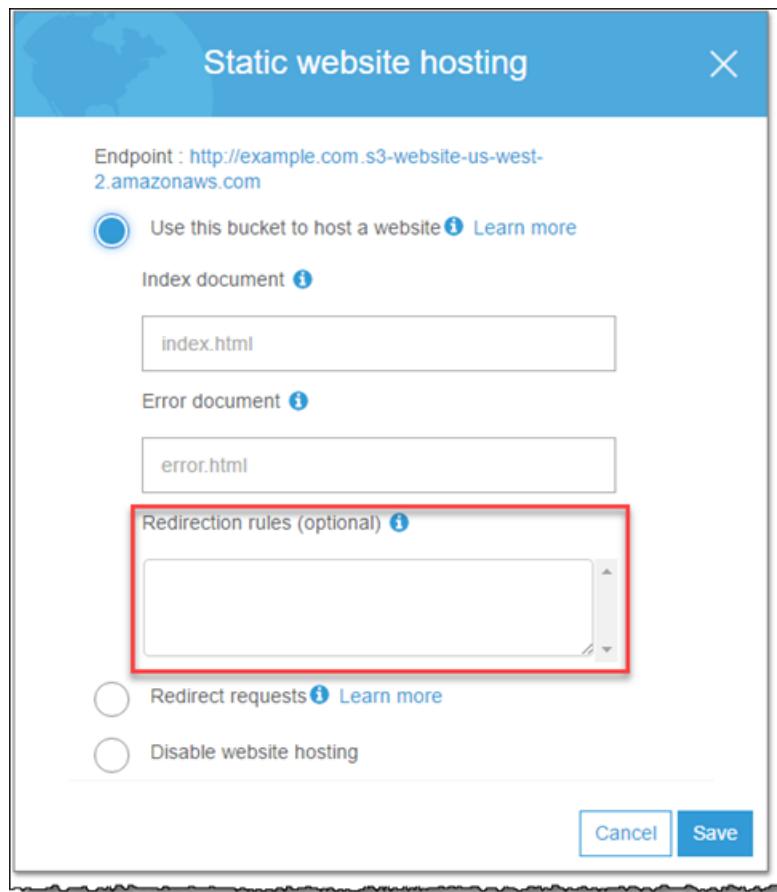
- Para manter o conteúdo de `page1.html` e apenas redirecionar as solicitações de página, envie uma solicitação [PUT objeto - Copiar](#) para criar um novo objeto `page1.html` que usa o objeto `page1.html` existente como origem. Na sua solicitação, você define o cabeçalho `x-amz-website-redirect-location`. Quando a solicitação for concluída, você terá a página original com o conteúdo inalterado, mas o Amazon S3 redirecionará todas as solicitações da página para o local de redirecionamento especificado.
- Para excluir o conteúdo do objeto `page1.html` e redirecionar as solicitações da página, você pode enviar uma solicitação [PUT objeto](#) para fazer upload de um objeto com zero byte com a mesma chave de objeto: `page1.html`. Na solicitação [PUT](#), você define `x-amz-website-redirect-location` para `page1.html` como o novo objeto. Quando a solicitação for concluída, `page1.html` não terá nenhum conteúdo, e as solicitações serão redirecionadas para o local que é especificado por `x-amz-website-redirect-location`.

Quando você recupera o objeto usando a ação [Objeto GET](#) com outros metadados de objeto, o Amazon S3 retorna o cabeçalho `x-amz-website-redirect-location` na resposta.

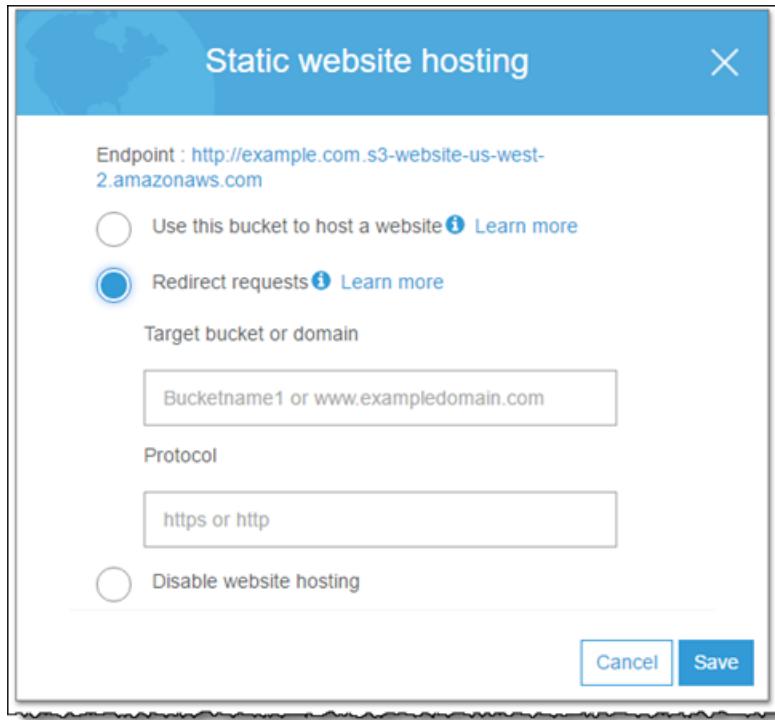
## Redirecionamentos condicionais avançados

Usando regras de redirecionamentos avançadas, você pode rotear solicitações condicionalmente de acordo com nomes de chave de objeto específicos, prefixos na solicitação ou códigos da resposta. Por exemplo, suponha que você exclua ou dê outro nome a um objeto em seu bucket. Você pode adicionar uma regra de roteamento que redireciona a solicitação a outro objeto. Se você deseja tornar uma pasta indisponível, será possível adicionar uma regra de roteamento para redirecionar a solicitação para outra página da Web. Você também pode adicionar uma regra de roteamento para processar condições de erro, encaminhando solicitações que retornam o erro para outro domínio quando ele é processado.

Ao configurar o bucket para hospedagem de sites, você tem a opção de especificar regras avançadas de redirecionamento.



Para redirecionar todas as solicitações ao endpoint de site do bucket para outro host, você só precisa fornecer o nome do host.



Você descreve as regras usando XML. A próxima seção fornece a sintaxe geral e exemplos de como especificar regras de redirecionamento.

## Sintaxe para especificar regras de roteamento

Veja a seguir a sintaxe geral para definir regras de roteamento em uma configuração de site:

```
<RoutingRules> =
  <RoutingRules>
    <RoutingRule>...</RoutingRule>
    [<RoutingRule>...</RoutingRule>
     ...]
  </RoutingRules>

<RoutingRule> =
  <RoutingRule>
    [ <Condition>...</Condition> ]
    <Redirect>...</Redirect>
  </RoutingRule>

<Condition> =
  <Condition>
    [ <KeyPrefixEquals>...</KeyPrefixEquals> ]
    [ <HttpErrorCodeReturnedEquals>...</HttpErrorCodeReturnedEquals> ]
  </Condition>
  Note: <Condition> must have at least one child element.

<Redirect> =
  <Redirect>
    [ <HostName>...</HostName> ]
    [ <Protocol>...</Protocol> ]
    [ <ReplaceKeyPrefixWith>...</ReplaceKeyPrefixWith> ]
    [ <ReplaceKeyWith>...</ReplaceKeyWith> ]
    [ <HttpRedirectCode>...</HttpRedirectCode> ]
```

```
</Redirect>
Note: <Redirect> must have at least one child element.
Also, you can have either ReplaceKeyPrefix with or ReplaceKeyWith,
but not both.
```

A tabela a seguir descreve os elementos na regra de roteamento.

Nome	Descrição
<b>RoutingRules</b>	Contêiner para um conjunto de elementos <b>RoutingRule</b> .
<b>RoutingRule</b>	<p>Uma regra que identifica uma condição e o redirecionamento que é aplicado quando a condição é satisfeita.</p> <p>Condição: um contêiner <b>RoutingRules</b> deve ter pelo menos uma regra de roteamento.</p>
<b>Condition</b>	O contêiner para descrever uma condição que deve ser satisfeita para que o redirecionamento especificado seja aplicado. Se a regra de roteamento não incluir uma condição, a regra será aplicada a todas as solicitações.
<b>KeyPrefixEquals</b>	<p>O prefixo do nome da chave de objeto do qual as solicitações são redirecionadas.</p> <p><b>KeyPrefixEquals</b> será obrigatório se <b>HttpErrorCodeReturnedEquals</b> não for especificado. Se <b>KeyPrefixEquals</b> e <b>HttpErrorCodeReturnedEquals</b> forem especificados, ambos deverão ser verdadeiros para a condição ser satisfeita.</p>
<b>HttpErrorCodeReturnedEquals</b>	<p>O código de erro HTTP que deve corresponder para que o redirecionamento seja aplicado. Se ocorrer um erro, e se o código de erro satisfizer esse valor, o redirecionamento especificado será aplicado.</p> <p><b>HttpErrorCodeReturnedEquals</b> será obrigatório se <b>KeyPrefixEquals</b> não for especificado. Se <b>KeyPrefixEquals</b> e <b>HttpErrorCodeReturnedEquals</b> forem especificados, ambos deverão ser verdadeiros para a condição ser satisfeita.</p>
<b>Redirect</b>	O elemento do contêiner que fornece instruções para o redirecionamento da solicitação. Você pode redirecionar solicitações para outro host, ou outra página, ou pode especificar outro protocolo a ser usado. Uma <b>RoutingRule</b> deve ter um elemento <b>Redirect</b> . Um elemento <b>Redirect</b> deve conter pelo menos um dos seguintes elementos irmãos: <b>Protocol</b> , <b>HostName</b> , <b>ReplaceKeyPrefixWith</b> , <b>ReplaceKeyWith</b> ou <b>HttpRedirectCode</b> .
<b>Protocol</b>	O protocolo, http or https, que será usado no cabeçalho <b>Location</b> que é retornado na resposta.  Se um dos irmãos for fornecido, <b>Protocol</b> não será obrigatório.
<b>HostName</b>	O nome do host a ser usado no cabeçalho <b>Local</b> que é retornado na resposta.  Se um dos irmãos for fornecido, <b>HostName</b> não será obrigatório.

Nome	Descrição
ReplaceKeyPrefixWith	O prefixo do nome de chave de objeto que substitui o valor de KeyPrefixEquals na solicitação de redirecionamento.  Se um dos irmãos for fornecido, ReplaceKeyPrefixWith não será obrigatório. Poderá ser fornecido somente se ReplaceKeyWith não for fornecido.
ReplaceKeyWith	A chave de objeto a ser usada no cabeçalho Local que é retornado na resposta.  Se um dos irmãos for fornecido, ReplaceKeyWith não será obrigatório. Poderá ser fornecido somente se ReplaceKeyPrefixWith não for fornecido.
HttpRedirectCode	O código de redirecionamento HTTP a ser usado no cabeçalho Local que é retornado na resposta.  Se um dos irmãos for fornecido, HttpRedirectCode não será obrigatório.

Os seguintes exemplos explicam tarefas comuns de redirecionamento:

#### Example 1: Redirecionar após trocar o nome de um prefixo de chave

Suponha que seu bucket contenha os seguintes objetos:

- index.html
- docs/article1.html
- docs/article2.html

Você decide renomear a pasta de docs/ para documents/. Depois de fazer essa alteração, você precisará redirecionar as solicitações do prefixo docs/ para documents/. Por exemplo, as solicitações para docs/article1.html serão redirecionadas para documents/article1.html.

Nesse caso, você adiciona a seguinte regra de roteamento à configuração de site:

```
<RoutingRules>
  <RoutingRule>
    <Condition>
      <KeyPrefixEquals>docs/</KeyPrefixEquals>
    </Condition>
    <Redirect>
      <ReplaceKeyPrefixWith>documents/</ReplaceKeyPrefixWith>
    </Redirect>
  </RoutingRule>
</RoutingRules>
```

#### Example 2: Redirecionar solicitações de uma pasta excluída para uma página

Suponha que você tenha excluído a pasta images/ (ou seja, você excluiu todos os objetos com o prefixo de chave images/). Você pode adicionar uma regra de roteamento que redirecione as solicitações para os objetos com o prefixo de chave images/ a uma página chamada folderdeleted.html.

```
<RoutingRules>
  <RoutingRule>
```

```
<Condition>
    <KeyPrefixEquals>images/</KeyPrefixEquals>
</Condition>
<Redirect>
    <ReplaceKeyWith>folderdeleted.html</ReplaceKeyWith>
</Redirect>
</RoutingRule>
</RoutingRules>
```

#### Example 3: Redirecionar para um erro HTTP

Suponha que, quando um objeto solicitado não for encontrado, você queira redirecionar as solicitações para uma instância do Amazon Elastic Compute Cloud (Amazon EC2). Adicione uma regra de redirecionamento para que, quando um código HTTP de status 404 (não encontrado) for retornado, o visitante do site seja redirecionado para uma instância do Amazon EC2 que processa a solicitação. O exemplo a seguir também insere o prefixo de chave de objeto `report-404/` no redirecionamento. Por exemplo, se você solicitar uma página `ExamplePage.html` e ela resultar em um erro HTTP 404, a solicitação será redirecionada a uma página `report-404/ExamplePage.html` na instância do Amazon EC2 especificada. Se não houver nenhuma regra de roteamento e o erro HTTP 404 ocorrer, o documento de erro que é especificado na configuração será retornado.

```
<RoutingRules>
    <RoutingRule>
        <Condition>
            <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals >
        </Condition>
        <Redirect>
            <HostName>ec2-11-22-333-44.compute-1.amazonaws.com</HostName>
            <ReplaceKeyPrefixWith>report-404/</ReplaceKeyPrefixWith>
        </Redirect>
    </RoutingRule>
</RoutingRules>
```

## Demonstrações de exemplo - Hospedagem de sites no Amazon S3

### Tópicos

- [Exemplo: configuração de um site estático \(p. 509\)](#)
- [Exemplo: configurar um site estático usando um domínio personalizado \(p. 511\)](#)
- [Exemplo: acelere seu site com o Amazon CloudFront \(p. 517\)](#)
- [Apagar recursos de exemplo \(p. 520\)](#)

Esta seção fornece dois exemplos. No primeiro, você configura um bucket para hospedagem de sites, faz upload de um exemplo de documento de índice e testa o site usando o endpoint de site do Amazon S3 para o bucket. O segundo exemplo mostra como você pode usar seu próprio domínio, como `example.com`, em vez do endpoint de site de bucket do S3, e exibir conteúdo de um bucket do Amazon S3 configurado como um site. O exemplo também mostra como o Amazon S3 fornece suporte a domínio raiz.

## Exemplo: configuração de um site estático

Você pode configurar um bucket do Amazon S3 para funcionar como um site. Este exemplo conduz você pelas etapas de hospedagem de um site no Amazon S3.

## Tópicos

- [Etapa 1: criação de um bucket e sua configuração como um site \(p. 510\)](#)
- [Etapa 2: adição de uma política de bucket que torna o conteúdo do bucket publicamente disponível \(p. 510\)](#)
- [Etapa 3: upload de um documento de índice \(p. 511\)](#)
- [Etapa 4: teste do site \(p. 511\)](#)

## Etapa 1: criação de um bucket e sua configuração como um site

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Crie um bucket.

Para obter instruções passo a passo, consulte [Como criar um bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

Para ver as diretrizes de nomeação de bucket, consulte [Restrições e limitações do bucket \(p. 59\)](#). Se você tiver um nome de domínio registrado, consulte [Personalizar URLs do Amazon S3 com CNAMEs \(p. 49\)](#) para obter informações adicionais sobre nomeação de buckets.

3. Abra o painel Properties (Propriedades) do bucket, escolha Static Website Hosting (Hospedagem de site estático) e faça o seguinte:
  - a. Escolha Use this bucket to host a website (Usar este bucket para hospedar um site).
  - b. No campo Index Document (Documento de índice), digite o nome do seu documento de índice. Normalmente, o nome é `index.html`.
  - c. Escolha Save (Salvar) para salvar a configuração do site.
  - d. Anote o Endpoint.

Este é o endpoint de site fornecido pelo Amazon S3 para seu bucket. Você usa esse endpoint nas seguintes etapas para testar seu site.

## Etapa 2: adição de uma política de bucket que torna o conteúdo do bucket publicamente disponível

1. No painel Properties (Propriedades) do bucket, escolha Permissions (Permissões).
2. Escolha Add Bucket Policy (Adicionar política de bucket).
3. Para hospedar um site, seu bucket deve ter acesso público de leitura. É intencional que todos tenham acesso de leitura a esse bucket. Copie e cole a política de bucket a seguir no editor de política de bucket.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "PublicReadForGetBucketObjects",  
            "Effect": "Allow",  
            "Principal": "*",  
            "Action": ["s3:GetObject"],  
            "Resource": ["arn:aws:s3:::example-bucket/*"]  
        }  
    ]  
}
```

4. Na política, substitua `example-bucket` pelo nome do seu bucket.
5. Escolha Save (Salvar).

## Etapa 3: upload de um documento de índice

1. Crie um documento. Dê a ele o mesmo nome que você deu ao documento de índice anteriormente.
2. Usando o console, faça upload do documento de índice no bucket.

Para obter instruções, consulte [Upload de objetos do S3](#) no Guia do usuário do console do Amazon Simple Storage Service.

## Etapa 4: teste do site

Digite o URL a seguir no navegador, substituindo `example-bucket` pelo nome do seu bucket e `website-region` pelo nome da região da AWS em que você implantou seu bucket. Para obter informações sobre nomes de regiões da AWS, consulte [Endpoints de site \(p. 495\)](#).

As duas formas gerais de um endpoint de site do Amazon S3 são:

`http://example-bucket.s3-website-region.amazonaws.com`

`http://example-bucket.s3-website.region.amazonaws.com`

Se seu navegador exibir a página `index.html`, o site foi implantado com sucesso.

### Note

Não há suporte para o acesso HTTPS ao site.

Agora você tem um site hospedado no Amazon S3. Esse site está disponível no endpoint de site do Amazon S3. No entanto, você pode ter um domínio, como `example.com`, que deseja usar para exibir o conteúdo do site que criou. Talvez você também queira usar o suporte ao domínio raiz do Amazon S3 para atender solicitações para `http://www.example.com` e `http://example.com`. Isso exige etapas adicionais. Para ver um exemplo, consulte [Exemplo: configurar um site estático usando um domínio personalizado \(p. 511\)](#).

## Exemplo: configurar um site estático usando um domínio personalizado

Suponha que você queira hospedar seu site estático no Amazon S3. Você registrou um domínio (por exemplo, `example.com`), e deseja que as solicitações para `http://www.example.com` e `http://example.com` sejam atendidas por seu conteúdo do Amazon S3. Se você tiver um site estático existente que deseja hospedar no Amazon S3 ou se estiver começando do zero, use este exemplo para saber como hospedar sites no Amazon S3.

### Tópicos

- [Antes de começar \(p. 512\)](#)
- [Etapa 1: Registrar um domínio \(p. 512\)](#)
- [Etapa 2: Criar e configurar buckets e fazer upload dos dados \(p. 512\)](#)
- [Etapa 3: Adicionar registros de alias para example.com e www.example.com \(p. 515\)](#)
- [Etapa 4: Testes \(p. 517\)](#)

## Antes de começar

Ao seguir as etapas deste exemplo, você trabalha com os seguintes serviços:

Amazon Route 53 – você pode usar o Route 53 para registrar domínios e definir onde você deseja rotear o tráfego de internet para o seu domínio. Explicamos como criar registros de alias do Route 53 que direcionam o tráfego para seu domínio (example.com) e subdomínio (www.example.com) para um bucket do Amazon S3 que contém um arquivo HTML.

Amazon S3 – você usa o Amazon S3 para criar buckets, fazer upload de uma página de site de exemplo, configurar permissões para que todos possam visualizar conteúdo e, em seguida, configurar os buckets para hospedagem do site.

## Etapa 1: Registrar um domínio

Se você não tiver um nome de domínio registrado, como example.com, precisará registrar um com o Route 53. Para obter mais informações, consulte [Registrar um novo domínio](#) no Guia do desenvolvedor do Amazon Route 53. Depois de registrar o nome de domínio, a próxima tarefa é criar e configurar os buckets do Amazon S3 para hospedagem do site e carregar o conteúdo do seu site.

## Etapa 2: Criar e configurar buckets e fazer upload dos dados

Para oferecer suporte a solicitações no domínio raiz, como o example.com, e no subdomínio, como o www.example.com, crie dois buckets. Um bucket contém o conteúdo. Você configura o outro bucket para redirecionar solicitações.

### Etapa 2.1: Criar dois buckets

Os nomes dos buckets devem corresponder aos nomes do site que você está hospedando. Por exemplo, para hospedar o site example.com no Amazon S3, você cria um bucket denominado example.com. Para hospedar um site sob www.example.com, você denomina o bucket www.example.com. Neste exemplo, seu site oferece suporte a solicitações de example.com e de www.example.com.

Nesta etapa, você faz login no console do Amazon S3 com as credenciais de sua conta da AWS e cria os dois buckets a seguir.

- [example.com](#)
- [www.example.com](#)

#### Note

Como no caso de domínios, os subdomínios devem ter seus próprios buckets do S3, e os buckets devem compartilhar exatamente os mesmos nomes que os subdomínios. Neste exemplo, estamos criando o subdomínio www.example.com, portanto, você também precisa um bucket do S3 denominado www.example.com.

Para criar os buckets e fazer upload do conteúdo do site para hospedagem

1. Faça login no Console de gerenciamento da AWS e abra o console da Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Crie dois buckets que correspondam ao nome do domínio e do subdomínio. Por exemplo, [example.com](#) e [www.example.com](#).

Para obter instruções passo a passo, consulte [Como criar um bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

3. Faça upload dos dados do site no bucket [example.com](#).

Você hospedará o conteúdo fora do bucket de domínio raiz ([example.com](#)), e você redirecionará as solicitações para [www.example.com](#) para o bucket de domínio raiz. Você pode armazenar conteúdo em qualquer um dos buckets. Para este exemplo, você hospedará o conteúdo no bucket [example.com](#). O conteúdo pode ser arquivos de texto, fotos de família, vídeos ou qualquer coisa que desejar. Se ainda não tiver criado um site, você precisará apenas de um arquivo para este exemplo. Você pode fazer upload de qualquer arquivo. Por exemplo, você pode criar um arquivo usando o seguinte HTML e carregá-lo no bucket. O nome do arquivo da home page de um site é geralmente index.html, mas você pode fornecer qualquer nome ao arquivo. Em uma etapa posterior, você fornecerá esse nome de arquivo como o nome do documento do índice do site.

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
    <title>My Website Home Page</title>
</head>
<body>
    <h1>Welcome to my website</h1>
    <p>Now hosted on Amazon S3!</p>
</body>
</html>
```

Para obter instruções passo a passo, consulte [Como fazer upload de um objeto em um bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

4. Para hospedar um site, seu bucket deve ter acesso público de leitura. É intencional que todos tenham acesso de leitura a esse bucket. Para conceder acesso público de leitura, anexe a seguinte política ao bucket [example.com](#), substituindo o nome do seu bucket por [example.com](#). Para obter instruções passo a passo para anexar uma política de bucket, consulte [Como adicionar uma política de bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": ["s3:GetObject"],
            "Resource": ["arn:aws:s3:::example.com/*"]
        }
    ]
}
```

Você agora tem dois buckets, [example.com](#) e [www.example.com](#), e você fez o download do conteúdo do site no bucket [example.com](#). Na próxima etapa, você configurará [www.example.com](#) para redirecionar solicitações para o bucket [example.com](#). Com o redirecionamento de objetos, você pode manter apenas uma cópia do conteúdo do site. Os visitantes que digitarem [www](#) e os que especificarem somente o domínio raiz serão roteados para o mesmo conteúdo do site no bucket [example.com](#).

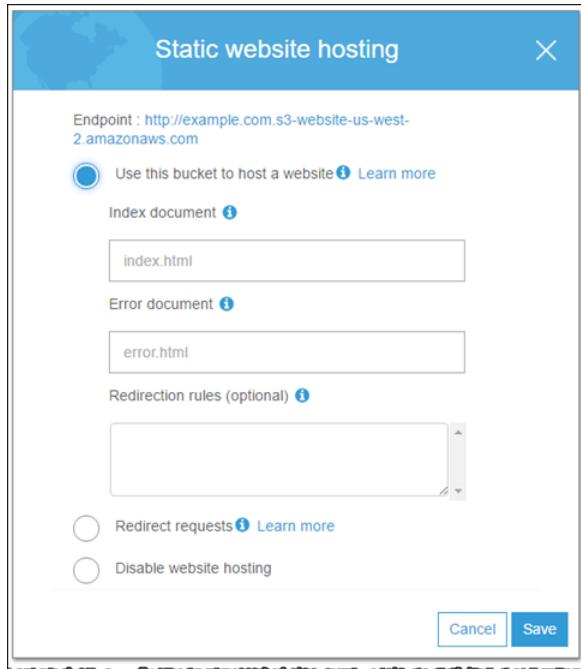
## Etapa 2.2: Configurar buckets para hospedagem de site

Quando configura um bucket para hospedagem de site, você pode acessar o site usando o endpoint do site do bucket atribuído ao Amazon S3.

Nesta etapa, você configura os dois buckets para hospedagem de site. Primeiro, você configura [example.com](#) como um site e, em seguida, configura [www.example.com](#) para redirecionar todas as solicitações para o bucket [example.com](#).

Para configurar os buckets para hospedagem de site

1. Faça login no Console de gerenciamento da AWS e abra o console da Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Na lista Bucket name (Nome do bucket), escolha o nome do bucket para o qual você deseja habilitar a hospedagem de site estático.
3. Escolha Properties (Propriedades).
4. Escolha Static website hosting (Hospedagem de sites estáticos).
5. Configure o bucket **example.com** para hospedagem de site, Na caixa Index Document (Documento do índice), digite o nome que você deu à página de índice.



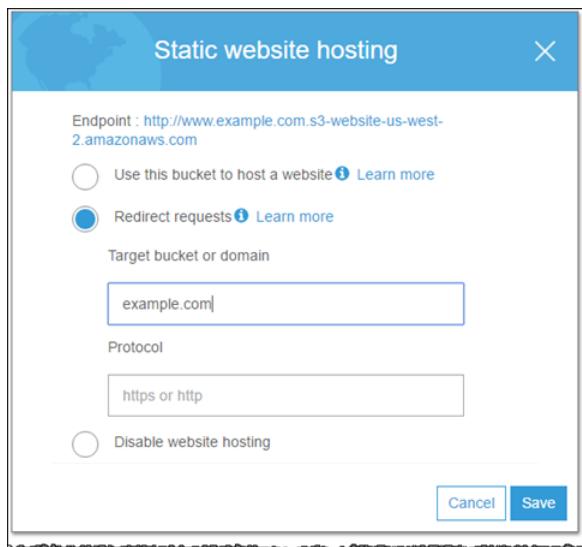
6. Escolha Save (Salvar).

### Etapa 2.3: Configurar o redirecionamento do site

Agora que você configurou o bucket para hospedagem de site, configure o bucket **www.example.com** para redirecionar todas as solicitações de **www.example.com** para **example.com**.

Para redirecionar as solicitações de **www.example.com** para **example.com**

1. No console do Amazon S3, na lista de Buckets, escolha seu bucket (**www.example.com**, neste exemplo).
2. Escolha Properties (Propriedades).
3. Escolha Static website hosting (Hospedagem de sites estáticos).
4. Escolha Redirect requests (Redirecionar solicitações). Na caixa Target bucket or domain (Bucket ou domínio de destino), digite **example.com**.
5. Escolha Save (Salvar).



## Etapa 2.4: Configurar o registro em log para o tráfego do site

Opcionalmente, você pode configurar o registro em log para acompanhar o número de visitantes que acessam o site. Para fazer isso, você habilita o registro em log para o bucket do domínio raiz. Para obter mais informações, consulte [\(Opcional\) Configurar o registro em log de tráfego da web \(p. 500\)](#).

## Etapa 2.5: Testar seu endpoint e redirecionamento

Para testar o site, digite a URL do endpoint no navegador. Sua solicitação é redirecionada e o navegador exibe o documento de índice para [example.com](http://example.com).

Na próxima etapa, você usa o Amazon Route 53 para permitir que os clientes usem todos os URLs para navegar para o site.

## Etapa 3: Adicionar registros de alias para example.com e www.example.com

Nessa etapa, você cria os registros de alias adicionados à zona hospedada de seu domínio que mapeiam [example.com](http://example.com) e [www.example.com](http://www.example.com) para os buckets correspondentes do S3. Em vez de usar endereços IP, os registros de alias usam os endpoints de site do Amazon S3. O Amazon Route 53 mantém um mapeamento entre os registros de alias e os endereços IP onde os buckets do Amazon S3 residem.

Para rotear o tráfego para o seu site

1. Abra o console do Route 53 em <https://console.aws.amazon.com/route53/>.

Note

Se você não tiver usado o Amazon Route 53, pode começar [aqui](#). Depois que você concluir a configuração, pode retomar às instruções abaixo.

2. Na lista de zonas hospedadas, escolha o nome do domínio.
3. Escolha Create Record Set (Criar conjunto de registros).

Note

Cada registro contém informações sobre como você deseja rotear o tráfego de um domínio (example.com) ou subdomínio (www.example.com). Os registros são armazenados na zona hospedada do domínio.

4. Especifique os seguintes valores:

Nome

Para o primeiro registro que você criar, aceite o valor padrão, que é o nome de sua zona hospedada e seu domínio. Isso roteará o tráfego de Internet para o bucket que tem o mesmo nome que o seu domínio.

Repita esta etapa para criar um segundo registro para seu subdomínio. No segundo registro, digite www. Isso roteará o tráfego da Internet para o bucket www.*example.com*.

Tipo

Escolha A – IPv4 address (A – endereço IPv4).

Alias

Escolha Sim.

Alvo do alias

Digite o nome de seu endpoint de bucket do Amazon S3, por exemplo, *example.com* (s3-website-us-west-2).

Note

Especifique o mesmo valor para Alias Target (Destino do alias) para os dois registros. O Route 53 descobre para qual bucket rotear o tráfego com base no nome do registro.

Política de roteamento

Aceite o valor padrão de Simples.

Avaliar status do alvo

Aceite o valor padrão de Não.

5. Escolha Create (Criar).
6. Para www.*example.com*, repita as etapas 4 a 6 para criar um registro.

A captura de tela a seguir mostra o registro do alias de *example.com* como uma ilustração. Você também precisa criar um registro de alias para www.*example.com*.

The screenshot shows the AWS Route 53 console. On the left, there's a navigation pane with options like Dashboard, Hosted zones, Traffic flow, Traffic policies, Policy records, Domains, Registered domains, and Pending requests. The 'Hosted zones' option is selected. The main area displays a table of record sets for a specific zone. The columns are 'Name', 'Type', and 'Value'. There are two entries:

Name	Type	Value
example.com.	NS	ns-165.awsdns-20.com. ns-1897.awsdns-45.co.uk. ns-1026.awsdns-00.org. ns-783.awsdns-33.net.
example.com.	SOA	ns-165.awsdns-20.com. awsdns-hostmaster.amazon. ns-165.awsdns-20.com. awsdns-22.com. awsdns-23.com. awsdns-24.com.

On the right side, there are several panels: 'Create Record Set' (with fields for Name, Type, and Alias Target), 'Routing Policies' (listing CloudFront, Elastic Load Balancing, S3 website endpoint, and Resource record set), and 'Evaluate Target Health' and 'Health Checks' buttons.

#### Note

A criação, a alteração e a exclusão de conjuntos de registros de recursos demoram para serem propagadas para os servidores DNS do Route 53. As alterações são geralmente propagadas para todos os servidores de nome do Route 53 em alguns minutos. Em raras circunstâncias, a propagação pode levar até 30 minutos.

## Etapa 4: Testes

Para verificar se o site está funcionando corretamente, em seu navegador, teste as seguintes URLs:

- <http://example.com> – exibe o documento de índice no bucket `example.com`.
- <http://www.example.com> – redireciona a solicitação para <http://example.com>.

Em alguns casos, você pode precisar limpar o cache do navegador da web para ver o comportamento esperado.

## Exemplo: acelere seu site com o Amazon CloudFront

É possível usar [Amazon CloudFront](#) para melhorar o desempenho de seu site. O CloudFront torna os arquivos de seu site (como HTML, imagens e vídeo) disponíveis a partir de datacenters ao redor do mundo (denominados pontos de presença). Quando um visitante solicita um arquivo em seu site, o CloudFront redireciona automaticamente a solicitação para uma cópia do arquivo no ponto de presença mais próximo. Isso resulta em tempos de download mais rápidos se o visitante tiver solicitado o conteúdo em um datacenter localizado mais longe.

O CloudFront armazena em cache o conteúdo em pontos de presença por um período especificado por você. Se um visitante solicitar conteúdo que foi armazenado em cache por mais tempo que a data

de expiração, o CloudFront verificará o servidor de origem para saber se há uma versão mais nova do conteúdo disponível. Se houver uma versão mais nova à disposição, o CloudFront copiará a nova versão para o ponto de presença. As alterações feitas no conteúdo original são replicadas para pontos de presença à medida que os visitantes solicitam o conteúdo.

Para agilizar o site, use o CloudFront para concluir as tarefas a seguir.

#### Tarefas

- [Criar uma distribuição do CloudFront \(p. 518\)](#)
- [Atualizar os conjuntos de registros do domínio e do subdomínio \(p. 519\)](#)
- [\(Opcional\) Verificar os arquivos de log \(p. 519\)](#)

## Criar uma distribuição do CloudFront

Primeiro, você cria uma distribuição do CloudFront. Isso torna seu site disponível em datacenters em todo o mundo.

Para criar uma distribuição com uma origem do Amazon S3

1. Abra o console do CloudFront em <https://console.aws.amazon.com/cloudfront/>.
2. Escolha Create Distribution (Criar distribuição).
3. Na página Select a delivery method for your content (Selecionar um método de entrega para seu conteúdo), para Web, escolha Get Started (Conceitos básicos).
4. Na página Create Distribution (Criar distribuição), na seção Origin Settings (Configurações de origem), em Origin Domain Name (Nome do domínio de origem), digite o endpoint de hospedagem de site estático do Amazon S3 para o bucket. Por exemplo, `example.com.s3.amazonaws.com`.  
O CloudFront preenche o Origin ID (ID de origem) para você.
5. Em Default Cache Behavior Settings (Configurações do comportamento de cache padrão), deixe os valores padrão definidos. Para obter mais informações sobre essas opções de configuração, consulte [Valores que você especifica quando cria ou atualiza uma distribuição da Web](#) no Guia do desenvolvedor do Amazon CloudFront.
6. Para Distribution Settings (Configurações de distribuição), faça o seguinte:
  - a. Deixe Price Class (Classe de preço) definida como Use All Edge Locations (Best Performance) (Usar todos os pontos de presença [melhor desempenho]).
  - b. Defina Alternate Domain Names (CNAMEs) (Nomes de domínio alternativos [CNAMEs]) como o domínio raiz e o subdomínio `www`; neste tutorial, eles são `example.com` e `www.example.com`. Esses valores devem ser definidos antes de você criar aliases para os registros A que conectam os nomes de domínio especificados à distribuição do CloudFront.
  - c. Defina Default Root Object (Objeto raiz padrão) como `index.html`. Esta é a página padrão que a distribuição do CloudFront retornará se a URL usada para acessar a distribuição não contiver um nome de arquivo. Este valor deve corresponder ao valor do documento `index` definido em [Configuração de bucket para hospedagem de site \(p. 496\)](#).
  - d. Defina Logging (Registro em log) como On (Ligado).
  - e. Em Bucket for Logs (Bucket para logs), escolha o bucket para o registro em log que você criou.
  - f. Para armazenar os logs gerados pelo tráfego para a distribuição do CloudFront em uma pasta chamada `cfn`, no bucket de log, digite `cfn/` em Log Prefix (Prefixo do log).
  - g. Deixe as outras configurações nos valores padrão.
7. Escolha Create Distribution (Criar distribuição).

Para ver o status atual da distribuição, localize a distribuição no console e verifique a coluna Status. Um status `InProgress` indica que a distribuição ainda não foi totalmente implantada.

Depois que a distribuição estiver implantada, você pode fazer referência ao conteúdo com o novo nome do domínio do CloudFront. Registre o valor do Domain Name (Nome do domínio) mostrado no console do CloudFront. Você precisará dele na próxima etapa. Neste exemplo, o valor é `dj4p1rv6mvubz.cloudfront.net`.

Para verificar se a distribuição do CloudFront está funcionando, digite o nome de domínio da distribuição em um navegador da Web. Se estiver funcionando, seu site estará visível.

## Atualizar os conjuntos de registros do domínio e do subdomínio

Agora que você criou uma distribuição do CloudFront com êxito, atualize os registros A no Route 53 para apontarem para a nova distribuição do CloudFront.

Para atualizar os registros A a fim de apontar para uma distribuição do CloudFront

1. Abra o console do Route 53 em <https://console.aws.amazon.com/route53/>.
2. Na página Hosted Zones (Zonas hospedadas), escolha a zona hospedada criada por você para o domínio.
3. Escolha Go to Record Sets (Ir para conjuntos de registros).
4. Escolha o registro A criado por você para o subdomínio `www`.
5. Para Alias Target (Destino do alias), escolha a distribuição do CloudFront.
6. Escolha Save Record Set (Salvar conjunto de registros).
7. Para redirecionar o registro A do domínio raiz para a distribuição do CloudFront, repita esse procedimento.

A atualização feita nos conjuntos de registros entrará em vigor de 2 a 48 horas depois. Para saber se os novos registros A entraram em vigor, em um navegador da Web, digite `http://www.example.com`. Se o navegador deixar de redirecionar você para `http://example.com`, os novos registros A estarão implantados.

Essa mudança no comportamento ocorre porque o tráfego roteado pelo registro A anterior para o bucket S3 do subdomínio `www` é redirecionado pelas configurações no Amazon S3 para o domínio raiz. Quando o novo registro A entra em vigor, o tráfego roteado pelo novo registro A para a distribuição do CloudFront não é redirecionado para o domínio raiz.

### Tip

Os navegadores podem armazenar em cache configurações de redirecionamento. Se você acreditar que as novas configurações do registro A devem ter entrado em vigor, mas o navegador ainda redirecionar `http://www.example.com` para `http://example.com`, tente limpar o histórico e limpar o cache do navegador, fechando e reabrindo o aplicativo do navegador ou usando outro navegador da Web.

Quando os novos registros A entram em vigor, qualquer visitante que faça referência ao site usando `http://example.com` ou `http://www.example.com` é redirecionado para o ponto de presença do CloudFront mais próximo, onde ele aproveita tempos de download menores.

Se tiver criado o site apenas como um exercício de aprendizado, você poderá excluir os recursos alocações, de maneira que deixe de acumular cobranças. Para isso, vá para [Apagar recursos de exemplo \(p. 520\)](#). Depois que você excluir os recursos da AWS, o site deixará de estar disponível.

## (Opcional) Verificar os arquivos de log

Os logs de acesso informam quantas pessoas estão visitando o site. Eles também contêm dados comerciais valiosos que você pode analisar com outros serviços, como o [Amazon EMR](#).

No bucket, os arquivos de log mais antigos do Amazon S3 estão localizados na pasta `root`. Todos os novos arquivos de log, que devem ser logs do CloudFront, estão localizados na pasta `cfn`. O Amazon S3 grava os logs de acesso ao site em seu bucket de logs a cada duas horas. O CloudFront grava logs em seu bucket de logs em até 24 horas após a realização das solicitações correspondentes.

Para ver os arquivos de log do site

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Escolha o bucket de registro em log do site.
3. Para ver os arquivos de log armazenados na pasta `cfn` ou `root`, escolha `cfn` ou `root`.
4. Abra os arquivos de log do Amazon S3, que são arquivos de texto, em um navegador. Faça download dos arquivos `.gzip` gravados pelo CloudFront antes de abri-los.

## Apagar recursos de exemplo

Se você tiver criado o site estático apenas como um exercício de aprendizado, não se esqueça de excluir os recursos da AWS alocados, de maneira que deixe de acumular cobranças. Depois que você excluir os recursos da AWS, o site deixará de estar disponível.

Tarefas

- [Excluir a distribuição do Amazon CloudFront \(p. 520\)](#)
- [Excluir a zona hospedada do Route 53 \(p. 520\)](#)
- [Excluir o S3 Bucket \(p. 521\)](#)

## Excluir a distribuição do Amazon CloudFront

Antes de excluir uma distribuição do Amazon CloudFront, você deverá desabilitá-la. Uma distribuição desabilitada deixa de ser funcional e não acumula encargos. É possível habilitar uma distribuição desabilitada a qualquer momento. Depois que você excluir uma distribuição desabilitada, ela deixará de estar disponível.

Para desabilitar e excluir uma distribuição do CloudFront

1. Abra o console do CloudFront em <https://console.aws.amazon.com/cloudfront/>.
2. Selecione a distribuição que você deseja desabilitar e escolha Disable (Desabilitar).
3. Quando a confirmação for solicitada, escolha Yes, Disable (Sim, desabilitar).
4. Selecione a distribuição desabilitada e escolha Delete (Excluir).
5. Quando a confirmação for solicitada, escolha Yes, Delete (Sim, excluir).

## Excluir a zona hospedada do Route 53

Para excluir a zona hospedada, você deve excluir os conjuntos de registros criados. Você não precisa excluir os registros de Start of Authority (SOA – Início da autoridade) e Name Server (NS – Servidor de nomes); eles são excluídos automaticamente quando se exclui a zona hospedada.

Para excluir os conjuntos de registros

1. Abra o console do Route 53 em <https://console.aws.amazon.com/route53/>.
2. Na lista de nomes de domínio, selecione o nome do seu domínio e escolha Go to Record Sets (Ir para conjuntos de registros).

3. Na lista de conjuntos de registros, selecione os registros A que você criou. O tipo de cada conjunto de registros está listado na coluna Type (Tipo).
4. Escolha Delete Record Set (Excluir conjunto de registros).
5. Quando a confirmação for solicitada, escolha Confirm (Confirmar).

Para excluir uma zona hospedada do Route 53

1. Continuando o procedimento anterior, escolha Back to Hosted Zones (Voltar para zonas hospedadas).
2. Selecione o nome do seu domínio e escolha Delete Hosted Zone (Excluir zona hospedada).
3. Quando a confirmação for solicitada, escolha Confirm (Confirmar).

## Excluir o S3 Bucket

Antes de excluir o bucket do S3, verifique se o registro está desativado para o bucket. Caso contrário, a AWS continuará gravando logs para o bucket à medida que você o excluir.

Para desabilitar o registro em log para um bucket

1. Abra o console do Amazon S3 em <https://console.aws.amazon.com/s3/>.
2. Selecione o bucket e escolha Properties (Propriedades).
3. Em Properties (Propriedades), escolha Logging (Registro).
4. Desmarque a caixa de seleção Enabled (Habilitado).
5. Escolha Save (Salvar).

Agora, você pode excluir seu bucket. Para obter mais informações, consulte [Como faço para excluir um bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

# Configurar notificações de evento do Amazon S3

O recurso de notificação do Amazon S3 permite receber notificações quando certos eventos ocorrem no bucket. Para habilitar notificações, você deve primeiro adicionar uma configuração de notificação que identifique os eventos que deseja que o Amazon S3 publique, e os destinos para os quais deseja que o Amazon S3 envie as notificações de evento. Você armazena essa configuração no sub-recurso notificação (consulte [Opções de configuração de bucket \(p. 57\)](#)) associado a um bucket. O Amazon S3 fornece uma API para o gerenciamento desse sub-recurso.

## Tópicos

- [Visão geral \(p. 522\)](#)
- [Como habilitar notificações de evento \(p. 524\)](#)
- [Tipos e destinos de notificações de evento \(p. 525\)](#)
- [Configurar notificações com filtragem de nomes de chaves de objetos \(p. 527\)](#)
- [Conceder permissões para publicar mensagens de notificação de evento a um destino \(p. 531\)](#)
- [Passo a passo do exemplo 1: configurar um bucket para notificações \(destino da mensagem: tópico do SNS e fila do SQS\) \(p. 533\)](#)
- [Passo a passo do exemplo 2: configurar um bucket para notificações \(destino da mensagem: AWS Lambda\) \(p. 539\)](#)
- [Estrutura de mensagens de evento \(p. 539\)](#)

## Visão geral

Atualmente, o Amazon S3 pode publicar notificações para os seguintes eventos:

- Evento de um novo objeto criado — o Amazon S3 oferece suporte a várias APIs para criação de objetos. Você pode solicitar uma notificação apenas quando uma API específica for usada (por exemplo, `s3:ObjectCreated:Put`) ou pode usar um caractere curinga (por exemplo, `s3:ObjectCreated:*`) para solicitar uma notificação quando um objeto for criado independentemente da API usada.
- Evento de remoção de um objeto — o Amazon S3 oferece suporte a exclusões de objetos com e sem versionamento. Para obter informações sobre versionamento de objetos, consulte [Versionamento de objeto \(p. 111\)](#) e [Usar versionamento \(p. 448\)](#).

Você pode solicitar uma notificação quando um objeto for excluído ou quando um objeto com versionamento for excluído permanentemente usando o tipo de evento `s3:ObjectRemoved:Delete`. Ou você pode solicitar uma notificação quando um marcador de exclusão for criado para um objeto com versionamento usando `s3:ObjectRemoved:DeleteMarkerCreated`. Você também pode usar um caractere curinga `s3:ObjectRemoved:*` para solicitar uma notificação sempre que um objeto for excluído. Para obter informações sobre como excluir objetos com versionamento, consulte [Exclusão de versões de objeto \(p. 461\)](#).

- Restaurar eventos de objeto — O Amazon S3 dá suporte à restauração de objetos arquivados na classe de armazenamento GLACIER. Você solicita ser notificado da conclusão da restauração do objeto

usando `s3:ObjectRestore:Completed`. Você usa `s3:ObjectRestore:Post` para solicitar a notificação do início de uma restauração.

- Evento de perda de um objeto Reduced Redundancy Storage (RRS) — o Amazon S3 envia uma mensagem de notificação quando detecta que um objeto da classe de armazenamento RRS foi perdido.

Para obter uma lista dos tipos de evento com suporte, consulte [Tipos de evento compatíveis \(p. 525\)](#).

O Amazon S3 oferece suporte aos seguintes destinos onde pode publicar eventos:

- Tópico do Amazon Simple Notification Service (Amazon SNS)

O Amazon SNS é um serviço de mensagens por push flexível, totalmente gerenciado. Usando esse serviço, você pode enviar mensagens por push a dispositivos móveis ou serviços distribuídos. Com o SNS você pode publicar uma mensagem uma vez e entregá-la uma ou mais vezes. Um tópico do SNS é um ponto de acesso ao qual os destinatários podem se inscrever dinamicamente para receber notificações de evento. Para obter mais informações sobre o SNS, consulte a página de detalhes do produto [Amazon SNS](#).

- Fila do Amazon Simple Queue Service (Amazon SQS)

O Amazon SQS é um serviço de fila de mensagens confiável, dimensionável e totalmente gerenciado. Você pode usar o SQS para transmitir qualquer volume de dados sem exigir que outros serviços estejam sempre disponíveis. Na configuração de notificação você pode solicitar que o Amazon S3 publique eventos em uma fila do SQS. Para obter mais informações sobre o SQS, consulte a página de detalhes do produto [Amazon SQS](#).

- AWS Lambda

O AWS Lambda é um serviço de computação que facilita a criação de aplicativos que respondam rapidamente a novas informações. O AWS Lambda executa seu código em resposta a eventos, como uploads de imagens, atividades do aplicativo, cliques de sites ou saídas de dispositivos conectados. Você pode usar o AWS Lambda para estender outros serviços da AWS com lógica personalizada ou criar seu próprio back-end que opere na escala, desempenho e segurança da AWS. Com o AWS Lambda, você pode criar facilmente aplicativos discretos controlados por eventos que executam apenas quando necessário e dimensionam automaticamente de algumas solicitações por dia para milhares por segundo.

O AWS Lambda pode executar código personalizado em resposta a eventos de buckets do Amazon S3. Você carrega o código personalizado no AWS Lambda e cria o que é chamado de uma função do Lambda. Quando o Amazon S3 detecta um evento de um tipo específico (por exemplo, um evento de objeto criado), ele pode publicar o evento no AWS Lambda e invocar sua função no Lambda. Em resposta, o AWS Lambda executa a função. Para obter mais informações, consulte a página de detalhes do produto [AWS Lambda](#).

As seções a seguir oferecem mais detalhes sobre como habilitar notificações de evento em um bucket. Os subtópicos também fornecem instruções passo a passo para ajudá-lo a explorar o recurso de notificação.

- [Passo a passo do exemplo 1: configurar um bucket para notificações \(destino da mensagem: tópico do SNS e fila do SQS\) \(p. 533\)](#)
- [Passo a passo do exemplo 2: configurar um bucket para notificações \(destino da mensagem: AWS Lambda\) \(p. 539\)](#)

# Como habilitar notificações de evento

Habilitar notificações é uma operação em nível de bucket. Ou seja, você armazena informações de configuração de notificação no sub-recurso notificação associado a um bucket. Você pode usar qualquer um dos métodos a seguir para gerenciar a configuração de notificação:

- Usar o console do Amazon S3

A interface do usuário do console permite definir uma configuração de notificação em um bucket sem necessidade de escrever nenhum código. Para obter instruções, consulte [Como habilitar e configurar notificações de evento para um bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

- Usar os AWS SDKs de maneira programática

## Note

Se necessário, você também pode chamar a API REST do Amazon S3 diretamente do seu código. Contudo, isso pode ser complicado, porque exige que você grave código para autenticar suas solicitações.

Internamente, o console e os SDKs chamam a API REST do Amazon S3 para gerenciar sub-recursos de notificação associados ao bucket. Para obter exemplos da configuração de notificação usando o AWS SDK, consulte o link para as instruções passo a passo na seção anterior.

Independentemente do método usado, o Amazon S3 armazena a configuração de notificação como XML no sub-recurso notificação associado a um bucket. Para obter informações sobre sub-recursos de bucket, consulte [Opções de configuração de bucket \(p. 57\)](#). Por padrão, as notificações não estão habilitadas para nenhum tipo de evento. Portanto, no início, o sub-recurso notificação armazena uma configuração vazia.

```
<NotificationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
</NotificationConfiguration>
```

Para habilitar notificações para eventos de tipos específicos, você substitui o XML pela configuração apropriada que identifica os tipos de evento que deseja que o Amazon S3 publique e o destino onde deseja que os eventos sejam publicados. Para cada destino, você adiciona uma configuração correspondente de XML. Por exemplo:

- Publicar mensagens de eventos em uma fila do SQS — para definir uma fila do SQS como o destino das notificações de um ou mais tipos de evento, você adiciona o `QueueConfiguration`.

```
<NotificationConfiguration>
  <QueueConfiguration>
    <Id>optional-id-string</Id>
    <Queue>sqs-queue-arn</Queue>
    <Event>event-type</Event>
    <Event>event-type</Event>
    ...
  </QueueConfiguration>
  ...
</NotificationConfiguration>
```

- Publicar mensagens de eventos em um tópico do SNS — para definir um tópico do SNS como o destino das notificações para tipos específicos de evento, você adiciona o `TopicConfiguration`.

```
<NotificationConfiguration>
  <TopicConfiguration>
    <Id>optional-id-string</Id>
    <Topic>sns-topic-arn</Topic>
```

```
<Event>event-type</Event>
<Event>event-type</Event>
...
</TopicConfiguration>
...
</NotificationConfiguration>
```

- Invoque a função do AWS Lambda e forneça uma mensagem de evento como argumento — para definir uma função do Lambda como o destino de notificação para tipos específicos de evento, adicione o CloudFunctionConfiguration.

```
<NotificationConfiguration>
  <CloudFunctionConfiguration>
    <Id>optional-id-string</Id>
    <Cloudcode>cloud-function-arn</Cloudcode>
    <Event>event-type</Event>
    <Event>event-type</Event>
    ...
  </CloudFunctionConfiguration>
  ...
</NotificationConfiguration>
```

Para remover todas as notificações configuradas em um bucket, você salva um elemento `<NotificationConfiguration/>` vazio no sub-recurso notificação.

Quando o Amazon S3 detecta um evento do tipo específico, ele publica uma mensagem com as informações do evento. Para obter mais informações, consulte [Estrutura de mensagens de evento \(p. 539\)](#).

## Tipos e destinos de notificações de evento

Esta seção descreve os tipos de notificação de evento que são compatíveis com o Amazon S3 e os tipos de destino onde as notificações podem ser publicadas.

### Tipos de evento compatíveis

O Amazon S3 pode publicar eventos dos seguintes tipos. Você especifica esses tipos de evento na configuração de notificação.

Tipos de evento	Descrição
s3:ObjectCreated: <sup>*</sup>	As APIs do Amazon S3, como PUT, POST e COPY, podem criar um objeto. Com esses tipos de evento, você pode habilitar uma notificação quando um objeto é criado usando uma API específica ou pode usar o tipo de evento s3:ObjectCreated: <sup>*</sup> para solicitar notificação independentemente da API usada para criar um objeto.
s3:ObjectCreated:Put	
s3:ObjectCreated:Post	
s3:ObjectCreated:Copy	
s3:ObjectCreated:CompleteMultipartUpload	Você não receberá notificações de evento de operações que falharam.
s3:ObjectRemoved: <sup>*</sup>	Usando os tipos de evento ObjectRemoved, você pode habilitar a notificação quando um objeto ou um lote de objetos é removido de um bucket.
s3:ObjectRemoved:Delete	
s3:ObjectRemoved:DeleteMarkerCreated	

Tipos de evento	Descrição
	<p>Você pode solicitar uma notificação quando um objeto for excluído ou quando um objeto com versionamento for excluído permanentemente usando o tipo de evento <code>s3:ObjectRemoved:Delete</code>. Ou você pode solicitar uma notificação quando um marcador de exclusão for criado para um objeto com versionamento usando <code>s3:ObjectRemoved:DeleteMarkerCreated</code>. Para obter informações sobre como excluir objetos com versionamento, consulte <a href="#">Exclusão de versões de objeto (p. 461)</a>. Você também pode usar um caractere curinga <code>s3:ObjectRemoved:*</code> para solicitar uma notificação sempre que um objeto for excluído.</p> <p>Você não receberá notificações de evento de exclusões automáticas de políticas de ciclo de vida ou de operações com falha.</p>
<code>s3:ObjectRestore:Post</code> <code>s3:ObjectRestore:Completed</code>	<p>Usando tipos de evento do objeto de restauração você pode receber notificações de início e conclusão durante a restauração de objetos na classe de armazenamento GLACIER.</p> <p>Você pode usar o <code>s3:ObjectRestore:Post</code> para solicitar uma notificação do início da restauração do objeto. Você pode usar o <code>s3:ObjectRestore:Completed</code> para solicitar uma notificação da conclusão de restauração.</p>
<code>s3:ReducedRedundancyLostObject</code>	<p>Você pode usar esse tipo de evento para solicitar que o Amazon S3 envie uma mensagem de notificação quando o Amazon S3 detectar que um objeto da classe de armazenamento RRS for perdido.</p>

## Destinos compatíveis

O Amazon S3 pode enviar mensagens de notificação de evento aos seguintes destinos. Você especifica o valor do ARN desses destinos na configuração de notificação.

- Publicar mensagens de evento em um tópico do Amazon Simple Notification Service (Amazon SNS)
- Publicar mensagens de evento em uma fila do Amazon Simple Queue Service (Amazon SQS)

**Note**

Se a fila de destino estiver habilitada para SSE, o Amazon S3 precisará ter acesso à chave do KMS associada para habilitar a criptografia de mensagens.

- Publicar mensagens de eventos no AWS Lambda invocando uma função do Lambda e fornecendo a mensagem de evento como um argumento

Você deve conceder permissões ao Amazon S3 para postar mensagens em um tópico do Amazon SNS ou em uma fila do Amazon SQS. Você também deve conceder permissão ao Amazon S3 para invocar uma função do AWS Lambda em seu nome. Para obter informações sobre como conceder essas permissões, consulte [Conceder permissões para publicar mensagens de notificação de evento a um destino \(p. 531\)](#).

# Configurar notificações com filtragem de nomes de chaves de objetos

Você pode configurar as notificações para serem filtradas pelo prefixo e pelo sufixo do nome da chave de objetos. Por exemplo, você pode definir uma configuração para que uma notificação seja enviada a você apenas quando arquivos de imagem com uma extensão ".jpg" forem adicionados a um bucket. Ou ter uma configuração que entregue uma notificação a um tópico do Amazon SNS quando um objeto com o prefixo "images" for adicionado ao bucket, e que notificações para objetos com um prefixo "logs/" no mesmo bucket sejam entregues a uma função do AWS Lambda.

Você pode definir configurações de notificação que usem filtragem de nomes de chave de objeto no console do Amazon S3 usando APIs do Amazon S3 por meio dos SDKs da AWS ou das APIs REST diretamente. Para obter informações sobre como usar a interface do usuário do console para definir uma configuração de notificação em um bucket, consulte [Como habilitar e configurar notificações de evento para um bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

O Amazon S3 armazena a configuração de notificação como XML no sub-recurso notificação associado a um bucket, conforme descrito em [Como habilitar notificações de evento \(p. 524\)](#). Você usa a estrutura XML do `Filter` para definir regras para que as notificações sejam filtradas pelo prefixo e/ou pelo sufixo do nome da chave de um objeto. Para obter informações sobre os detalhes da estrutura XML do `Filter`, consulte [Notificação de bucket de PUT](#) no Amazon Simple Storage Service API Reference.

Configurações de notificação que usam o `Filter` não podem definir regras de filtragem com prefixes sobrepostos, sufíxos sobrepostos ou prefixes e sufíxos sobrepostos. As seções a seguir apresentam exemplos de configurações válidas de notificação com filtragem de nome de chave de objeto e exemplos de configurações de notificação que são inválidas devido à sobreposição de prefixes/sufixos.

## Exemplos de configurações válidas de notificação com filtragem de nome de chave de objeto

A configuração de notificação a seguir contém uma configuração de fila que identifica uma fila do Amazon SQS para a qual o Amazon S3 publica eventos do tipo `s3:ObjectCreated:Put`. Os eventos serão publicados sempre que um objeto que tenha um prefixo `images/` e um sufixo `.jpg` seja `PUT` em um bucket.

```
<NotificationConfiguration>
  <QueueConfiguration>
    <Id>1</Id>
    <Filter>
      <S3Key>
        <FilterRule>
          <Name>prefix</Name>
          <Value>images/</Value>
        </FilterRule>
        <FilterRule>
          <Name>suffix</Name>
          <Value>.jpg</Value>
        </FilterRule>
      </S3Key>
    </Filter>
    <Queue>arn:aws:sqs:us-west-2:444455556666:s3notificationqueue</Queue>
    <Event>s3:ObjectCreated:Put</Event>
  </QueueConfiguration>
</NotificationConfiguration>
```

A configuração de notificação a seguir tem vários prefixos não sobrepostos. A configuração define que as notificações para solicitações PUT na pasta `images/` irão para queue-A enquanto as notificações para solicitações PUT na pasta `logs/` irão para queue-B.

```
<NotificationConfiguration>
<QueueConfiguration>
  <Id>1</Id>
  <Filter>
    <S3Key>
      <FilterRule>
        <Name>prefix</Name>
        <Value>images/</Value>
      </FilterRule>
    </S3Key>
  </Filter>
  <Queue>arn:aws:sqs:us-west-2:444455556666:sqs-queue-A</Queue>
  <Event>s3:ObjectCreated:Put</Event>
</QueueConfiguration>
<QueueConfiguration>
  <Id>2</Id>
  <Filter>
    <S3Key>
      <FilterRule>
        <Name>prefix</Name>
        <Value>logs/</Value>
      </FilterRule>
    </S3Key>
  </Filter>
  <Queue>arn:aws:sqs:us-west-2:444455556666:sqs-queue-B</Queue>
  <Event>s3:ObjectCreated:Put</Event>
</QueueConfiguration>
</NotificationConfiguration>
```

A configuração de notificação a seguir tem vários sufixos não sobrepostos. A configuração define que todas as imagens `.jpg` recém-adicionadas ao bucket sejam processadas pela função A de nuvem do Lambda e todas as imagens `.png` recém-adicionadas sejam processadas pela função B de nuvem. Os sufixos `.png` e `.jpg` não estão sendo sobrepostos, embora tenham a mesma última letra. Dois sufixos são considerados sobrepostos se uma determinada sequência puder terminar com ambos os sufixos. Uma sequência não pode terminar com `.png` e com `.jpg`, portanto, os sufixos na configuração de exemplo não são sufixos sobrepostos.

```
<NotificationConfiguration>
<CloudFunctionConfiguration>
  <Id>1</Id>
  <Filter>
    <S3Key>
      <FilterRule>
        <Name>suffix</Name>
        <Value>.jpg</Value>
      </FilterRule>
    </S3Key>
  </Filter>
  <Cloudcode>arn:aws:lambda:us-west-2:444455556666:cloud-function-A</Cloudcode>
  <Event>s3:ObjectCreated:Put</Event>
</CloudFunctionConfiguration>
<CloudFunctionConfiguration>
  <Id>2</Id>
  <Filter>
    <S3Key>
      <FilterRule>
        <Name>suffix</Name>
        <Value>.png</Value>
      </FilterRule>
    </S3Key>
  </Filter>
```

```
</S3Key>
</Filter>
<Cloudcode>arn:aws:lambda:us-west-2:444455556666:cloud-function-B</Cloudcode>
<Event>s3:ObjectCreated:Put</Event>
</CloudFunctionConfiguration>
</NotificationConfiguration>
```

As configurações de notificação que usam o `Filter` não podem definir regras de filtragem com prefixos sobrepostos para os mesmos tipos de evento, a menos que os prefixos sobrepostos sejam usados com suffixos não sobrepostos. A configuração de exemplo a seguir mostra como objetos criados com um prefixo comum, mas com suffixos não sobrepostos, podem ser entregues a destinos diferentes.

```
<NotificationConfiguration>
<CloudFunctionConfiguration>
<Id>1</Id>
<Filter>
<S3Key>
<FilterRule>
<Name>prefix</Name>
<Value>images</Value>
</FilterRule>
<FilterRule>
<Name>suffix</Name>
<Value>.jpg</Value>
</FilterRule>
</S3Key>
</Filter>
<Cloudcode>arn:aws:lambda:us-west-2:444455556666:cloud-function-A</Cloudcode>
<Event>s3:ObjectCreated:Put</Event>
</CloudFunctionConfiguration>
<CloudFunctionConfiguration>
<Id>2</Id>
<Filter>
<S3Key>
<FilterRule>
<Name>prefix</Name>
<Value>images</Value>
</FilterRule>
<FilterRule>
<Name>suffix</Name>
<Value>.png</Value>
</FilterRule>
</S3Key>
</Filter>
<Cloudcode>arn:aws:lambda:us-west-2:444455556666:cloud-function-B</Cloudcode>
<Event>s3:ObjectCreated:Put</Event>
</CloudFunctionConfiguration>
</NotificationConfiguration>
```

## Exemplos de configurações de notificação com sobreposição inválida de prefixo/sufixo

As configurações de notificação que usam o `Filter`, na maior parte, não podem definir regras de filtragem com prefixos sobrepostos, suffixos sobrepostos ou combinações sobrepostas de prefixos e suffixos para os mesmos tipos de evento. (Você pode ter prefixos sobrepostos desde que os suffixos não se sobreponham. Para ver um exemplo, consulte [Configurar notificações com filtragem de nomes de chaves de objetos \(p. 527\)](#).)

Você pode usar filtros de nomes de chave de objetos sobrepostos com diferentes tipos de evento. Por exemplo, você pode criar uma configuração de notificação que use o prefixo `image/` para o tipo de evento `ObjectCreated:Put` e o prefixo `image/` para o tipo de evento `ObjectDeleted:*`.

Você receberá um erro se tentar salvar uma configuração de notificação que tenha filtros de nomes sobrepostos inválidos para os mesmos tipos de evento, ao usar o console do AWS Amazon S3 ou ao usar a API do Amazon S3. Esta seção mostra exemplos de configurações de notificação que são inválidas devido aos filtros de nomes sobrepostos.

Presume-se que qualquer regra de configuração de notificação tenha um prefixo e um sufixo padrão que correspondam a qualquer outro prefixo e um sufixo respectivamente. A configuração de notificação a seguir é inválida porque tem prefixos sobrepostos, em que o prefixo raiz sobrepõe qualquer outro prefixo. (A mesma coisa seria verdadeira se usássemos um sufixo em vez de um prefixo neste exemplo. O sufixo raiz sobrepõe qualquer outro sufixo.)

```
<NotificationConfiguration>
    <TopicConfiguration>
        <Topic>arn:aws:sns:us-west-2:444455556666:sns-notification-one</Topic>
        <Event>s3:ObjectCreated:*</Event>
    </TopicConfiguration>
    <TopicConfiguration>
        <Topic>arn:aws:sns:us-west-2:444455556666:sns-notification-two</Topic>
        <Event>s3:ObjectCreated:*</Event>
        <Filter>
            <S3Key>
                <FilterRule>
                    <Name>prefix</Name>
                    <Value>images</Value>
                </FilterRule>
            </S3Key>
        </Filter>
    </TopicConfiguration>
</NotificationConfiguration>
```

A configuração de notificação a seguir é inválida porque tem sufixos sobrepostos. Dois sufixos são considerados sobrepostos se uma determinada sequência puder terminar com ambos os sufixos. Uma sequência pode terminar com `jpg` e `pg`, portanto, os sufixos estão sobrepostos. (O mesmo é verdadeiro para prefixos, dois prefixos serão considerados como sobrepostos se uma determinada sequência puder começar com os dois prefixos.)

```
<NotificationConfiguration>
    <TopicConfiguration>
        <Topic>arn:aws:sns:us-west-2:444455556666:sns-topic-one</Topic>
        <Event>s3:ObjectCreated:*</Event>
        <Filter>
            <S3Key>
                <FilterRule>
                    <Name>suffix</Name>
                    <Value>jpg</Value>
                </FilterRule>
            </S3Key>
        </Filter>
    </TopicConfiguration>
    <TopicConfiguration>
        <Topic>arn:aws:sns:us-west-2:444455556666:sns-topic-two</Topic>
        <Event>s3:ObjectCreated:Put</Event>
        <Filter>
            <S3Key>
                <FilterRule>
                    <Name>suffix</Name>
                    <Value>pg</Value>
                </FilterRule>
            </S3Key>
        </Filter>
    </TopicConfiguration>
</NotificationConfiguration>
```

```
</FilterRule>
</S3Key>
</Filter>
</TopicConfiguration>
</NotificationConfiguration>
```

A configuração de notificação a seguir é inválida porque tem prefixos e sufixos sobrepostos.

```
<NotificationConfiguration>
  <TopicConfiguration>
    <Topic>arn:aws:sns:us-west-2:444455556666:sns-topic-one</Topic>
    <Event>s3:ObjectCreated:*</Event>
    <Filter>
      <S3Key>
        <FilterRule>
          <Name>prefix</Name>
          <Value>images</Value>
        </FilterRule>
        <FilterRule>
          <Name>suffix</Name>
          <Value>jpg</Value>
        </FilterRule>
      </S3Key>
    </Filter>
  </TopicConfiguration>
  <TopicConfiguration>
    <Topic>arn:aws:sns:us-west-2:444455556666:sns-topic-two</Topic>
    <Event>s3:ObjectCreated:Put</Event>
    <Filter>
      <S3Key>
        <FilterRule>
          <Name>suffix</Name>
          <Value>jpg</Value>
        </FilterRule>
      </S3Key>
    </Filter>
  </TopicConfiguration>
</NotificationConfiguration>
```

## Conceder permissões para publicar mensagens de notificação de vento a um destino

Para que o Amazon S3 possa publicar mensagens em um destino, você deve conceder a principal do Amazon S3 as permissões necessárias para chamar a API relevante para publicar mensagens em um tópico do SNS, em uma fila do SQS ou em uma função do Lambda.

## Conceder permissões para invocar uma função do AWS Lambda

O Amazon S3 publica mensagens de eventos no AWS Lambda invocando uma função do Lambda e fornecendo a mensagem de evento como um argumento.

Ao usar o console do Amazon S3 para configurar notificações de evento em um bucket do Amazon S3 para uma função do Lambda, o console do Amazon S3 configurará as permissões necessárias na função do Lambda para que o Amazon S3 tenha as permissões para invocar a função no bucket. Para obter mais

informações, consulte [Como habilitar e configurar notificações de evento para um bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

Você também pode conceder permissões ao Amazon S3 no AWS Lambda para invocar a função do Lambda. Para obter mais informações, consulte o [Tutorial: usar o AWS Lambda com o Amazon S3](#) no AWS Lambda Developer Guide.

## Conceder permissões para publicar mensagens em um tópico do SNS ou em uma fila do SQS

Você anexa uma política do IAM ao tópico do SNS de destino ou à fila do SQS para conceder permissões ao Amazon S3 para publicar mensagens no tópico do SNS ou na fila do SQS.

Exemplo de uma política do IAM que você anexa ao tópico do SNS de destino.

```
{  
    "Version": "2008-10-17",  
    "Id": "example-ID",  
    "Statement": [  
        {  
            "Sid": "example-statement-ID",  
            "Effect": "Allow",  
            "Principal": {  
                "Service": "s3.amazonaws.com"  
            },  
            "Action": [  
                "SNS:Publish"  
            ],  
            "Resource": "SNS-ARN",  
            "Condition": {  
                "ArnLike": { "aws:SourceArn": "arn:aws:s3:::bucket-name" }  
            }  
        }  
    ]  
}
```

Exemplo de uma política do IAM que você anexa à fila do SQS de destino.

```
{  
    "Version": "2008-10-17",  
    "Id": "example-ID",  
    "Statement": [  
        {  
            "Sid": "example-statement-ID",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "*"  
            },  
            "Action": [  
                "SQS:SendMessage"  
            ],  
            "Resource": "SQS-ARN",  
            "Condition": {  
                "ArnLike": { "aws:SourceArn": "arn:aws:s3:::bucket-name" }  
            }  
        }  
    ]  
}
```

Observe que para as políticas do IAM do Amazon SNS e do Amazon SQS, você pode especificar a condição `StringLike` na política, em vez da condição `ArnLike`.

```
"Condition": {  
    "StringLike": { "aws:SourceArn": "arn:aws:s3:::bucket-name" }  
}
```

Exemplo de uma política de chaves que você anexa à chave do KMS associada se a fila do SQS estiver habilitada para SSE.

```
{  
    "Version": "2012-10-17",  
    "Id": "example-ID",  
    "Statement": [  
        {  
            "Sid": "example-statement-ID",  
            "Effect": "Allow",  
            "Principal": {  
                "Service": "s3.amazonaws.com"  
            },  
            "Action": [  
                "kms:GenerateDataKey",  
                "kms:Decrypt"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

A política fornece ao serviço do Amazon S3 a permissão principal para ações específicas do KMS que são necessárias para criptografar mensagens adicionadas à fila.

Para obter um exemplo de como anexar uma política a um tópico do SNS ou a uma fila do SQS, consulte [Passo a passo do exemplo 1: configurar um bucket para notificações \(destino da mensagem: tópico do SNS e fila do SQS\) \(p. 533\)](#).

Para obter mais informações sobre permissões, consulte os tópicos a seguir:

- [Casos de exemplo para controle de acesso do Amazon SNS no Guia do desenvolvedor do Amazon Simple Notification Service](#)
- [Controle de acesso que usa o AWS Identity and Access Management \(IAM\) no Guia do desenvolvedor do Amazon Simple Queue Service](#)

## Passo a passo do exemplo 1: configurar um bucket para notificações (destino da mensagem: tópico do SNS e fila do SQS)

### Tópicos

- [Resumo do passo a passo \(p. 534\)](#)
- [Etapa 1: Criar um tópico do Amazon SNS \(p. 534\)](#)
- [Etapa 2: Criar uma fila do Amazon SQS \(p. 535\)](#)
- [Etapa 3: Adicionar a configuração de notificação ao bucket \(p. 536\)](#)
- [Etapa 4: Testar a configuração \(p. 539\)](#)

## Resumo do passo a passo

Neste passo a passo você adiciona a configuração de notificação em um bucket solicitando ao Amazon S3 para:

- Publicar eventos do tipo `s3:ObjectCreated:*` em uma fila do Amazon SQS.
- Publicar eventos do tipo `s3:ReducedRedundancyLostObject` em um tópico do Amazon SNS.

Para obter informações sobre configuração de notificação, consulte [Configurar notificações de evento do Amazon S3 \(p. 522\)](#).

Você pode executar todas essas etapas usando o console, sem escrever nenhum código. Além disso, exemplos de código, que usam os AWS SDKs para Java e .NET também são fornecidos, portanto, você pode adicionar a configuração de notificação de maneira programática.

Neste passo a passo, você fará o seguinte:

1. Crie um tópico do Amazon SNS.

Usando o console do Amazon SNS, você cria um tópico do SNS e se cadastra no tópico para que todos os eventos postados nele sejam entregues a você. Você especificará e-mail como o protocolo de comunicações. Depois de criar um tópico, o Amazon SNS enviará um e-mail. Você deve clicar em um link no e-mail para confirmar a assinatura de tópico.

Você anexará uma política de acesso ao tópico para conceder ao Amazon S3 permissão para postar mensagens.

2. Crie uma fila do Amazon SQS.

Usando o console do Amazon SQS, você cria uma fila do SQS. Você pode acessar todas as mensagens que o Amazon S3 envia à fila de maneira programática. Mas para este passo a passo, você verificará as mensagens de notificação no console.

Você anexará uma política de acesso ao tópico para conceder ao Amazon S3 permissão para postar mensagens.

3. Adicione a configuração de notificação a um bucket.

## Etapa 1: Criar um tópico do Amazon SNS

Siga as etapas para criar e assinar um tópico do Amazon Simple Notification Service (Amazon SNS).

1. Usando o console do Amazon SNS, crie um tópico. Para obter instruções, consulte [Criar um tópico](#) no Guia do desenvolvedor do Amazon Simple Notification Service.
2. Inscreva-se no tópico. Neste exercício, use o e-mail como o protocolo de comunicação. Para obter instruções, consulte [Inscrever-se em um tópico](#) no Guia do desenvolvedor do Amazon Simple Notification Service.

Você receberá um e-mail solicitando que confirme sua assinatura no tópico. Confirme a assinatura.

3. Substitua a política de acesso anexada ao tópico pela seguinte política. Você deve atualizar a política inserindo o nome de recurso da Amazon (ARN) do tópico do SNS e o nome do bucket:

```
{  
  "Version": "2008-10-17",  
  "Id": "example-ID",  
  "Statement": [  
    {  
      "Sid": "example-statement-ID",  
      "Effect": "Allow",  
      "Action": "SQS:SendMessage",  
      "Resource": "arn:aws:sns:us-east-1:123456789012:MyTopic",  
      "Condition": {"StringEquals": {"aws:SourceArn": "arn:aws:s3:us-east-1:123456789012:mybucket"}},  
      "Principal": "arn:aws:s3:::mybucket"  
    }  
  ]  
}
```

```
"Effect": "Allow",
"Principal": {
    "AWS": "*"
},
>Action": [
    "SNS:Publish"
],
"Resource": "SNS-topic-ARN",
"Condition": {
    "ArnLike": { "aws:SourceArn": "arn:aws:s3:::bucket-name" }
}
]
```

4. Anote o ARN do tópico.

O tópico do SNS que você criou é outro recurso em sua conta da AWS, e tem um Nome de recurso da Amazon (ARN) exclusivo. Você precisará desse ARN na próxima etapa. O ARN terá o seguinte formato:

```
arn:aws:sns:aws-region:account-id:topic-name
```

## Etapa 2: Criar uma fila do Amazon SQS

Siga as etapas para criar e assinar uma fila do Amazon Simple Queue Service (Amazon SQS).

1. Usando o console do Amazon SQS, crie uma fila. Para obter instruções, consulte [Conceitos básicos do Amazon SQS](#) no Guia do desenvolvedor do Amazon Simple Queue Service.
2. Substitua a política de acesso anexada à fila com a política a seguir (no console do SQS, você seleciona a fila e, na guia Permissions (Permissões), clica em Edit Policy Document (Advanced) (Editar documento da política [Avançado]).

```
{
    "Version": "2012-10-17",
    "Id": "example-ID",
    "Statement": [
        {
            "Sid": "example-statement-ID",
            "Effect": "Allow",
            "Principal": {
                "AWS": "*"
            },
            "Action": [
                "SQS:SendMessage"
            ],
            "Resource": "SQS-queue-ARN",
            "Condition": {
                "ArnLike": { "aws:SourceArn": "arn:aws:s3:::bucket-name" }
            }
        ]
    }
}
```

3. (Opcional) Se a fila do Amazon SQS for habilitada para Server-Side Encryption (SSE – Criptografia no lado do servidor), adicione a seguinte política à Customer Master Key (CMK – Chave mestra do cliente) personalizada associada do AWS Key Management Service (AWS KMS). Você deve adicionar a política a uma CMK personalizada porque a CMK gerenciada da AWS padrão para o Amazon SQS não pode ser modificada. Para obter mais informações sobre o uso do SSE para Amazon SQS com

AWS KMS, consulte [Proteger dados usando Server-Side Encryption \(SSE – Criptografia no lado do servidor\) e AWS KMS](#).

```
{  
    "Version": "2012-10-17",  
    "Id": "example-ID",  
    "Statement": [  
        {  
            "Sid": "example-statement-ID",  
            "Effect": "Allow",  
            "Principal": {  
                "Service": "s3.amazonaws.com"  
            },  
            "Action": [  
                "kms:GenerateDataKey",  
                "kms:Decrypt"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

4. Anotar o ARN da fila.

A fila do SQS que você criou é outro recurso em sua conta da AWS e tem um Nome de recurso da Amazon (ARN) exclusivo. Você precisará desse ARN na próxima etapa. O ARN terá o seguinte formato:

```
arn:aws:sqs:aws-region:account-id:queue-name
```

## Etapa 3: Adicionar a configuração de notificação ao bucket

Você pode habilitar notificações de bucket usando o console do Amazon S3 ou de maneira programática usando os AWS SDKs. Escolha qualquer uma das opções para configurar notificações no bucket. Esta seção fornece exemplos de código que usam os AWS SDKs para Java e .NET.

### Etapa 3 (opção a): Habilitar notificações em um bucket usando o console

Usando o console do Amazon S3, adicione uma configuração de notificação que solicita ao Amazon S3 para:

- Publicar eventos do tipo `s3:ObjectCreated:*` na fila do Amazon SQS.
- Publicar eventos do tipo `s3:ReducedRedundancyLostObject` no tópico do Amazon SNS.

Depois de salvar a configuração de notificação, o Amazon S3 postará uma mensagem de teste, que você receberá via e-mail.

Para obter instruções, consulte [Como habilitar e configurar notificações de evento para um bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

## Etapa 3 (opção b): Habilitar notificações em um bucket usando o AWS SDK para .NET

O exemplo de código C# a seguir fornece uma lista completa de códigos que adicionam uma configuração de notificação a um bucket. Você precisará atualizar o código e fornecer o nome do bucket e o ARN do tópico do SNS. Para obter informações sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

### Example

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class EnableNotificationsTest
    {
        private const string bucketName = "**** bucket name ****";
        private const string snsTopic = "**** SNS topic ARN ****";
        private const string sqsQueue = "**** SQS topic ARN ****";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            EnableNotificationAsync().Wait();
        }

        static async Task EnableNotificationAsync()
        {
            try
            {
                PutBucketNotificationRequest request = new PutBucketNotificationRequest
                {
                    BucketName = bucketName
                };

                TopicConfiguration c = new TopicConfiguration
                {
                    Events = new List<EventType> { EventType.ObjectCreatedCopy },
                    Topic = snsTopic
                };
                request.TopicConfigurations = new List<TopicConfiguration>();
                request.TopicConfigurations.Add(c);
                request.QueueConfigurations = new List<QueueConfiguration>();
                request.QueueConfigurations.Add(new QueueConfiguration()
                {
                    Events = new List<EventType> { EventType.ObjectCreatedPut },
                    Queue = sqsQueue
                });

                PutBucketNotificationResponse response = await
                client.PutBucketNotificationAsync(request);
            }
        }
    }
}
```

```
        catch (AmazonS3Exception e)
        {
            Console.WriteLine("Error encountered on server. Message:'{0}' ",
e.Message);
        }
        catch (Exception e)
        {
            Console.WriteLine("Unknown error encountered on server. Message:'{0}' ",
e.Message);
        }
    }
}
```

## Etapa 3 (opção c): Habilitar notificações em um bucket usando o AWS SDK for Java

O seguinte exemplo mostra como adicionar uma configuração de notificação a um bucket. Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

### Example

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-
developer-guide/blob/master/LICENSE-SAMPLECODE.)

import java.io.IOException;
import java.util.EnumSet;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.BucketNotificationConfiguration;
import com.amazonaws.services.s3.model.TopicConfiguration;
import com.amazonaws.services.s3.model.QueueConfiguration;
import com.amazonaws.services.s3.model.S3Event;
import com.amazonaws.services.s3.model.SetBucketNotificationConfigurationRequest;

public class EnableNotificationOnABucket {

    public static void main(String[] args) throws IOException {
        String bucketName = "*** Bucket name ***";
        String clientRegion = "*** Client region ***";
        String snsTopicARN = "*** SNS Topic ARN ***";
        String sqsQueueARN = "*** SQS Queue ARN ***";

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

            // Add an SNS topic notification.
            notificationConfiguration.addConfiguration("snsTopicConfig",
                new TopicConfiguration(snsTopicARN,
                EnumSet.of(S3Event.ObjectCreated)));

            // Add an SQS queue notification.
        }
    }
}
```

```
        notificationConfiguration.addConfiguration("sqsQueueConfig",
            new QueueConfiguration(sqsQueueARN,
                EnumSet.of(S3Event.ObjectCreated)));

        // Create the notification configuration request and set the bucket
        // notification configuration.
        SetBucketNotificationConfigurationRequest request = new
        SetBucketNotificationConfigurationRequest(
            bucketName, notificationConfiguration);
        s3Client.setBucketNotificationConfiguration(request);
    }
    catch(AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    }
    catch(SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}
```

## Etapa 4: Testar a configuração

Agora você pode testar a configuração carregando um objeto no bucket e verificando a notificação de evento no console do Amazon SQS. Para obter instruções, consulte [Receber uma mensagem](#) no Amazon Simple Queue Service Guia do desenvolvedor seção “Conceitos básicos”.

## Passo a passo do exemplo 2: configurar um bucket para notificações (destino da mensagem: AWS Lambda)

Para obter um exemplo de uso de notificações do Amazon S3 com o AWS Lambda, consulte [Usar o AWS Lambda com o Amazon S3](#) no AWS Lambda Developer Guide.

## Estrutura de mensagens de evento

A mensagem de notificação que o Amazon S3 envia para publicar um evento tem o formato JSON. O exemplo a seguir mostra a estrutura da mensagem JSON.

Observe a seguinte informação sobre o exemplo:

- O valor da chave `eventVersion` contém uma versão principal e secundária no formulário `<major>.<minor>`.

A versão principal é incrementada se o Amazon S3 faz uma alteração para a estrutura do evento que não é compatível com as versões anteriores. Isso inclui a remoção de um campo JSON que já está presente ou a alteração de como os conteúdos de um campo são representados (por exemplo, um formato de data).

A versão secundária é incrementada se o Amazon S3 adiciona novos campos à estrutura do evento. Isso pode ocorrer se novas informações forem fornecidas para alguns ou todos os eventos existentes

ou somente em tipos de evento recém-introduzidos. Os aplicativos devem ignorar novos campos para permanecerem compatíveis com novas versões secundárias da estrutura do evento.

Se novos tipos de evento são introduzidos, mas a estrutura do evento não é modificada, a versão do evento não muda.

Para garantir que seus aplicativos possam analisar a estrutura do evento corretamente, recomendamos que você faça uma comparação "igual a" no número da versão principal. Para garantir que os campos esperados pelo seu aplicativo estejam presentes, também recomendamos fazer uma comparação "maior que ou igual a" na versão secundária.

- O valor da chave `responseElements` será útil se você quiser rastrear uma solicitação acompanhando o AWS Support. O `x-amz-request-id` e o `x-amz-id-2` ajudam o Amazon S3 a rastrear uma solicitação individual. Esses valores são os mesmos que os retornados pelo Amazon S3 na resposta à solicitação que inicia os eventos, para que possam ser usados para corresponder o evento à solicitação.
- A chave `s3` fornece informações sobre o bucket e o objeto envolvidos no evento. O valor do nome da chave de objetos é codificado para URL. Por exemplo, "red flower.jpg" se torna "red+flower.jpg" (o Amazon S3 retorna o "application/x-www-form-urlencoded" como o tipo do conteúdo na resposta).
- A chave `sequencer` fornece uma maneira de determinar a sequência de eventos. Não há garantia de que as notificações de evento cheguem na ordem em que os eventos ocorreram. No entanto, as notificações de eventos que criam objetos (PUTs) e excluem objetos contêm um `sequencer` que pode ser usado para determinar a ordem dos eventos de uma determinada chave de objeto.

Se você comparar a sequências do `sequencer` nas duas notificações de evento na mesma chave de objeto, a notificação de evento com o valor hexadecimal maior de `sequencer` será o evento que ocorreu depois. Se estiver usando notificações de evento para manter um banco de dados ou um índice separado dos objetos do Amazon S3, você provavelmente desejará comparar e armazenar os valores de `sequencer` ao processar cada notificação de evento.

Observe o seguinte:

- O `sequencer` não pode ser usado para determinar a ordem de eventos em chaves de objetos diferentes.
- Os sequenciadores podem ter comprimentos diferentes. Portanto, para comparar esses valores, primeiro você preenche zeros à esquerda do menor valor e, em seguida, faz uma comparação lexicográfica.
- A chave `glacierEventData` só está visível para eventos `s3:ObjectRestore:Completed`.
- A chave `restoreEventData` contém atributos relacionados à solicitação de restauração.

O exemplo a seguir mostra a versão 2.1 da mensagem do evento para a estrutura JSON, que é a versão usada atualmente pelo Amazon S3.

```
{  
  "Records": [  
    {  
      "eventVersion": "2.1",  
      "eventSource": "aws:s3",  
      "awsRegion": "us-west-2",  
      "s3": {  
        "approximateCreationTimestamp": "1537136349.123",  
        "bucket": "my-bucket",  
        "objectKey": "my-object-key",  
        "size": 1024  
      }  
    }  
  ]  
}
```

```

    "eventTime":The time, in ISO-8601 format, for example, 1970-01-01T00:00:00.000Z, when Amazon S3 finished processing the request,
    "eventName":"event-type",
    "userIdentity":{
        "principalId":"Amazon-customer-ID-of-the-user-who-caused-the-event"
    },
    "requestParameters":{
        "sourceIPAddress":"ip-address-where-request-came-from"
    },
    "responseElements":{
        "x-amz-request-id":"Amazon S3 generated request ID",
        "x-amz-id-2":"Amazon S3 host that processed the request"
    },
    "s3":{
        "s3SchemaVersion":"1.0",
        "configurationId":"ID found in the bucket notification configuration",
        "bucket":{
            "name":"bucket-name",
            "ownerIdentity":{
                "principalId":"Amazon-customer-ID-of-the-bucket-owner"
            },
            "arn":"bucket-ARN"
        },
        "object":{
            "key":"object-key",
            "size":object-size,
            "eTag":"object eTag",
            "versionId":"object version if bucket is versioning-enabled, otherwise null",
            "sequencer": "a string representation of a hexadecimal value used to determine event sequence, only used with PUTs and Deletes"
        }
    },
    "glacierEventData": {
        "restoreEventData": {
            "lifecycleRestorationExpiryTime": "The time, in ISO-8601 format, for example, 1970-01-01T00:00:00.000Z, of Restore Expiry",
            "lifecycleRestoreStorageClass": "Source storage class for restore"
        }
    }
}
]
}
}

```

O exemplo a seguir mostra a versão 2.0 da estrutura da mensagem do evento, que deixou de ser usada pelo Amazon S3.

```

{
    "Records": [
        {
            "eventVersion":"2.0",
            "eventSource":"aws:s3",
            "awsRegion":"us-west-2",
            "eventTime":The time, in ISO-8601 format, for example, 1970-01-01T00:00:00.000Z, when S3 finished processing the request,
            "eventName":"event-type",
            "userIdentity":{
                "principalId":"Amazon-customer-ID-of-the-user-who-caused-the-event"
            },
            "requestParameters":{
                "sourceIPAddress":"ip-address-where-request-came-from"
            },
            "responseElements":{

```

```
"x-amz-request-id": "Amazon S3 generated request ID",
"x-amz-id-2": "Amazon S3 host that processed the request"
},
"s3": {
    "s3SchemaVersion": "1.0",
    "configurationId": "ID found in the bucket notification configuration",
    "bucket": {
        "name": "bucket-name",
        "ownerIdentity": {
            "principalId": "Amazon-customer-ID-of-the-bucket-owner"
        },
        "arn": "bucket-ARN"
    },
    "object": {
        "key": "object-key",
        "size": object-size,
        "eTag": "object eTag",
        "versionId": "object version if bucket is versioning-enabled, otherwise
null",
        "sequencer": "a string representation of a hexadecimal value used to
determine event sequence,
only used with PUTs and Deletes"
    }
}
]
}
```

As seguintes são mensagens de exemplo:

- Mensagem de teste — quando você configura uma notificação de evento em um bucket, o Amazon S3 envia a seguinte mensagem de teste:

```
{
    "Service": "Amazon S3",
    "Event": "s3:TestEvent",
    "Time": "2014-10-13T15:57:02.089Z",
    "Bucket": "bucketname",
    "RequestId": "5582815E1AEA5ADF",
    "HostId": "8cLeGAmw098X5cv4Zkwcmo8vvZa3eH3eKxsPzbB9wrR+YstdA6Knx4Ip8EXAMPLE"
}
```

- Mensagem de exemplo quando um objeto é criado usando a solicitação PUT — a mensagem a seguir é um exemplo de uma mensagem que o Amazon S3 envia para publicar um evento do s3:ObjectCreated:Put:

```
{
    "Records": [
        {
            "eventVersion": "2.1",
            "eventSource": "aws:s3",
            "awsRegion": "us-west-2",
            "eventTime": "1970-01-01T00:00:00.000Z",
            "eventName": "ObjectCreated:Put",
            "userIdentity": {
                "principalId": "AIDAJDPLRKLG7UEEXAMPLE"
            },
            "requestParameters": {
                "sourceIPAddress": "127.0.0.1"
            },
            "responseElements": {
                "x-amz-request-id": "C3D13FE58DE4C810",
                "ETag": "EAD2D8A83E8A8D8A83E8A83E8A83E8A8"
            }
        }
    ]
}
```

```
"x-amz-id-2": "FMyUVURIY8/IgAtTv8xRjskZQpcIZ9KG4V5Wp6S7S/  
JRWeUWerMUE5JgHvANOjpD"  
,  
"s3":{  
    "s3SchemaVersion": "1.0",  
    "configurationId": "testConfigRule",  
    "bucket":{  
        "name": "mybucket",  
        "ownerIdentity":{  
            "principalId": "A3NL1KOZZKExample"  
        },  
        "arn": "arn:aws:s3:::mybucket"  
    },  
    "object":{  
        "key": "HappyFace.jpg",  
        "size": 1024,  
        "eTag": "d41d8cd98f00b204e9800998ecf8427e",  
        "versionId": "096fKKXTRTtl3on89fVO.nfljtsv6qko",  
        "sequencer": "0055AED6DCD90281E5"  
    }  
}  
}  
]  
}
```

# Replicação entre regiões

A replicação entre regiões (CRR) permite a cópia automática assíncrona de objetos entre buckets em diferentes regiões da AWS. Buckets configurados para replicação entre regiões podem ser de propriedade da mesma conta da AWS ou de diferentes contas.

A replicação entre regiões é habilitada com uma configuração no nível do bucket. Você adiciona uma configuração de replicação ao bucket de origem. Na configuração mínima, você fornece o seguinte:

- O bucket de destino no qual você quer que o Amazon S3 replique os objetos
- Uma função do AWS IAM que o Amazon S3 pode assumir para replicar objetos em seu nome

Estão disponíveis opções de configuração adicionais.

## Quando usar CRR

A replicação entre regiões pode ajudar você a fazer o seguinte:

- Cumprir os requisitos de conformidade — Embora o Amazon S3 armazene seus dados em diversas zonas de disponibilidade geograficamente distantes por padrão, requisitos de conformidade podem ditar que você armazene os dados em distâncias ainda maiores. A replicação entre regiões permite que você replique dados entre regiões da AWS distantes para satisfazer esses requisitos.
- Minimizar a latência — Se os seus clientes estiverem em duas localizações geográficas distintas, é possível minimizar a latência no acesso de objetos ao manter cópias deles nas regiões da AWS geograficamente mais próximas dos usuários.
- Aumentar a eficiência operacional — Se você computar clusters em duas regiões da AWS diferentes que analisem o mesmo grupo de objetos, você pode optar por manter cópias do objeto nessas Regiões.
- Manter cópias de objetos em diferentes propriedades — Independentemente de quem é o proprietário do objeto de origem, você pode orientar o Amazon S3 a alterar a propriedade sobre a réplica para a conta da AWS que tem o bucket de destino. Isso se chama opção de substituição do proprietário. Você pode usar esta opção para restringir acesso às replicas do objeto.

## Requisitos para CRR

A replicação entre regiões exige o seguinte:

- Tanto o bucket de origem quanto o de destino devem ter o versionamento habilitado.
- Os buckets de origem e destino devem estar em regiões da AWS diferentes.
- O Amazon S3 deve ter permissões para replicar objetos do bucket de origem para o bucket de destino em seu nome.

- Se o proprietário do bucket de origem não for proprietário do objeto no bucket, o proprietário do objeto precisará conceder ao proprietário do bucket as permissões READ e READ\_ACP com a ACL do projeto. Para obter mais informações, consulte [Gerenciar o acesso com ACLs \(p. 390\)](#).

Para obter mais informações, consulte [Visão geral da configuração da CRR \(p. 547\)](#).

Se você estiver definindo a configuração de replicação em um cenário entre contas (onde os buckets de origem e de destino pertencem a diferentes contas da AWS), os seguintes requisitos adicionais se aplicarão:

- O proprietário do bucket de destino precisa conceder ao proprietário do bucket de origem permissões para replicar objetos com uma política do bucket. Para obter mais informações, consulte [Concessão de permissões quando os buckets de origem e de destino forem de propriedade de diferentes contas da AWS \(p. 559\)](#).

## O que o Amazon S3 replica?

O Amazon S3 replica somente itens específicos nos buckets que estão configurados para replicação entre regiões.

### O que é replicado?

O Amazon S3 replica o seguinte:

- Objetos criados depois que você adiciona uma configuração de replicação, com as exceções descritas na próxima seção.
- Objetos não criptografados e objetos criptografados usando as chaves gerenciadas pelo Amazon S3 (SSE-S3) ou chaves gerenciadas pelo AWS KMS (SSE-KMS), ainda que você precise habilitar explicitamente a opção para replicar objetos criptografados usando as chaves do KMS. A cópia replicada do objeto é criptografada usando o mesmo tipo de criptografia do lado do servidor que foi usada para o objeto de origem. Para obter mais informações sobre criptografia no lado do servidor, consulte [Proteção de dados usando criptografia no lado do servidor \(p. 410\)](#).
- Metadados do objeto.
- Somente os objetos no bucket de origem para os quais o proprietário do bucket tenha permissões para ler objetos e listas de controle de acesso (ACLs). Para obter mais informações sobre propriedade de recurso, consulte [Sobre o proprietário de recursos \(p. 283\)](#).
- A ACL do objeto é atualizada, a menos que você oriente o Amazon S3 a alterar a propriedade da réplica quando os buckets de origem e de destino não forem de propriedade das mesmas contas (consulte [Configuração adicional da CRR: alteração do proprietário da réplica \(p. 560\)](#)).

Pode levar um tempo até o Amazon S3 sincronizar as duas ACLs. Isso aplica-se apenas a objetos criados depois que você adiciona uma configuração de replicação ao bucket.

- Tags de objeto, se houver.

## Como a exclusão de operações afeta a CRR

Se você excluir um objeto do bucket de origem, ocorrerá o seguinte:

- Se você fizer a solicitação `DELETE` sem especificar o ID da versão do objeto, o Amazon S3 vai adicionar um marcador de exclusão. O Amazon S3 lida da seguinte forma com o marcador de exclusão:
  - Se estiver usando a versão mais recente da configuração de replicação, ou seja, se você especificar o elemento `Filter` em uma regra de configuração da replicação, o Amazon S3 não vai replicar o marcador de exclusão.
  - Se não especificar o elemento `Filter`, o Amazon S3 vai pressupor que a configuração da replicação é uma V1 de versão anterior. Na versão anterior, o Amazon S3 lidava de outro jeito com a replicação dos marcadores de exclusão. Para obter mais informações, consulte [Compatibilidade retroativa \(p. 556\)](#).
- Se você especificar um ID da versão do objeto a ser excluído na solicitação `DELETE`, o Amazon S3 excluirá essa versão do objeto no bucket de origem, mas não replicará a exclusão no bucket de destino. Em outras palavras: ele não exclui a mesma versão do objeto do bucket de destino. Isso protege os dados contra exclusões mal-intencionadas.

## O que não é replicado?

O Amazon S3 não replica o seguinte:

- Objetos que existiam antes de você adicionar a configuração de replicação ao bucket. Em outras palavras: o Amazon S3 não replica os objetos retroativamente.
- Os seguintes objetos criptografados:
  - Objetos criados com criptografia do lado do servidor usando as chaves de criptografia (SSE-C) fornecidas pelo usuário.
  - Objetos criados com criptografia no lado do servidor usando as chaves de criptografia gerenciadas pelo AWS KMS (SSE-KMS). Por padrão, o Amazon S3 não replica objetos criptografados usando as chaves do KMS. No entanto, você pode habilitar explicitamente a replicação desses objetos na configuração de replicação e fornecer as informações pertinentes para que o Amazon S3 consiga replicar esses objetos.

Para obter mais informações sobre criptografia no lado do servidor, consulte [Proteção de dados usando criptografia no lado do servidor \(p. 410\)](#).

- Objetos no bucket de origem para os quais o proprietário do bucket não tenha permissão (quando ele não for o proprietário do objeto). Para obter informações sobre como o proprietário de um objeto pode conceder permissões ao proprietário do bucket, consulte [Conceder permissões entre contas para fazer upload de objetos garantindo que o proprietário do bucket tenha controle total \(p. 364\)](#).
- Atualizações nos sub-recursos no nível do bucket. Por exemplo, se você alterar a configuração do ciclo de vida ou adicionar uma configuração de notificação ao bucket de origem, essas alterações não serão aplicadas ao bucket de destino. Isso permite termos diferentes configurações nos buckets de origem e de destino.

- Ações realizadas pela configuração do ciclo de vida.

Por exemplo, se a configuração de ciclo de vida estiver habilitada apenas no seu bucket de origem, o Amazon S3 criará marcadores de exclusão para objetos expirados, mas não replicará esses marcadores. Caso queira que as mesmas configurações de ciclo de vida sejam aplicadas ao bucket de origem e de destino, habilite a mesma configuração de ciclo de vida em ambos.

Para obter mais informações sobre a configuração do ciclo de vida, consulte [Gerenciamento do ciclo de vida de objetos \(p. 122\)](#).

Note

Se estiver usando a versão mais recente da configuração de replicação (o XML especifica `Filter` como filho de `Rule`), não serão replicados os marcadores de exclusão criados por ação do usuário ou pelo Amazon S3 como parte da ação do ciclo de vida. No entanto, se estiver usando uma versão anterior da configuração de replicação (o XML especifica `Prefix` como filho de `Rule`), os marcadores de exclusão resultantes das ações do usuário são replicados.

Para obter mais informações, consulte [Compatibilidade retroativa \(p. 556\)](#).

- Objetos no bucket de origem que são réplicas criadas por replicação entre regiões.

Você pode replicar objetos de um bucket de origem para apenas um bucket de destino. Depois que o Amazon S3 replica um objeto, ele não pode ser replicado novamente. Por exemplo, se você alterar o bucket de destino em uma configuração de replicação existente, o Amazon S3 não replicará o objeto novamente.

Outro exemplo: vamos supor que você configure a replicação entre regiões em que o bucket A é a origem e o bucket B é o destino. Agora suponha que você adicione outra configuração da replicação entre regiões em que o bucket B é a origem e o bucket C é o destino. Neste caso, os objetos no bucket B que são réplicas de objetos no bucket A não serão replicados para o bucket C.

## Tópicos relacionados

[Replicação entre regiões \(p. 544\)](#)

[Visão geral da configuração da CRR \(p. 547\)](#)

[Informação sobre o status da replicação entre regiões \(p. 585\)](#)

## Visão geral da configuração da CRR

Para habilitar a replicação entre regiões (CRR), adicione uma configuração de replicação ao bucket de origem. A configuração diz ao Amazon S3 para replicar objetos, conforme especificado. Na configuração de replicação, você deve fornecer o seguinte:

- O bucket de destino — O bucket no qual você quer que o Amazon S3 replique os objetos.
- Os objetos que você deseja replicar — Você pode replicar todos os objetos no bucket de origem ou em um subgrupo. Identifique um subgrupo fornecendo, na configuração, um prefixo do nome da chave, uma ou mais tags de objeto ou ambos. Por exemplo, se você configurar a replicação entre regiões para replicar somente objetos com o prefixo de nome da chave `Tax/`, o Amazon S3 vai replicar objetos com chaves como `Tax/doc1` ou `Tax/doc2`, mas não objetos com a chave `Lega1/doc3`. Se você especificar tanto o prefixo quanto uma ou mais tags, o Amazon S3 vai replicar somente objetos com o prefixo de chaves específico e as tags.

A réplica tem os mesmos nomes de chave e metadados (por exemplo, hora de criação, metadados definidos pelo usuário e ID da versão) que o objeto original. O Amazon S3 criptografa todos os dados em trânsito nas regiões da AWS usando Secure Sockets Layer (SSL).

Além desses requisitos mínimos, você pode escolher as seguintes opções:

- Por padrão, o Amazon S3 armazena réplicas de objetos usando a mesma classe de armazenamento que o objeto de origem. Você pode especificar uma classe de armazenamento diferente para as réplicas.
- Como isso pressupõe que a réplica do objeto continue pertencendo ao proprietário do objeto de origem, quando o Amazon S3 replicar os objetos, ele também vai replicar a lista de controle de acesso do objeto correspondente. Se os buckets de origem e destino forem de propriedade de diferentes contas da AWS, você poderá configurar a CRR para alterar o proprietário de uma réplica para a conta da AWS que é proprietária do bucket de destino.

Estão disponíveis opções de configuração adicionais. Para obter mais informações, consulte [Configurações adicionais de CRR \(p. 560\)](#).

**Important**

Se você tem uma política de ciclo de vida de expiração do objeto em seu bucket sem versão e quer manter o mesmo comportamento de exclusão permanente quando ativar o controle de versão, precisará adicionar uma política de expiração de versão desatualizada. A política de expiração do ciclo de vida gerenciará as exclusões de versões desatualizadas de objeto no bucket habilitado para versão. (Um bucket habilitado para versão mantém uma versão atual e zero ou mais versões desatualizadas de objeto.) Para mais informações, consulte [Como faço para criar uma política de ciclo de vida para um bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

O Amazon S3 fornece APIs em suporte à replicação entre regiões. Para obter mais informações, consulte os seguintes tópicos em Amazon Simple Storage Service API Reference:

- [Replicação do PUT Bucket](#)
- [Replicação do GET Bucket](#)
- [DELETE replicação de bucket](#)

Em vez de criar essas chamadas de API diretamente do código, você pode adicionar uma configuração de replicação a um bucket com AWS SDK, AWS CLI ou console do Amazon S3. É mais fácil usar o console. Para ver exemplos com instruções detalhadas, consulte [Demonstrações da replicação entre regiões \(CRR\) \(p. 567\)](#).

Se para você a configuração da CRR for novidade, recomendamos ler as visões gerais a seguir antes de explorar exemplos e configurações opcionais. Os exemplos dão instruções detalhadas para fazer configurações básicas da CRR. Para obter mais informações, consulte [Demonstrações da replicação entre regiões \(CRR\) \(p. 567\)](#).

**Tópicos**

- [Visão geral da configuração da replicação \(p. 548\)](#)
- [Configurar permissões para CRR \(p. 556\)](#)

## Visão geral da configuração da replicação

O Amazon S3 armazena uma configuração de replicação como XML. No arquivo XML de configuração da replicação, você especifica uma função AWS Identity and Access Management (IAM) e uma ou mais regras.

```
<ReplicationConfiguration>
  <Role>IAM-role-ARN</Role>
  <Rule>
    ...
  </Rule>
  <Rule>
    ...
  </Rule>
  ...
</ReplicationConfiguration>
```

O Amazon S3 não pode replicar objetos sem sua permissão. Você concede permissões com a função do IAM especificada na configuração da replicação. O Amazon S3 pressupõe a função do IAM para replicar objetos em seu lugar. Primeiro você precisa fornecer as permissões necessárias à função do IAM. Para obter mais informações sobre como gerenciar permissões, consulte [Configurar permissões para CRR \(p. 556\)](#).

Você adiciona uma regra na configuração da replicação nos seguintes cenários:

- Você quer replicar todos os objetos.
- Você quer replicar um subgrupo de objetos. Você identifica o subgrupo do objeto adicionando um filtro à regra. No filtro, você especifica um prefixo de chaves do objeto, tags ou uma combinação de ambos, de maneira a identificar o subgrupo de objetos aos quais a regra se aplica.

Você adiciona várias regras a uma configuração de replicação, caso deseje selecionar um subgrupo diferente de objetos. Em cada regra, você especifica um filtro que seleciona um subgrupo diferente de objetos. Por exemplo, você pode optar por replicar objetos com os prefixos de chaves `tax/` ou `document/`. Você precisaria adicionar duas regras e especificar o filtro do prefixo de chaves `tax/` em uma regra e o prefixo de chaves `document/` na outra.

As seções a seguir fornecem informações adicionais.

#### Tópicos

- [A configuração da regra básica \(p. 549\)](#)
- [Opcional: Especificação de um filtro \(p. 550\)](#)
- [Configurações adicionais de destino \(p. 551\)](#)
- [Exemplo de configurações de replicação \(p. 552\)](#)
- [Compatibilidade retroativa \(p. 556\)](#)

## A configuração da regra básica

Cada regra precisa incluir o status e a prioridade da regra, além de indicar se esses marcadores de exclusão devem ou não ser replicados.

- **Status** indica se a regra está habilitada ou desabilitada. Se uma regra estiver desabilitada, o Amazon S3 não executará as ações especificadas nela.
- **Priority** indica qual regra tem prioridade, quando várias regras forem aplicáveis a um objeto.
- **No momento**, os marcadores de exclusão não são replicados. Por isso, defina `DeleteMarkerReplication` para `Disabled`.

Na configuração de destino, dê o nome do bucket onde você quer que o Amazon S3 replique objetos.

O código a seguir mostra os requisitos mínimos para a regra:

```
...
```

```
<Rule>
  <ID>Rule-1</ID>
  <Status>rule-Enabled-or-Disabled</Status>
  <Priority>integer</Priority>
  <DeleteMarkerReplication>
    <Status>Disabled</Status>
  </DeleteMarkerReplication>
  <Destination>
    <Bucket>arn:aws:s3:::bucket-name</Bucket>
  </Destination>
</Rule>
<Rule>
  ...
</Rule>
  ...
...
...
```

Você também pode especificar outras opções de configuração. Por exemplo: você pode optar pelo uso de uma classe de armazenamento para réplicas de objetos diferentes da classe do objeto de origem.

## Opcional: Especificação de um filtro

Para escolher um subgrupo de objetos aos quais a regra se aplique, adicione um filtro opcional. Você pode filtrar por prefixo de chaves do objeto, tags do objeto ou uma combinação dos dois. Se você filtrar tanto por prefixo de chaves quanto por tags de objeto, o Amazon S3 combinará os filtros usando o operador lógico AND. Em outras palavras: a regra se aplica ao subgrupo de objetos com o prefixo de chaves específico e as tags específicas.

Para especificar uma regra com um filtro baseado no prefixo de chaves de um objeto, use o código a seguir. Você pode especificar apenas um prefixo.

```
<Rule>
  ...
  <Filter>
    <Prefix>key-prefix</Prefix>
  </Filter>
  ...
</Rule>
...
```

Para especificar uma regra com um filtro baseado nas tags do objeto, use o código a seguir. Você pode especificar uma ou mais tags de objeto.

```
<Rule>
  ...
  <Filter>
    <And>
      <Tag>
        <Key>key1</Key>
        <Value>value1</Value>
      </Tag>
      <Tag>
        <Key>key2</Key>
        <Value>value2</Value>
      </Tag>
      ...
    </And>
  </Filter>
  ...
</Rule>
...
```

Para especificar um filtro de regra com uma combinação de prefixo de chaves e tags de objeto, use este código. Você engloba esses filtros em um elemento-pai AND. O Amazon S3 faz lógica E operação para combinar os filtros. Em outras palavras: a regra se aplica ao subgrupo de objetos com o prefixo de chaves específico e as tags específicas.

```
<Rule>
  ...
  <Filter>
    <And>
      <Prefix>key-prefix</Prefix>
      <Tag>
        <Key>key1</Key>
        <Value>value1</Value>
      </Tag>
      <Tag>
        <Key>key2</Key>
        <Value>value2</Value>
      </Tag>
      ...
    </And>
  </Filter>
  ...
</Rule>
...
```

## Configurações adicionais de destino

Na configuração de destino, especifique o bucket onde você quer que o Amazon S3 replique objetos. Você pode configurar a CRR para replicar objetos de um bucket de origem para um bucket de destino. Se você adicionar várias regras em uma configuração de replicação, todas elas deverão identificar o mesmo bucket de destino.

```
...
<Destination>
  <Bucket>arn:aws:s3:::destination-bucket</Bucket>
</Destination>
...
```

Você tem as opções a seguir para adicionar ao elemento <Destination>:

- Você pode especificar a classe de armazenamento para as réplicas do objeto. Por padrão, o Amazon S3 usa a classe de armazenamento do objeto de origem para criar as réplicas do objeto. Por exemplo,

```
...
<Destination>
  <Bucket>arn:aws:s3:::destinationbucket</Bucket>
  <StorageClass>storage-class</StorageClass>
</Destination>
...
```

- Quando os buckets de origem e de destino não forem de propriedade da mesma conta, você poderá alterar a propriedade da réplica para a conta da AWS que é proprietária do bucket de destino, adicionando o elemento AccessControlTranslation:

```
...
<Destination>
  <Bucket>arn:aws:s3:::destinationbucket</Bucket>
  <Account>destination-bucket-owner-account-id</Account>
  <AccessControlTranslation>
    <Owner>Destination</Owner>
  </AccessControlTranslation>
...
```

```
</Destination>
...

```

Se você não adicionar esse elemento à configuração de replicação, as réplicas serão de propriedade da mesma conta da AWS que possui o objeto de origem. Para obter mais informações, consulte [Configuração adicional da CRR: alteração do proprietário da réplica \(p. 560\)](#).

- O bucket de origem pode conter objetos criados com criptografia do lado do servidor usando as chaves gerenciadas pelo AWS KMS. Por padrão, o Amazon S3 não replica esses objetos. Você também pode orientar o Amazon S3 a replicar esses objetos ao primeiro optar explicitamente por esse recurso ao adicionar o elemento SourceSelectionCriteria e depois fornecer a chave do AWS KMS (para a região da AWS do bucket de destino) para uso ao criptografar réplicas do objeto.

```
...
<SourceSelectionCriteria>
    <SseKmsEncryptedObjects>
        <Status>Enabled</Status>
    </SseKmsEncryptedObjects>
</SourceSelectionCriteria>
<Destination>
    <Bucket>arn:aws:s3:::dest-bucket-name</Bucket>
    <EncryptionConfiguration>
        <ReplicaKmsKeyId>AWS KMS key IDs to use for encrypting object replicas</ReplicaKmsKeyId>
    </EncryptionConfiguration>
</Destination>
...

```

Para obter mais informações, consulte [Outra configuração de CRR: replicar objetos criados com a criptografia do lado do servidor \(SSE\) usando chaves de criptografia gerenciadas pelo AWS KMS \(p. 563\)](#).

## Exemplo de configurações de replicação

Para começar, você pode adicionar os seguintes exemplos de configuração de replicação ao bucket, conforme adequado.

### Important

Para adicionar uma configuração de replicação a um bucket, é preciso ter a permissão `iam:PassRole`. Com essa permissão, você pode aprovar a função do IAM que concede as permissões de replicação do Amazon S3. Você especifica a função do IAM ao fornecer o nome de recurso da Amazon (ARN) usado no elemento `Role` na configuração de replicação XML. Para obter mais informações, consulte [Conceder permissões ao usuário para aprovar uma função para um serviço da AWS](#), no Guia do usuário do IAM.

### Example 1: Configuração de replicação com uma regra

A configuração de replicação básica a seguir especifica uma regra. A regra especifica uma função do IAM que o Amazon S3 pode assumir e um bucket de destino para réplicas de objetos. A regra `Status` indica que a regra está em vigor.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
    <Role>arn:aws:iam::AcctID:role/role-name</Role>
    <Rule>
        <Status>Enabled</Status>

        <Destination><Bucket>arn:aws:s3:::destinationbucket</Bucket></Destination>
```

```
</Rule>  
</ReplicationConfiguration>
```

Para escolher um subgrupo de objetos a serem replicados, adicione um filtro. Na configuração a seguir, o filtro especifica um prefixo de chaves do objeto. Essa regra se aplica aos objetos que têm o prefixo Tax/ no nome da chave.

```
<?xml version="1.0" encoding="UTF-8"?>  
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">  
  <Role>arn:aws:iam::AcctID:role/role-name</Role>  
  <Rule>  
    <Status>Enabled</Status>  
    <Priority>1</Priority>  
    <DeleteMarkerReplication>  
      <Status>string</Status>  
    </DeleteMarkerReplication>  
  
    <Filter>  
      <Prefix>Tax/</Prefix>  
    </Filter>  
  
    <Destination><Bucket>arn:aws:s3:::destinationbucket</Bucket></Destination>  
  
  </Rule>  
</ReplicationConfiguration>
```

Se você especificar o elemento **Filter**, inclua também os elementos **Priority** e **DeleteMarkerReplication**. Neste exemplo, a prioridade é irrelevante, pois há somente uma regra.

Na configuração a seguir, o filtro especifica um prefixo e duas tags. A regra se aplica ao subgrupo de objetos com o prefixo de chaves e as tags especificados. Mais especificamente, ela se aplica ao objeto com o prefixo Tax/ no nome da chave e às duas tags de objeto especificadas. Não se aplica prioridade, pois há somente uma regra.

```
<?xml version="1.0" encoding="UTF-8"?>  
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">  
  <Role>arn:aws:iam::AcctID:role/role-name</Role>  
  <Rule>  
    <Status>Enabled</Status>  
    <Priority>1</Priority>  
    <DeleteMarkerReplication>  
      <Status>string</Status>  
    </DeleteMarkerReplication>  
  
    <Filter>  
      <And>  
        <Prefix>Tax/</Prefix>  
        <Tag>  
          <Tag>  
            <Key>tagA</Key>  
            <Value>valueA</Value>  
          </Tag>  
        </Tag>  
        <Tag>  
          <Tag>  
            <Key>tagB</Key>  
            <Value>valueB</Value>  
          </Tag>  
        </Tag>  
      </And>  
    </Filter>
```

```
<Destination><Bucket>arn:aws:s3:::destinationbucket</Bucket></Destination>

</Rule>
</ReplicationConfiguration>
```

Você pode especificar uma classe de armazenamento para as réplicas, da seguinte forma:

```
<?xml version="1.0" encoding="UTF-8"?>

<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>
    <Destination>
      <Bucket>arn:aws:s3:::destinationbucket</Bucket>
      <StorageClass>storage-class</StorageClass>
    </Destination>
  </Rule>
</ReplicationConfiguration>
```

Especifique qualquer classe de armazenamento compatível com o Amazon S3.

#### Example 2: Configuração de replicação com duas regras

##### Example

Na seguinte configuração de replicação:

- Cada regra filtra um diferente prefixo de chaves, de maneira que cada uma delas se aplica a um subgrupo distinto de objetos. O Amazon S3 replica objetos com nomes de chave Tax/doc1.pdf e Project/project1.txt, mas não replica objetos com o nome de chave PersonalDoc/documentA.
- A prioridade da regra é irrelevante, pois as regras se aplicam a dois grupos distintos de objetos. O exemplo a seguir mostra o que acontece quando aplicamos uma prioridade de regras.
- A segunda regra especifica uma classe de armazenamento para réplicas de objetos. O Amazon S3 usa a classe de armazenamento especificada para essas réplicas de objetos.
- Ambas as regras especificam o mesmo bucket de destino. Você pode especificar somente um bucket de destino, não importa quantas regras forem especificadas.

```
<?xml version="1.0" encoding="UTF-8"?>

<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
      <Status>string</Status>
    </DeleteMarkerReplication>
    <Filter>
      <Prefix>Tax</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Destination>
      <Bucket>arn:aws:s3:::destinationbucket</Bucket>
    </Destination>
    ...
  </Rule>
  <Rule>
    <Status>Enabled</Status>
```

```
<Priority>2</Priority>
<DeleteMarkerReplication>
    <Status>string</Status>
</DeleteMarkerReplication>
<Filter>
    <Prefix>Project</Prefix>
</Filter>
<Status>Enabled</Status>
<Destination>
    <Bucket>arn:aws:s3:::destinationbucket</Bucket>
    <StorageClass>STANDARD_IA</StorageClass>
</Destination>
...
</Rule>

</ReplicationConfiguration>
```

#### Example 3: Configuração da replicação com duas regras e prefixos sobrepostos

Nessa configuração, as duas regras especificam filtros com prefixes de chaves sobrepostos, `star` / e `starship`. As duas regras se aplicam aos objetos com o keyname `starship-x`. Neste caso, o Amazon S3 usa a prioridade da regra para determinar qual regra deve ser aplicada.

```
<ReplicationConfiguration>

<Role>arn:aws:iam::AcctID:role/role-name</Role>

<Rule>
    <Status>Enabled</Status>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
        <Status>string</Status>
    </DeleteMarkerReplication>
    <Filter>
        <Prefix>star</Prefix>
    </Filter>
    <Destination>
        <Bucket>arn:aws:s3:::destinationbucket</Bucket>
    </Destination>
</Rule>
<Rule>
    <Status>Enabled</Status>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
        <Status>string</Status>
    </DeleteMarkerReplication>
    <Filter>
        <Prefix>starship</Prefix>
    </Filter>
    <Destination>
        <Bucket>arn:aws:s3:::destinationbucket</Bucket>
    </Destination>
</Rule>
</ReplicationConfiguration>
```

#### Example 4: Demonstrações de exemplo

Para obter exemplos de demonstrações, consulte [Demonstrações da replicação entre regiões \(CRR\) \(p. 567\)](#).

Para obter mais informações sobre a estrutura XML da configuração de replicação, consulte [PutBucketReplication](#) na Amazon Simple Storage Service API Reference.

## Compatibilidade retroativa

A versão mais recente do XML de configuração da replicação é V2. Para compatibilidade reversa, o Amazon S3 continua a oferecer suporte à configuração da V1. Se você tiver usado a V1 do XML de configuração da replicação, leve em consideração as questões a seguir que afetam a compatibilidade reversa:

- A V2 do XML de configuração da replicação inclui o elemento `Filter` para regras. Com o elemento `Filter`, você pode especificar filtros do objeto com base no prefixo de chaves do objeto, tags ou ambos para colocar dentro do escopo os objetos aos quais a regra se aplica. A V1 do XML de configuração da replicação oferecia suporte a filtros com base somente no prefixo de chaves. Nesse caso, você adiciona `Prefix` diretamente como elemento-filho do elemento `Rule`. Por exemplo,

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::AcctID:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>
    <Prefix>key-prefix</Prefix>
    <Destination><Bucket>arn:aws:s3:::destinationbucket</Bucket></Destination>
  </Rule>
</ReplicationConfiguration>
```

Para compatibilidade reversa, o Amazon S3 continua a oferecer suporte à configuração da V1.

- Ao excluir um objeto do bucket de origem sem especificar um ID da versão do objeto, o Amazon S3 adicionará um marcador de exclusão. Se você usar a V1 do XML de configuração da replicação, o Amazon S3 replicará os marcadores de exclusão que resultaram das ações do usuário. Em outras palavras: se o usuário tiver excluído o objeto, e não se o Amazon S3 tiver excluído porque o objeto expirou, como parte da ação do ciclo de vida. Na V2, o Amazon S3 não replica marcadores de exclusão e, portanto, você deve definir o elemento `DeleteMarkerReplication` como `Disabled`.

```
...
<Rule>
  <ID>Rule-1</ID>
  <Status>rule=Enabled-or-Diabled</Priority>
  <Priority>integer</Status>
  <DeleteMarkerReplication>
    <Status>Disabled</Status>
  </DeleteMarkerReplication>
  <Destination>
    <Bucket>arn:aws:s3:::bucket-name</Bucket>
  </Destination>
</Rule>
...
```

## Configurar permissões para CRR

Ao definir replicação entre regiões, é preciso adquirir as permissões necessárias, da seguinte forma:

- Crie uma função do IAM — O Amazon S3 precisa de permissão para replicar objetos em seu nome. Você concede essas permissões criando uma função do IAM e especificando a função na configuração da replicação.
- Quando os buckets de origem e de destino não forem de propriedade da mesma conta, o proprietário do bucket de destino deverá conceder ao proprietário do bucket de origem as permissões para armazenar as réplicas.

## Tópicos

- [Criação de uma função do IAM \(p. 557\)](#)
- [Concessão de permissões quando os buckets de origem e de destino forem de propriedade de diferentes contas da AWS \(p. 559\)](#)

## Criação de uma função do IAM

Por padrão, todos os recursos do Amazon S3 — buckets, objetos e sub-recursos relacionados — são privados: somente o proprietário do recurso pode acessá-lo. Para ler os objetos do bucket de origem e replicá-los ao bucket de destino, o Amazon S3 precisa de permissões para executar essas tarefas. Você concede essas permissões criando uma função do IAM e, em seguida, especificando a função na configuração da replicação.

Esta seção explica a política de confiança e a política de permissão mínima obrigatória. As demonstrações de exemplo dão instruções passo a passo para criar uma função do IAM. Para obter mais informações, consulte [Demonstrações da replicação entre regiões \(CRR\) \(p. 567\)](#).

- Uma política de confiança na qual você identifica o Amazon S3 como o principal de serviço que pode assumir a função:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "Service": "s3.amazonaws.com"  
            },  
            "Action": "sts:AssumeRole"  
        }  
    ]  
}
```

Para obter mais informações sobre funções do IAM, consulte [Funções do IAM](#) no Guia do usuário do IAM.

- Uma política de acesso em que você concede à função permissões para as tarefas de replicação em seu nome. Quando o Amazon S3 assumir a função, ele terá as permissões que você especificar nessa política.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:GetReplicationConfiguration",  
                "s3>ListBucket"  
            ],  
            "Resource": [  
                "arn:aws:s3:::source-bucket"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:PutReplicationConfiguration"  
            ]  
        }  
    ]  
}
```

```
    "s3:GetObjectVersion",
    "s3:GetObjectVersionAcl",
    "s3:GetObjectVersionTagging"

],
"Resource": [
    "arn:aws:s3:::source-bucket/*"
]
},
{
    "Effect":"Allow",
    "Action": [
        "s3:ReplicateObject",
        "s3:ReplicateDelete",
        "s3:ReplicateTags"
    ],
    "Resource": "arn:aws:s3:::destination-bucket/*"
}
]
```

A política de acesso concede permissões a estas ações:

- `s3:GetReplicationConfiguration` e `s3>ListBucket` — As permissões para essas ações no bucket de origem permitem que o Amazon S3 recupere a configuração de replicação e liste o conteúdo do bucket (o modelo atual de permissões exige a permissão `s3>ListBucket` para acesso aos marcadores de exclusão).
- `s3GetObjectVersion` e `s3GetObjectVersionAcl` — As permissões para essas ações concedidas em todos os objetos permitem que o Amazon S3 obtenha uma versão específica do objeto e as listas de controle de acesso (ACL) associadas aos objetos.
- `s3:ReplicateObject` e `s3:ReplicateDelete` — As permissões para essas ações em objetos no bucket de destino permitem que o Amazon S3 replique objetos ou marcadores de exclusão para o bucket de destino. Para obter mais informações sobre marcadores de exclusão, consulte [Como a exclusão de operações afeta a CRR \(p. 546\)](#).

#### Note

As permissões para a ação `s3:ReplicateObject` no bucket de destino também permitem replicação de tags dos objetos. Por isso, você não precisa conceder permissões explicitamente à ação `s3:ReplicateTags`.

- `s3GetObjectVersionTagging` — As permissões para esta ação em objetos no bucket de origem permitem que o Amazon S3 leia tags de objeto para replicação (consulte [Marcação de objetos \(p. 114\)](#)). Se o Amazon S3 não tiver essas permissões, ele replicará os objetos, mas não as tags do objeto.

Para obter uma lista das ações do Amazon S3 consulte [Especificação de permissões em uma política \(p. 330\)](#).

#### Important

A conta da AWS proprietária da função do IAM precisa ter permissões para as ações que conceder à função do IAM.

Por exemplo, suponha que o bucket de origem contenha objetos pertencentes a outra conta da AWS. O proprietário dos objetos deve conceder explicitamente à conta da AWS que é de propriedade da função do IAM as permissões exigidas por meio da ACL do objeto. Caso contrário, o Amazon S3 não conseguirá acessar os objetos e haverá falha na replicação entre regiões dos objetos. Para obter informações sobre as permissões da ACL, consulte [Visão geral da Lista de controle de acesso \(ACL\) \(p. 390\)](#).

As permissões descritas aqui estão relacionadas à configuração mínima da replicação. Se você optar por adicionar configurações de replicação opcionais, vai precisar conceder permissões

adicionais ao Amazon S3. Para obter mais informações, consulte [Configurações adicionais de CRR \(p. 560\)](#).

## Concessão de permissões quando os buckets de origem e de destino forem de propriedade de diferentes contas da AWS

Quando os buckets de origem e de destino não forem de propriedade da mesma conta, o proprietário do bucket de destino também deverá adicionar uma política de bucket para conceder ao proprietário do bucket de origem a permissão para executar ações de replicação, da seguinte forma:

```
{  
    "Version": "2008-10-17",  
    "Id": "PolicyForDestinationBucket",  
    "Statement": [  
        {  
            "Sid": "1",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "SourceBucket-AcctID"  
            },  
            "Action": [  
                "s3:ReplicateDelete",  
                "s3:ReplicateObject"  
            ],  
            "Resource": "arn:aws:s3:::destinationbucket/*"  
        },  
        {  
            "Sid": "2",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "SourceBucket-AcctID"  
            },  
            "Action": "s3>List*",  
            "Resource": "arn:aws:s3:::destinationbucket"  
        }  
    ]  
}
```

Para ver um exemplo, consulte [Exemplo 2: Configurar CRR quando os buckets de origem e de destino forem de propriedade de contas da AWS diferentes \(p. 575\)](#).

Se os objetos no bucket de origem estiverem marcados, observe o seguinte:

- Se o proprietário do bucket de origem conceder ao Amazon S3 permissão para as ações `s3:GetObjectVersionTagging` e `s3:ReplicateTags` para replicação de tags de objeto (por meio da função do IAM), o Amazon S3 replicará as tags com os objetos. Para obter informações sobre a função do IAM, consulte [Criação de uma função do IAM \(p. 557\)](#).
- Se o proprietário do bucket de destino não quiser replicar as tags, ele poderá adicionar a seguinte declaração à política de bucket de destino para negar explicitamente permissão para a ação `s3:ReplicateTags`:

```
...  
    "Statement": [  
        {  
            "Effect": "Deny",  
            "Principal": {  
                "AWS": "arn:aws:iam::SourceBucket-AcctID:root"  
            },  
            "Action": ["s3:ReplicateTags"],  
            "Resource": "arn:aws:s3:::destinationbucket/  
        }  
    ]  
}
```

```
        "Resource": "arn:aws:s3:::destinationbucket/*"
    }
...
}
```

## Alteração da propriedade da réplica

Quando diferentes contas da AWS tiverem a propriedade dos buckets de origem de destino, você poderá dizer ao Amazon S3 para alterar a propriedade da réplica para a conta da AWS à qual pertence o bucket de destino. Isso se chama opção de substituição do proprietário. Para obter mais informações, consulte [Configuração adicional da CRR: alteração do proprietário da réplica \(p. 560\)](#).

# Configurações adicionais de CRR

Esta seção descreve configurações adicionais da replicação entre regiões. Para obter informações sobre a replicação principal, consulte [Visão geral da configuração da CRR \(p. 547\)](#).

### Tópicos

- [Configuração adicional da CRR: alteração do proprietário da réplica \(p. 560\)](#)
- [Outra configuração de CRR: replicar objetos criados com a criptografia do lado do servidor \(SSE\) usando chaves de criptografia gerenciadas pelo AWS KMS \(p. 563\)](#)

## Configuração adicional da CRR: alteração do proprietário da réplica

Na replicação entre regiões (CRR), por padrão o proprietário do objeto de origem também é proprietário da réplica. Quando os buckets de origem e destino forem de propriedade de diferentes contas da AWS, você poderá adicionar configurações opcionais para alterar a propriedade da réplica para a conta da AWS proprietária do bucket de destino. Você pode fazer essa opção, por exemplo, para restringir o acesso às réplicas de objeto. Isso é chamado de opção de substituição do proprietário da configuração de replicação. Esta seção explica somente as configurações adicionais relevantes. Para obter mais informações sobre como definir a configuração de replicação, consulte [Replicação entre regiões \(p. 544\)](#).

Para configurar a substituição do proprietário, faça o seguinte:

- Adicione a opção de substituição do proprietário à configuração da replicação para dizer ao Amazon S3 para alterar a propriedade da réplica.
- Conceda permissões ao Amazon S3 para alterar a propriedade da réplica.
- Adicione a permissão na política do bucket de destino para permitir a alteração da propriedade da réplica. Isso permite que o proprietário do bucket de destino aceite a propriedade das réplicas do objeto.

As seções a seguir descrevem como executar essas tarefas. Para ver um exemplo funcional com instruções detalhadas, consulte [Exemplo 3: Alteração do proprietário da réplica quando os buckets de origem e de destino forem de propriedade de diferentes contas da AWS \(p. 576\)](#).

### Adição da opção de substituição do proprietário à configuração da replicação

#### Warning

Adicione somente a opção de substituição do proprietário quando os buckets de origem e destino forem de propriedade de contas diferentes da AWS. O Amazon S3 não verifica se os buckets são

de propriedade de contas iguais ou diferentes. Se você adicionar a substituição do proprietário quando ambos os buckets pertencerem à mesma conta da AWS, o Amazon S3 aplicará a substituição do proprietário. Ele concede permissões completas ao proprietário do bucket de destino e não replica as atualizações subsequentes para a lista de controle de acesso (ACL) do objeto de origem. O proprietário da réplica pode alterar diretamente na ACL associada a uma réplica com uma solicitação PUT ACL, mas não por replicação.

Para especificar a opção de substituição do proprietário, adicione o seguinte ao elemento Destino:

- O elemento `AccessControlTranslation`, que diz ao Amazon S3 para alterar a propriedade da réplica
- O elemento `Account`, que especifica a conta da AWS do proprietário do bucket de destino

```
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  ...
  <Destination>
    ...
    <AccessControlTranslation>
      <Owner>Destination</Owner>
    </AccessControlTranslation>
    <Account>destination-bucket-owner-account-id</Account>
  </Destination>
</Rule>
</ReplicationConfiguration>
```

A configuração da replicação do exemplo a seguir diz ao Amazon S3 para replicar os objetos que têm o prefixo de chaves Tax ao bucket de destino e altere a propriedade das réplicas.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <ID>Rule-1</ID>
    <Priority>1</Priority>
    <Status>Enabled</Status>
    <Status>Enabled</Status>
    <DeleteMarkerReplication>
      <Status>Disabled</Status>
    </DeleteMarkerReplication>
    <Filter>
      <Prefix>Tax</Prefix>
    </Filter>
    <Destination>
      <Bucket>arn:aws:s3:::destination-bucket</Bucket>
      <Account>destination-bucket-owner-account-id</Account>
      <AccessControlTranslation>
        <Owner>Destination</Owner>
      </AccessControlTranslation>
    </Destination>
  </Rule>
</ReplicationConfiguration>
```

## Concessão de permissões ao Amazon S3 para alterar a propriedade da réplica

Conceda permissões do Amazon S3 para alterar a propriedade da réplica ao adicionar permissão para a ação `s3:ObjectOwnerOverrideToBucketOwner` na política da permissão associada à função do IAM. Esta é a função do IAM que você especificou na configuração da replicação que permite ao Amazon S3 assumir e replicar objetos em seu nome.

```
...
{
    "Effect": "Allow",
    "Action": [
        "s3:ObjectOwnerOverrideToBucketOwner"
    ],
    "Resource": "arn:aws:s3:::destination-bucket/*"
}
...
```

## Adição da permissão na política do bucket de destino para permitir a alteração da propriedade da réplica

O proprietário do bucket de destino deve conceder ao proprietário da permissão do bucket de origem para alterar a propriedade da réplica. A propriedade do bucket de destino concede ao proprietário do bucket de origem permissão para a ação `s3:ObjectOwnerOverrideToBucketOwner`. Isso permite que o proprietário do bucket de origem aceite a propriedade das réplicas do objeto. A declaração de política do bucket de exemplo a seguir mostra como fazer isso:

```
...
{
    "Sid": "1",
    "Effect": "Allow",
    "Principal": {"AWS": "source-bucket-account-id" },
    "Action": ["s3:ObjectOwnerOverrideToBucketOwner"],
    "Resource": "arn:aws:s3:::destination-bucket/*"
}
...
```

## Considerações adicionais

Ao configurar a opção de substituição da propriedade, aplicam-se as seguintes considerações:

- Por padrão, o proprietário do objeto de origem também é proprietário da réplica. O Amazon S3 replica a versão do objetivo e a ACL associada a ela.

Se você adicionar a substituição do proprietário, o Amazon S3 replicará somente a versão do objeto, não a ACL. Além disso, o Amazon S3 não replica as alterações subsequentes na ACL do objeto de origem. O Amazon S3 define a ACL na réplica que concede controle total ao proprietário do bucket de destino.

- Ao atualizar uma configuração de replicação para habilitar ou desabilitar a substituição do proprietário, ocorre o seguinte:
  - Se você adicionar a opção de substituição do proprietário à configuração da replicação

Quando o Amazon S3 replica uma versão do objeto, ele descarta a ACL associada ao objeto de origem. Em vez disso, ele define a ACL na réplica, dando o controle total ao proprietário do bucket de destino. Ele não replica as alterações subsequentes na ACL do objeto de origem. No entanto, essa alteração na ACL não se aplica às versões de objeto replicadas antes de você definir a opção de substituição do proprietário. As atualizações da ACL nos objetos de origem replicados antes da

substituição do proprietário foram definidas para continuarem a ser replicadas (porque o objeto e suas réplicas continuam a ter o mesmo proprietário).

- Se você remover a opção de substituição do proprietário da configuração da replicação

O Amazon S3 replica novos objetos que aparecem no bucket de origem e as ACLs associadas ao bucket de destino. Para objetos replicados antes de você ter removido a substituição do proprietário, o Amazon S3 não replicará as ACLs, pois a propriedade do objeto muda, de maneira que o Amazon S3 feito permanece em vigor. Em outras palavras: as ACLs colocaram a versão do objeto que foi replicada quando a substituição do proprietário tinha sido substituída para não continuarem a ser replicadas.

## Outra configuração de CRR: replicar objetos criados com a criptografia do lado do servidor (SSE) usando chaves de criptografia gerenciadas pelo AWS KMS

Por padrão, o Amazon S3 não replica objetos armazenados em repouso usando criptografia do lado do servidor com chaves gerenciadas pelo AWS KMS. Esta seção explica outras configurações que você adiciona para orientar o Amazon S3 a replicar esses objetos.

Para ver um exemplo com instruções detalhadas, consulte [Exemplo 4: Replicar objetos criptografados \(p. 580\)](#). Para obter informações sobre como criar uma configuração da replicação, consulte [Replicação entre regiões \(p. 544\)](#).

### Tópicos

- [Especificando informações adicionais na configuração de replicação \(p. 563\)](#)
- [Concessão de permissões adicionais para a função do IAM \(p. 564\)](#)
- [Concessão de permissões adicionais para cenários entre contas \(p. 567\)](#)
- [Considerações sobre o limite de transação do AWS KMS \(p. 567\)](#)

## Especificando informações adicionais na configuração de replicação

Na configuração de replicação, você faz o seguinte:

- Na configuração de Destino, adicione a chave AWS KMS que você quer que o Amazon S3 use para criptografar réplicas de objetos.
- Faça o "opt-in" explicitamente ao habilitar a replicação de objetos criptografados usando as chaves do AWS KMS adicionando o elemento SourceSelectionCriteria.

```
<ReplicationConfiguration>
  <Rule>
    ...
    <SourceSelectionCriteria>
      <SseKmsEncryptedObjects>
        <Status>Enabled</Status>
      </SseKmsEncryptedObjects>
    </SourceSelectionCriteria>

  <Destination>
```

```
...
<EncryptionConfiguration>
    <ReplicaKmsKeyId>AWS KMS key ID for the AWS region of the destination bucket.</ReplicaKmsKeyId>
</EncryptionConfiguration>
</Destination>
...
</Rule>
</ReplicationConfiguration>
```

#### Important

A chave AWS KMS deve ter sido criada na mesma região da AWS que o bucket de destino. A chave AWS KMS deve ser válida. A API de replicação do bucket PUT não verifica a validade das chaves AWS KMS. Se você usar uma chave inválida, receberá o código de status 200 OK como resposta, mas a replicação falhará.

O exemplo a seguir de uma configuração de replicação entre regiões que inclui os elementos de configuração opcionais:

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration>
    <Role>arn:aws:iam::account-id:role/role-name</Role>
    <Rule>
        <ID>Rule-1</ID>
        <Priority>1</Priority>
        <Status>Enabled</Status>
        <DeleteMarkerReplication>
            <Status>Disabled</Status>
        </DeleteMarkerReplication>
        <Filter>
            <Prefix>Tax</Prefix>
        </Filter>
        <Destination>
            <Bucket>arn:aws:s3:::destination-bucket</Bucket>
            <EncryptionConfiguration>
                <ReplicaKmsKeyId>The AWS KMS key ID for the AWS region of the destination bucket (S3 uses it to encrypt object replicas).</ReplicaKmsKeyId>
            </EncryptionConfiguration>
        </Destination>
        <SourceSelectionCriteria>
            <SseKmsEncryptedObjects>
                <Status>Enabled</Status>
            </SseKmsEncryptedObjects>
        </SourceSelectionCriteria>
    </Rule>
</ReplicationConfiguration>
```

Essa configuração de replicação tem uma regra. A regra se aplica a objetos com o prefixo de chaves Tax. O Amazon S3 usa o ID da chave do AWS KMS para criptografar essas réplicas de objetos.

## Concessão de permissões adicionais para a função do IAM

Para replicar objetos criados usando criptografia do lado do servidor com chaves gerenciadas pelo AWS KMS, conceda as seguintes permissões adicionais à função do IAM que especificar na configuração da replicação. Você concede essas permissões ao atualizar a política de permissões associada à função do IAM:

- Permissão para a ação s3:GetObjectVersionForReplication de objetos de origem. Com a permissão para esta ação, o Amazon S3 consegue replicar tanto objetos não criptografados quanto

objetos criados com criptografia do lado do servidor por meio de chaves de criptografia gerenciadas pelo Amazon S3 (SSE-S3) ou chaves de criptografia gerenciadas pelo AWS KMS (SSE-KMS).

Note

Recomendamos que você use a ação `s3:GetObjectVersionForReplication` em vez da ação `s3:GetObjectVersion`, pois ela fornece ao Amazon S3 somente as permissões mínimas necessárias para a replicação entre regiões. Além disso, a permissão para a ação `s3:GetObjectVersion` permite a replicação de objetos não criptografados e objetos criptografados com SSE-S3, mas não objetos criados usando uma chave de criptografia gerenciada por AWS KMS.

- Permissões para as seguintes ações do AWS KMS:

- Permissões `kms:Decrypt` para a chave do AWS KMS utilizada na criptografia de objetos de origem
- Permissões `kms:Encrypt` para a chave do AWS KMS utilizada na criptografia da réplica de objetos

Recomendamos que você restrinja essas permissões a buckets e objetos específicos usando as chaves de condição do AWS KMS, conforme exibido nas instruções de política deste exemplo:

```
{  
    "Action": ["kms:Decrypt"],  
    "Effect": "Allow",  
    "Condition": {  
        "StringLike": {  
            "kms:ViaService": "s3.source-bucket-region.amazonaws.com",  
            "kms:EncryptionContext:aws:s3:arn": [  
                "arn:aws:s3:::source-bucket-name/key-prefix1*",  
            ]  
        }  
    },  
    "Resource": [  
        "List of AWS KMS key IDs used to encrypt source objects.",  
    ]  
},  
{  
    "Action": ["kms:Encrypt"],  
    "Effect": "Allow",  
    "Condition": {  
        "StringLike": {  
            "kms:ViaService": "s3.destination-bucket-region.amazonaws.com",  
            "kms:EncryptionContext:aws:s3:arn": [  
                "arn:aws:s3:::destination-bucket-name/key-prefix1*",  
            ]  
        }  
    },  
    "Resource": [  
        "AWS KMS key IDs (for the AWS region of the destination bucket). S3 uses it to encrypt object replicas",  
    ]  
}
```

A conta da AWS proprietária da função do IAM precisa ter permissões para estas ações do AWS KMS (`kms:Encrypt` e `kms:Decrypt`) para chaves do AWS KMS listadas na política. Se as chaves do AWS KMS pertencerem a outra conta da AWS, o proprietário da chave precisará conceder essas permissões à conta da AWS proprietária da função do IAM. Para obter mais informações sobre como gerenciar o acesso a essas chaves, consulte [Usar políticas do IAM com o AWS KMS](#) no AWS Key Management Service Developer Guide.

Veja a seguir uma política completa do IAM que concede as permissões necessárias para replicar objetos não criptografados, objetos criados com criptografia do lado do servidor por meio de chaves de criptografia gerenciadas pelo Amazon S3 e as chaves de criptografia gerenciadas pelo AWS KMS.

### Note

Objetos criados com criptografia do lado do servidor usando as chaves de criptografia fornecidas pelo usuário (SSE-C) não serão replicados.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:GetReplicationConfiguration",  
                "s3>ListBucket"  
            ],  
            "Resource": [  
                "arn:aws:s3:::source-bucket"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:GetObjectVersionForReplication",  
                "s3:GetObjectVersionAcl"  
            ],  
            "Resource": [  
                "arn:aws:s3:::source-bucket/key-prefix1*"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:ReplicateObject",  
                "s3:ReplicateDelete"  
            ],  
            "Resource": "arn:aws:s3:::destination-bucket/key-prefix1*"  
        },  
        {  
            "Action": [  
                "kms:Decrypt"  
            ],  
            "Effect": "Allow",  
            "Condition": {  
                "StringLike": {  
                    "kms:ViaService": "s3.source-bucket-region.amazonaws.com",  
                    "kms:EncryptionContext:aws:s3:arn": [  
                        "arn:aws:s3:::source-bucket-name/key-prefix1*"  
                    ]  
                }  
            },  
            "Resource": [  
                "List of AWS KMS key IDs used to encrypt source objects."  
            ]  
        },  
        {  
            "Action": [  
                "kms:Encrypt"  
            ],  
            "Effect": "Allow",  
            "Condition": {  
                "StringLike": {  
                    "kms:ViaService": "s3.destination-bucket-region.amazonaws.com",  
                    "kms:EncryptionContext:aws:s3:arn": [  
                        "arn:aws:s3:::destination-bucket-name/prefix1*"  
                    ]  
                }  
            }  
        }  
    ]  
}
```

```
        ]
    }
},
"Resource": [
    "AWS KMS key IDs (for the AWS region of the destination bucket) to use for
    encrypting object replicas"
]
}
```

## Concessão de permissões adicionais para cenários entre contas

Em um cenário entre contas, no qual os buckets de *origem* e *destino* são de propriedade de diferentes contas da AWS, a chave do AWS KMS para criptografar réplicas de objetos deve ser uma chave-mestra do cliente (CMK). O proprietário da chave deve conceder ao proprietário do bucket de origem permissão para usar a chave.

Para conceder ao proprietário do bucket de origem permissão para usar a chave (console do IAM)

1. Faça login no Console de gerenciamento da AWS e abra o console da IAM em <https://console.aws.amazon.com/iam/>.
2. Escolha Encryption keys (Chaves de criptografia).
3. Escolha a chave AWS KMS.
4. Em Key Policy (Política de chaves), Key Users (Usuários da chave), External Accounts (Contas externas), escolha Add External Account (Adicionar conta externa).
5. Para arn:aws:iam::, digite o ID da conta do bucket de origem.
6. Escolha Save Changes (Salvar alterações).

Para conceder ao proprietário do bucket de origem permissão para usar a chave (AWS CLI)

- Para obter mais informações, consulte [put-key-policy](#) na Referência de comando da AWS CLI. Para obter informações sobre a API subjacente, consulte [PutKeyPolicy](#) em [AWS Key Management Service API Reference](#).

## Considerações sobre o limite de transação do AWS KMS

Ao adicionar muitos novos objetos com a criptografia do AWS KMS depois de habilitar replicação entre regiões (CRR), você pode experimentar limitação (erros de respostas HTTP 503 recebidas com lentidão). A limitação acontece quando o número de transações do KMS por segundo excede o limite atual. Para obter mais informações, consulte [Limites](#) no AWS Key Management Service Developer Guide.

Recomendamos que você solicite um aumento no limite da taxa de API do AWS KMS criando um caso no AWS Support Center. Para obter mais informações, consulte <https://console.aws.amazon.com/support/home#/>.

## Demonstrações da replicação entre regiões (CRR)

Os exemplos a seguir mostram como configurar replicação entre regiões (CRR) para casos de uso comuns. Os exemplos mostram a definição da configuração de replicação usando o console do Amazon S3, a interface da linha de comando (CLI) da AWS e os AWS SDKs (estão exibidos exemplos de Java e .NET). Para obter informações sobre como instalar e configurar a AWS CLI, veja os tópicos a seguir no Guia do usuário do AWS Command Line Interface.

- [Instalar a interface de linha de comando da AWS](#)
- [Configuração da AWS CLI](#) – Você precisa configurar pelo menos um perfil. Você vai precisar definir dois perfis se estiver explorando cenários entre contas.

Para obter informações sobre o AWS SDK, consulte [AWS SDK para Java](#) e [AWS SDK para .NET](#).

#### Tópicos

- [Exemplo 1: Configurar a CRR quando os buckets de origem e de destino forem de propriedade da mesma conta da AWS \(p. 568\)](#)
- [Exemplo 2: Configurar CRR quando os buckets de origem e de destino forem de propriedade de contas da AWS diferentes \(p. 575\)](#)
- [Exemplo 3: Alteração do proprietário da réplica quando os buckets de origem e de destino forem de propriedade de diferentes contas da AWS \(p. 576\)](#)
- [Exemplo 4: Replicar objetos criptografados \(p. 580\)](#)

## Exemplo 1: Configurar a CRR quando os buckets de origem e de destino forem de propriedade da mesma conta da AWS

Neste exemplo, você configurará a replicação entre regiões (CRR) em que os buckets de origem e destino pertençam a contas das mesmas contas da AWS. Fornecemos exemplos para usar o console do Amazon S3, a AWS Command Line Interface (AWS CLI) e AWS SDK for Java e AWS SDK para .NET.

### Configurar a CRR quando os buckets de origem e de destino forem de propriedade da mesma conta da AWS (Console)

Para instruções passo a passo, consulte [Como adiciono uma regra de replicação entre regiões \(CRR\) a um bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service. Este tópico traz instruções para definir a configuração da replicação quando os buckets forem de propriedade de contas iguais e diferentes da AWS.

### Configurar a CRR quando os buckets de origem e de destino forem de propriedade da mesma conta da AWS (AWS CLI)

Para ajustar a CRR quando os buckets de origem e de destino forem de propriedade da mesma conta da AWS com a AWS CLI, crie buckets de origem e destino, habilite o versionamento neles, crie uma função do IAM que dê permissão ao Amazon S3 de replicar objetos e adicione a configuração de replicação ao bucket de origem. Para verificar sua configuração, teste-a.

Para configurar sua replicação da CRR quando os buckets de origem e de destino forem de propriedade da mesma conta da AWS

1. Defina um perfil de credenciais para a AWS CLI. Neste exemplo, usamos o nome de perfil `acctA`. Para obter mais informações sobre a definição de perfis da credencial, consulte [Perfis nomeados](#) no Guia do usuário do AWS Command Line Interface.

#### Important

O perfil que você usar para este exercício deve ter as permissões necessárias. Por exemplo, na configuração da replicação, especifique a função do IAM que o Amazon S3 pode assumir. Você só pode fazer isso se o perfil usado tiver a permissão `iam:PassRole`. Para obter mais informações, consulte [Conceder permissões ao usuário para aprovar uma função para](#)

um serviço da AWS, no Guia do usuário do IAM. Se você usar as credenciais do usuário administrador para criar um perfil nomeado, você pode executar todas as tarefas.

- Crie um bucket de **origem** e habilite o versionamento nele. O código a seguir cria um bucket de **origem** na Região Leste dos EUA (Norte da Virgínia) (us-east-1).

```
aws s3api create-bucket \  
--bucket source \  
--region us-east-1 \  
--profile acctA
```

```
aws s3api put-bucket-versioning \  
--bucket source \  
--versioning-configuration Status=Enabled \  
--profile acctA
```

- Crie um bucket de **destino** e habilite o versionamento nele. O código a seguir cria um bucket de **destino** na Região Oeste dos EUA (Oregon) (us-west-2).

Note

Para fazer a configuração da replicação quando os buckets de origem e destino estiverem na mesma conta da AWS, use o mesmo perfil. Neste exemplo, usamos acctA. Para testar a configuração da replicação quando os buckets forem de propriedade de diferentes contas da AWS, especifique diferentes perfis para cada um. Neste exemplo, usamos o perfil acctB para o bucket de destino.

```
aws s3api create-bucket \  
--bucket destination \  
--region us-west-2 \  
--create-bucket-configuration LocationConstraint=us-west-2 \  
--profile acctA
```

```
aws s3api put-bucket-versioning \  
--bucket destination \  
--versioning-configuration Status=Enabled \  
--profile acctA
```

- Crie uma função do IAM. Você especifica esta função na configuração da replicação que adicionar ao bucket de **origem** depois. O Amazon S3 assume essa função para replicar objetos em seu nome. A função do IAM é criada em duas etapas:

- Criar uma função
- Anexar uma política de permissões à função.

- Crie a função do IAM.

- Copie a política de confiança a seguir e salve-a em um arquivo com o nome **S3-role-trust-policy.json** no diretório atual do seu computador local. Essa política concede ao Amazon S3 permissões de serviço principal para assumir a função.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "Service": "s3.amazonaws.com"
```

```
        },
        "Action":"sts:AssumeRole"
    ]
}
```

- ii. Execute o seguinte comando para criar uma função:

```
$ aws iam create-role \
--role-name crrRole \
--assume-role-policy-document file://s3-role-trust-policy.json \
--profile acctA
```

- b. Anexar uma política de permissões à função.

- i. Copie a política de permissões a seguir e salve-a em um arquivo com o nome s3-role-permissions-policy.json no diretório atual do seu computador local. Essa política concede permissões para várias ações de bucket e objeto do Amazon S3.

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect":"Allow",
            "Action":[
                "s3:GetObjectVersionForReplication",
                "s3:GetObjectVersionAcl"
            ],
            "Resource": [
                "arn:aws:s3:::source-bucket/*"
            ]
        },
        {
            "Effect":"Allow",
            "Action":[
                "s3>ListBucket",
                "s3:GetReplicationConfiguration"
            ],
            "Resource": [
                "arn:aws:s3:::source-bucket"
            ]
        },
        {
            "Effect":"Allow",
            "Action":[
                "s3:ReplicateObject",
                "s3:ReplicateDelete",
                "s3:ReplicateTags",
                "s3:GetObjectVersionTagging"
            ],
            "Resource": "arn:aws:s3:::destination-bucket/*"
        }
    ]
}
```

- ii. Execute o comando a seguir para criar uma política e ligá-la à função:

```
$ aws iam put-role-policy \
--role-name crrRole \
--policy-document file://s3-role-permissions-policy.json \
--policy-name crrRolePolicy \
--profile acctA
```

5. Adicione uma configuração de replicação ao bucket de *origem*.

- Embora a API do Amazon S3 exija configuração de replicação como XML, a AWS CLI exige que você especifique a configuração da replicação como JSON. Salve o JSON a seguir em um arquivo chamado `replication.json` no diretório local do seu computador.

```
{  
    "Role": "IAM-role-ARN",  
    "Rules": [  
        {  
            "Status": "Enabled",  
            "Priority": 1,  
            "DeleteMarkerReplication": { "Status": "Disabled" },  
            "Filter": { "Prefix": "Tax" },  
            "Destination": {  
                "Bucket": "arn:aws:s3:::destination-bucket"  
            }  
        }  
    ]  
}
```

- Atualize o JSON fornecendo valores para *destination-bucket* e *IAM-role-ARN*. Salve as alterações.
- Execute o comando a seguir para adicionar a configuração de replicação ao seu bucket de origem. Não deixe de dar um nome ao *source-bucket*.

```
$ aws s3api put-bucket-replication \  
--replication-configuration file://replication.json \  
--bucket source \  
--profile acctA
```

Para recuperar a configuração de replicação, use o comando `get-bucket-replication`:

```
$ aws s3api get-bucket-replication \  
--bucket source \  
--profile acctA
```

6. Teste a configuração no console do Amazon S3:

- Cadastre-se no Console de Gerenciamento da AWS e abra o console da Amazon S3 em <https://console.aws.amazon.com/s3/>
- No bucket de *origem*, crie uma pasta chamada Tax.
- Adicione objetos de amostra à pasta Tax no bucket de *origem*.

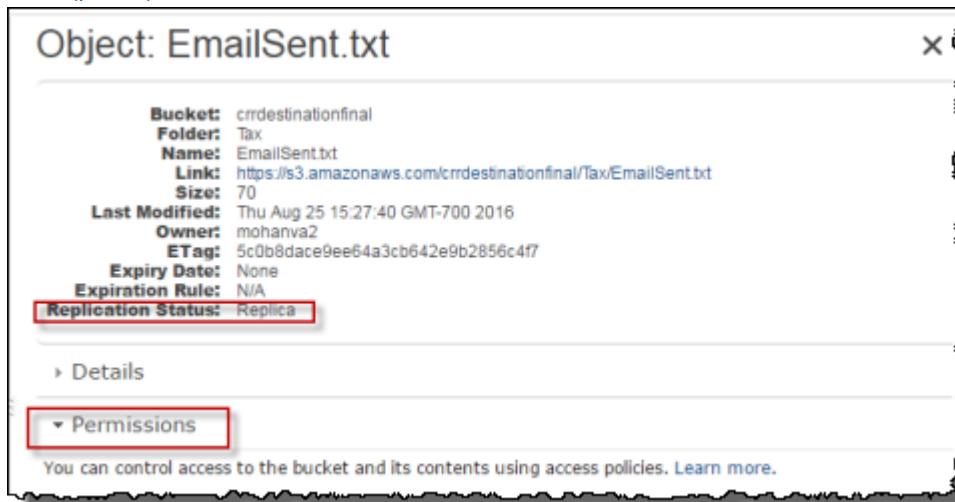
Note

O tempo que o Amazon S3 leva para replicar um objeto depende do tamanho do objeto. Para obter informações sobre como ver o status da replicação, consulte [Informação sobre o status da replicação entre regiões \(p. 585\)](#).

No bucket de *destino*, verifique o seguinte:

- Que o Amazon S3 replicou os objetos.
- Nas properties (propriedades) do objeto, que Replication Status (Status de replicação) está definido como Replica (identificando-o como um objeto de réplica).
- Nas properties (propriedades) do objeto, que a seção de permissão não mostra nenhuma permissão. Isso significa que a réplica ainda é de propriedade do proprietário do bucket de *origem* e que o proprietário do bucket de *destino* não tem permissão na réplica do

objeto. Você pode adicionar uma configuração opcional para orientar o Amazon S3 a alterar a propriedade da réplica. Para ver um exemplo, consulte [Exemplo 3: Alteração do proprietário da réplica quando os buckets de origem e de destino forem de propriedade de diferentes contas da AWS \(p. 576\)](#).



- d. Atualize a ACL de um objeto no bucket de *origem* e verifique se as alterações aparecem no bucket de *destino*.

Para obter instruções, consulte [Como defino permissões em um objeto?](#) no Guia do usuário do console do Amazon Simple Storage Service.

## Configurar a CRR quando os buckets de origem e de destino forem de propriedade da mesma conta da AWS (SDK da AWS)

Use os exemplos de código a seguir para adicionar uma configuração de replicação ao bucket com AWS SDK for Java e AWS SDK para .NET, respectivamente.

### Java

O exemplo a seguir adiciona uma configuração de replicação a um bucket e, depois, a recupera e verifica a configuração. Para obter instruções sobre criar e testar um exemplo funcional, consulte [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

import java.io.IOException;
import java.util.HashMap;
import java.util.Map;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.BucketReplicationConfiguration;
import com.amazonaws.services.s3.model.ReplicationDestinationConfig;
import com.amazonaws.services.s3.model.ReplicationRule;
import com.amazonaws.services.s3.model.ReplicationRuleStatus;
import com.amazonaws.services.s3.model.StorageClass;
```

```
public class CrossRegionReplication {

    public static void main(String[] args) throws IOException {
        String clientRegion = "*** Client region ***";
        String accountId = "*** Account ID ***";
        String roleName = "*** Role name ***";
        String sourceBucketName = "*** Source bucket name ***";
        String destBucketName = "*** Destination bucket name ***";
        String prefix = "Tax/";

        String roleARN = String.format("arn:aws:iam::%s:role/%s", accountId, roleName);
        String destinationBucketARN = "arn:aws:s3:::" + destBucketName;

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

            // Create the replication rule.
            List<ReplicationFilterPredicate> andOperands = new
ArrayList<ReplicationFilterPredicate>();
            andOperands.add(new ReplicationPrefixPredicate("prefix"));

            Map<String, ReplicationRule> replicationRules = new HashMap<String,
ReplicationRule>();
            replicationRules.put("ReplicationRule1",
                new ReplicationRule()
                    .withPriority(0)
                    .withStatus(ReplicationRuleStatus.Enabled)
                    .withDeleteMarkerReplication(new
DeleteMarkerReplication().withStatus(DeleteMarkerReplicationStatus.DISABLED))
                    .withFilter(new
ReplicationFilter().withPredicate(new ReplicationAndOperator(andOperands)))
                    .withDestinationConfig(new
ReplicationDestinationConfig()
                        .withBucketARN(destinationBucketARN)
                        .withStorageClass(StorageClass.Standard)));
            // Save the replication rule to the source bucket.
            s3Client.setBucketReplicationConfiguration(sourceBucketName,
                new
BucketReplicationConfiguration()
                    .withRoleARN(roleARN)
                    .withRules(replicationRules));

            // Retrieve the replication configuration and verify that the configuration
            // matches the rule we just set.
            BucketReplicationConfiguration replicationConfig =
s3Client.getBucketReplicationConfiguration(sourceBucketName);
            ReplicationRule rule = replicationConfig.getRule("ReplicationRule1");
            System.out.println("Retrieved destination bucket ARN: " +
rule.getDestinationConfig().getBucketARN());
            System.out.println("Retrieved source-bucket replication rule prefix: " +
rule.getPrefix());
            System.out.println("Retrieved source-bucket replication rule status: " +
rule.getStatus());
        }
        catch(AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
        }
    }
}
```

```
        e.printStackTrace();
    }
    catch(SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
```

C#

O exemplo de código de AWS SDK para .NET a seguir adiciona uma configuração de replicação a um bucket e, depois, a recupera. Para usar esse código, dê nomes aos seus buckets e o nome de recurso da Amazon (ARN) para sua função do IAM. Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

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

namespace Amazon.DocSamples.S3
{
    class CrossRegionReplicationTest
    {
        private const string sourceBucket = "*** source bucket ***";
        // Bucket ARN example - arn:aws:s3:::destinationbucket
        private const string destinationBucketArn = "*** destination bucket ARN ***";
        private const string roleArn = "*** IAM Role ARN ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint sourceBucketRegion =
RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
        public static void Main()
        {
            s3Client = new AmazonS3Client(sourceBucketRegion);
            EnableReplicationAsync().Wait();
        }
        static async Task EnableReplicationAsync()
        {
            try
            {
                ReplicationConfiguration replConfig = new ReplicationConfiguration
                {
                    Role = roleArn,
                    Rules =
                    {
                        new ReplicationRule
                        {
                            Prefix = "Tax",
                            Status = ReplicationRuleStatus.Enabled,
                            Destination = new ReplicationDestination
                            {
                                BucketArn = destinationBucketArn
                            }
                        }
                    };
            }
        }
    }
}
```

```
PutBucketReplicationRequest putRequest = new
PutBucketReplicationRequest
{
    BucketName = sourceBucket,
    Configuration = replConfig
};

PutBucketReplicationResponse putResponse = await
s3Client.PutBucketReplicationAsync(putRequest);

// Verify configuration by retrieving it.
await RetrieveReplicationConfigurationAsync(s3Client);
}

catch (AmazonS3Exception e)
{
    Console.WriteLine("Error encountered on server. Message:'{0}' when
writing an object", e.Message);
}
catch (Exception e)
{
    Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
}
}

private static async Task RetrieveReplicationConfigurationAsync(IAmazonS3
client)
{
    // Retrieve the configuration.
    GetBucketReplicationRequest getRequest = new GetBucketReplicationRequest
    {
        BucketName = sourceBucket
    };
    GetBucketReplicationResponse getResponse = await
client.GetBucketReplicationAsync(getRequest);
    // Print.
    Console.WriteLine("Printing replication configuration information...");
    Console.WriteLine("Role ARN: {0}", getResponse.Configuration.Role);
    foreach (var rule in getResponse.Configuration.Rules)
    {
        Console.WriteLine("ID: {0}", rule.Id);
        Console.WriteLine("Prefix: {0}", rule.Prefix);
        Console.WriteLine("Status: {0}", rule.Status);
    }
}
}
```

## Exemplo 2: Configurar CRR quando os buckets de origem e de destino forem de propriedade de contas da AWS diferentes

A definição de replicação entre regiões (CRR) quando os buckets de *origem* e *destino* forem de propriedade de diferentes contas da AWS é semelhante à definição de uma CRR quando os dois buckets forem de propriedade da mesma conta. A única diferença é que o proprietário do bucket de *destino* deve conceder ao proprietário do bucket de *origem* permissão para replicar objetos ao adicionar uma política do bucket.

Para configurar o CRR quando os buckets de origem e de destino forem de propriedade de diferentes contas da AWS

1. Neste exemplo, crie os buckets de *origem* e *destino* em duas contas diferentes da AWS. Você precisa ter dois perfis de credencial definidos para a AWS CLI (neste exemplo, usamos os nomes de perfil `acctA` e `acctB`). Para obter mais informações sobre como definir perfis da credencial, consulte [Perfis nomeados](#) no Guia do usuário do AWS Command Line Interface.
2. Siga instruções passo a passo em [Exemplo 1 de CRR: Mesma conta da AWS \(p. 568\)](#).com as seguintes alterações:
  - Para todos os comandos da CLI relacionados a atividades do bucket de *origem* (para criar o bucket de *origem*, habilitar versionamento e criar a função do IAM), use o perfil `acctA`. Use o perfil `acctB` para criar o bucket de *destino*.
  - Verifique se a política e permissões especifica os buckets de *origem* e *destino* que você criou para este exemplo.
3. No console do AWS, adicione a seguinte política do bucket ao bucket de *destino* para permitir que o proprietário do bucket de *origem* replique objetos. Não deixe de editar a política ao fornecer o ID da conta da AWS do proprietário do bucket de *origem* e o nome do bucket de *destino*.

```
{  
    "Version": "2008-10-17",  
    "Id": "",  
    "Statement": [  
        {  
            "Sid": "Stmt123",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::source-bucket-owner-AWS-acct-ID:root"  
            },  
            "Action": [ "s3:ReplicateObject", "s3:ReplicateDelete" ],  
            "Resource": "arn:aws:s3:::destination/*"  
        }  
    ]  
}
```

Escolha o bucket e adicione a política do bucket. Para obter instruções, consulte [Como adicionar uma política de bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

## Exemplo 3: Alteração do proprietário da réplica quando os buckets de origem e de destino forem de propriedade de diferentes contas da AWS

Quando os buckets de *origem* e *destino* na configuração da replicação entre regiões (CRR) forem de propriedade de diferentes contas da AWS, você poderá dizer ao Amazon S3 para alterar a propriedade da réplica para a conta da AWS que é proprietária do bucket de *destino*. Este exemplo explica como usar o console do Amazon S3 e a AWS CLI para alterar a propriedade da réplica. Para obter mais informações, consulte [Configuração adicional da CRR: alteração do proprietário da réplica \(p. 560\)](#).

### Alteração do proprietário da réplica quando os buckets de origem e de destino forem de propriedade de diferentes contas da AWS (console)

Para instruções passo a passo, consulte [Configurar uma regra CRR quando o bucket de destino estiver em outra conta da AWS](#) no Guia do usuário do console do Amazon Simple Storage Service.

## Alteração do proprietário da réplica quando os buckets de origem e de destino forem de propriedade de diferentes contas da AWS (AWS CLI)

Para alterar a propriedade da réplica usando a AWS CLI, crie buckets, habilite o versionamento neles, crie uma função do IAM que dê permissão ao Amazon S3 de replicar objetos e adicione a configuração da replicação ao bucket de origem. Na configuração da replicação, você orienta o Amazon S3 a alterar o proprietário da réplica. Você também testa a configuração.

Para alterar o proprietário da réplica quando os buckets de origem e de destino forem de propriedade de diferentes contas da AWS (AWS CLI)

- Neste exemplo, você cria os buckets de *origem* e *destino* em duas contas diferentes da AWS. Configurar a AWS CLI com dois perfis nomeados. Neste exemplo, usamos os perfis nomeados `acctA` e `acctB`, respectivamente. Para obter mais informações sobre como definir perfis da credencial, consulte [Perfis nomeados](#) no Guia do usuário do AWS Command Line Interface.

### Important

Os perfis que você usar para este exercício deve ter as permissões necessárias. Por exemplo, na configuração da replicação, especifique a função do IAM que o Amazon S3 pode assumir. Você só pode fazer isso se o perfil usado tiver a permissão `iam:PassRole`. Se você usar as credenciais do usuário administrador para criar um perfil nomeado, você pode executar todas as tarefas. Para obter mais informações, consulte [Conceder permissões ao usuário para aprovar uma função para um serviço da AWS](#), no Guia do usuário do IAM.

Você precisa garantir que esses perfis tenham as permissões necessárias. Por exemplo, a configuração de replicação inclui um IAM que o Amazon S3 pode assumir. O perfil nomeado que você usa para conectar essa configuração a um bucket só poderá fazer isso se tiver a permissão `iam:PassRole`. Se você especificar as credenciais do usuário administrador ao criar esses perfis nomeados, eles terão todas as permissões. Para obter mais informações, consulte [Conceder permissões ao usuário para aprovar uma função para um serviço da AWS](#), no Guia do usuário do IAM.

- Crie o bucket de *origem* e habilite o versionamento. Neste exemplo, criamos o bucket de *origem* na região Leste dos EUA (Norte da Virgínia) (us-east-1).

```
aws s3api create-bucket \
--bucket source \
--region us-east-1 \
--profile acctA
```

```
aws s3api put-bucket-versioning \
--bucket source \
--versioning-configuration Status=Enabled \
--profile acctA
```

- Crie um bucket de *destino* e habilite o versionamento. Neste exemplo, criamos o bucket de *destino* na região Oeste dos EUA (Oregon) (us-west-2). Use um perfil de conta da AWS diferente do usado para o bucket de *origem*.

```
aws s3api create-bucket \
--bucket destination \
--region us-west-2 \
--create-bucket-configuration LocationConstraint=us-west-2 \
--profile acctB
```

```
aws s3api put-bucket-versioning \
--bucket destination \
--versioning-configuration Status=Enabled \
```

```
--profile acctB
```

4. Crie uma função do IAM. Você especifica esta função na configuração da replicação que adicionar ao bucket de **origem** depois. O Amazon S3 assume essa função para replicar objetos em seu nome. A função do IAM é criada em duas etapas:

- Criar uma função
- Anexar uma política de permissões à função

- a. Crie uma função do IAM.

- i. Copie a política de confiança a seguir e salve-a em um arquivo com o nome **S3-role-trust-policy.json** no diretório atual do seu computador local. Essa política concede ao Amazon S3 permissões para assumir a função.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "Service": "s3.amazonaws.com"  
            },  
            "Action": "sts:AssumeRole"  
        }  
    ]  
}
```

- ii. Execute o seguinte comando de AWS CLI para criar uma função:

```
$ aws iam create-role \  
--role-name crrRole \  
--assume-role-policy-document file://s3-role-trust-policy.json \  
--profile acctA
```

- b. Anexe uma política de permissões à função.

- i. Copie a política de permissões a seguir e salve-a em um arquivo com o nome **s3-role-perm-pol-changeowner.json** no diretório atual do seu computador local. Essa política concede permissões para várias ações de bucket e objeto do Amazon S3. Nas etapas a seguir, crie uma função do IAM e anexe esta política à função.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:GetObjectVersionForReplication",  
                "s3:GetObjectVersionAcl"  
            ],  
            "Resource": [  
                "arn:aws:s3:::source/*"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3>ListBucket",  
                "s3:GetReplicationConfiguration"  
            ],  
            "Resource": "source/*"  
        }  
    ]  
}
```

```
        "Resource": [
            "arn:aws:s3:::source"
        ],
    },
{
    "Effect": "Allow",
    "Action": [
        "s3:ReplicateObject",
        "s3:ReplicateDelete",
        "s3:ObjectOwnerOverrideToBucketOwner",
        "s3:ReplicateTags",
        "s3:GetObjectVersionTagging"
    ],
    "Resource": "arn:aws:s3:::destination/*"
}
]
```

- ii. Para criar uma política e ligá-la à função, execute o comando a seguir:

```
$ aws iam put-role-policy \
--role-name crrRole \
--policy-document file://s3-role-perm-pol-changeowner.json \
--policy-name crrRolechangeownerPolicy \
--profile acctA
```

5. Adicione a configuração de replicação ao bucket de origem.

- a. A AWS CLI requer que você especifique a configuração de replicação como JSON. Salve o JSON a seguir em um arquivo chamado `replication.json` no diretório atual local do seu computador local. Na configuração, a adição de `AccessControlTranslation` indica alteração na propriedade da réplica.

```
{
    "Role": "IAM-role-ARN",
    "Rules": [
        {
            "Status": "Enabled",
            "Priority": "1",
            "DeleteMarkerReplication": {
                "Status": "Disabled"
            },
            "Filter": {
                "Prefix": "Tax"
            },
            "Status": "Enabled",
            "Destination": {
                "Bucket": "arn:aws:s3:::destination",
                "Account": "destination-bucket-owner-account-id",
                "AccessControlTranslation": {
                    "Owner": "Destination"
                }
            }
        }
    ]
}
```

- b. Edite o JSON fornecendo os valores do ID da conta do proprietário do bucket de `destino` e `IAM-role-ARN`. Salve as alterações.  
c. Para adicionar a configuração de replicação ao bucket de origem, execute o comando a seguir. Dê o nome do bucket de `origem`.

```
$ aws s3api put-bucket-replication \
--replication-configuration file://replication-changeowner.json \
--bucket source \
--profile accta
```

6. Verifique a propriedade da réplica no console do Amazon S3.
  - a. Cadastre-se no Console de Gerenciamento da AWS e abra o console da Amazon S3 em <https://console.aws.amazon.com/s3/>.
  - b. No bucket de *origem*, crie uma pasta chamada Tax.
  - c. Adicione objetos à pasta no bucket de *origem*. Verifique se o bucket de *destino* contém as réplicas do objeto e se a propriedade das réplicas mudou para a conta da AWS proprietária do bucket de *destino*.

## Alteração do proprietário da réplica quando os buckets de origem e de destino forem de propriedade de diferentes contas da AWS (SDK da AWS)

Para exemplo de código para adicionar a configuração da replicação, consulte [Configurar a CRR quando os buckets de origem e de destino forem de propriedade da mesma conta da AWS \(SDK da AWS\) \(p. 572\)](#). Você precisará modificar a configuração de replicação de acordo. Para informações conceituais, consulte [Configuração adicional da CRR: alteração do proprietário da réplica \(p. 560\)](#).

## Exemplo 4: Replicar objetos criptografados

Por padrão, o Amazon S3 não replica objetos armazenados em repouso usando criptografia do lado do servidor com chaves gerenciadas pelo AWS KMS. Para replicar objetos criptografados, modifique a configuração de replicação do bucket para dizer ao Amazon S3 que replique esses objetos. Este exemplo explica como usar o console do Amazon S3 e o AWS Command Line Interface (AWS CLI) para alterar a configuração de replicação do bucket de maneira que permita a replicação de objetos criptografados. Para obter mais informações, consulte [Outra configuração de CRR: replicar objetos criados com a criptografia do lado do servidor \(SSE\) usando chaves de criptografia gerenciadas pelo AWS KMS \(p. 563\)](#).

### Replicação de objetos criptografados (console)

Para instruções passo a passo, consulte [Como adiciono uma regra de replicação entre regiões \(CRR\) a um bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service. Este tópico traz instruções para definir a configuração da replicação quando os buckets forem de propriedade de contas iguais e diferentes da AWS.

### Replicação de objetos criptografados (AWS CLI)

Para replicar objetos criptografados com a AWS CLI, crie buckets, habilite o versionamento neles, crie uma função do IAM que dê permissão ao Amazon S3 de replicar objetos e adicione a configuração da replicação ao bucket de origem. A configuração de replicação fornece informações relacionadas à replicação de objetos criptografados usando as chaves do KMS. A permissão da função do IAM inclui as permissões necessárias para replicar os objetos criptografados. Você também testa a configuração.

#### Para replicar objetos criptografados (AWS CLI)

1. Neste exemplo, criamos tanto os buckets de *origem* quanto de *destino* na mesma conta da AWS. Defina um perfil de credenciais para a AWS CLI. Neste exemplo, usamos o nome de perfil accta. Para obter mais informações sobre como definir perfis da credencial, consulte [Perfis nomeados](#) no Guia do usuário do AWS Command Line Interface.
2. Crie o bucket de *origem* e habilite o versionamento nele. Neste exemplo, criamos o bucket de *origem* na região Leste dos EUA (Norte da Virgínia) (us-east-1).

```
aws s3api create-bucket \  
--bucket source \  
--region us-east-1 \  
--profile acctA
```

```
aws s3api put-bucket-versioning \  
--bucket source \  
--versioning-configuration Status=Enabled \  
--profile acctA
```

- Crie o bucket de **destino** e habilite o versionamento nele. Neste exemplo, criamos o bucket de **destino** na região Oeste dos EUA (Oregon) (us-west-2).

Note

Para fazer a configuração da replicação quando os buckets de **origem** e **destino** estiverem na mesma conta da AWS, use o mesmo perfil. Neste exemplo, usamos acctA. Para testar a configuração da replicação quando os buckets forem de propriedade de diferentes contas da AWS, especifique diferentes perfis para cada um. Neste exemplo, usamos o perfil acctB para o bucket de **destino**.

```
aws s3api create-bucket \  
--bucket destination \  
--region us-west-2 \  
--create-bucket-configuration LocationConstraint=us-west-2 \  
--profile acctA
```

```
aws s3api put-bucket-versioning \  
--bucket destination \  
--versioning-configuration Status=Enabled \  
--profile acctA
```

- Crie uma função do IAM. Você especifica esta função na configuração da replicação que adicionar ao bucket de **origem** depois. O Amazon S3 assume essa função para replicar objetos em seu nome. A função do IAM é criada em duas etapas:

- Criar uma função
- Anexar uma política de permissões à função

- Crie uma função do IAM.

- Copie a política de confiança a seguir e salve-a em um arquivo com o nome **s3-role-trust-policy-kmsobj.json** no diretório atual do seu computador local. Essa política concede ao serviço do Amazon S3 as principais permissões para assumir a função, de maneira que o Amazon S3 possa executar tarefas em seu nome.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "Service": "s3.amazonaws.com"  
            },  
            "Action": "sts:AssumeRole"  
        }  
    ]}
```

}

- ii. Crie uma função:

```
$ aws iam create-role \
--role-name crrRolekmsobj \
--assume-role-policy-document file://s3-role-trust-policy-kmsobj.json \
--profile acctA
```

- b. Anexar uma política de permissões à função. Essa política concede permissões para várias ações de bucket e objeto do Amazon S3.
- i. Copie a política de permissões a seguir e salve-a em um arquivo com o nome s3-role-permissions-policykmsobj.json no diretório atual do seu computador local. Crie uma função do IAM e anexe a política à ela depois.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3>ListBucket",
                "s3:GetReplicationConfiguration",
                "s3:GetObjectVersionForReplication",
                "s3:GetObjectVersionAcl"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::source",
                "arn:aws:s3:::source/*"
            ]
        },
        {
            "Action": [
                "s3:ReplicateObject",
                "s3:ReplicateDelete",
                "s3:ReplicateTags",
                "s3:GetObjectVersionTagging"
            ],
            "Effect": "Allow",
            "Condition": {
                "StringLikeIfExists": {
                    "s3:x-amz-server-side-encryption": [
                        "aws:kms",
                        "AES256"
                    ],
                    "s3:x-amz-server-side-encryption-aws-kms-key-id": [
                        "AWS KMS key IDs to use for encrypting object replicas"
                    ]
                }
            },
            "Resource": "arn:aws:s3:::destination/*"
        },
        {
            "Action": [
                "kms:Decrypt"
            ],
            "Effect": "Allow",
            "Condition": {
                "StringLike": {
                    "kms:ViaService": "s3.us-east-1.amazonaws.com",
                    "kms:EncryptionContext:aws:s3:arn": [
                        "arn:aws:s3:::source/*"
                    ]
                }
            }
        }
    ]
}
```

```
        }
    },
    "Resource":[
        "AWS KMS key IDs used to encrypt source objects."
    ]
},
{
    "Action":[
        "kms:Encrypt"
    ],
    "Effect":"Allow",
    "Condition":{
        "StringLike":{
            "kms:ViaService":"s3.us-west-2.amazonaws.com",
            "kms:EncryptionContext:aws:s3:arn":[
                "arn:aws:s3:::destination/*"
            ]
        }
    },
    "Resource":[
        "AWS KMS key IDs to use for encrypting object replicas"
    ]
}
]
```

- ii. Crie uma política e anexe-a à função:

```
$ aws iam put-role-policy \
--role-name crrRolekmsobj \
--policy-document file://s3-role-permissions-policykmsobj.json \
--policy-name crrRolechangeownerPolicy \
--profile accta
```

5. Adicione a seguinte configuração de replicação ao bucket de *origem*. Isso diz ao Amazon S3 para replicar objetos com o prefixo *Tax/* para o bucket de *destino*.

**Important**

Na configuração da replicação, especifique a função do IAM que o Amazon S3 pode assumir. Você só poderá fazer isso se tiver a permissão `iam:PassRole`. O perfil especificado no comando da CLI deve ter a permissão. Para obter mais informações, consulte [Conceder permissões ao usuário para aprovar uma função para um serviço da AWS](#), no Guia do usuário do IAM.

```
<ReplicationConfiguration>
<Role>IAM-Role-ARN</Role>
<Rule>
    <Status>Enabled</Status>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
        <Status>Disabled</Status>
    </DeleteMarkerReplication>
    <Filter>
        <Prefix>Tax</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <SourceSelectionCriteria>
        <SseKmsEncryptedObjects>
            <Status>Enabled</Status>
        </SseKmsEncryptedObjects>
    </SourceSelectionCriteria>
    <Destination>
        <Bucket>arn:aws:s3:::dest-bucket-name</Bucket>
    </Destination>
</Rule>

```

```
<EncryptionConfiguration>
    <ReplicaKmsKeyID>AWS KMS key IDs to use for encrypting object replicas</ReplicaKmsKeyID>
</EncryptionConfiguration>
</Destination>
</Rule>
</ReplicationConfiguration>
```

Para adicionar a configuração de replicação ao bucket de *origem*, faça o seguinte:

- A AWS CLI requer que você especifique a configuração de replicação como JSON. Salve o JSON a seguir em um arquivo (*replication.json*) no diretório atual local do seu computador local.

```
{
    "Role": "IAM-Role-ARN",
    "Rules": [
        {
            "Status": "Enabled",
            "Priority": "1",
            "DeleteMarkerReplication": {
                "Status": "Disabled"
            },
            "Filter": {
                "Prefix": "Tax"
            },
            "Destination": {
                "Bucket": "arn:aws:s3:::destination",
                "EncryptionConfiguration": {
                    "ReplicaKmsKeyID": "AWS KMS key IDs to use for encrypting object replicas"
                }
            },
            "SourceSelectionCriteria": {
                "SseKmsEncryptedObjects": {
                    "Status": "Enabled"
                }
            },
            "Status": "Enabled"
        }
    ]
}
```

- Edite o JSON para fornecer valores para o bucket de *destino* e para *IAM-role-ARN*. Salve as alterações.
- Adicione a configuração de replicação ao bucket de *origem*. Não deixe de dar um nome ao bucket de *origem*.

```
$ aws s3api put-bucket-replication \
--replication-configuration file://replicationkmsobj.json \
--bucket source \
--profile acctA
```

- Teste a configuração para verificar se os objetos criptografados estão replicados. No console do Amazon S3:
  - Cadastre-se no Console de Gerenciamento da AWS e abra o console da Amazon S3 em <https://console.aws.amazon.com/s3/>.
  - No bucket de *origem*, crie uma pasta chamada Tax.
  - Adicione objetos de amostra à pasta. Não se esqueça de escolher a opção de criptografia e especificar a chave do KMS para criptografar os objetos.

- d. Verifique se o bucket de *destino* contém as réplicas do objeto e se elas são criptografadas usando a chave de criptografia do KMS especificada na configuração.

## Replicação de objetos criptografados (AWS SDK)

Para exemplo de código para adicionar a configuração da replicação, consulte [Configurar a CRR quando os buckets de origem e de destino forem de propriedade da mesma conta da AWS \(SDK da AWS\) \(p. 572\)](#). Você precisará modificar a configuração de replicação de acordo. Para informações conceituais, consulte [Outra configuração de CRR: replicar objetos criados com a criptografia do lado do servidor \(SSE\) usando chaves de criptografia gerenciadas pelo AWS KMS \(p. 563\)](#).

# Informação sobre o status da replicação entre regiões

Para obter o status da replicação entre regiões (CRR) dos objetos dentro de um bucket, use a ferramenta de inventário do Amazon S3. O Amazon S3 envia um arquivo .csv ao bucket de destino que você especifica na configuração do inventário. Você também pode usar o Amazon Athena para consultar o status da replicação no relatório de inventário. Para obter mais informações sobre inventário do Amazon S3, consulte [Inventário do Amazon S3 \(p. 273\)](#).

Na CRR, você tem um bucket de origem em que configura a replicação e um bucket de destino onde o Amazon S3 replica objetos. Ao solicitar um objeto (usando GET objeto) ou metadados de objeto (usando HEAD objeto) nesses buckets, o Amazon S3 retornará o cabeçalho x-amz-replication-status na resposta da seguinte maneira:

- Ao solicitar um objeto no bucket de origem, o Amazon S3 retornará o cabeçalho x-amz-replication-status se o objeto em sua solicitação for qualificado para replicação.

Por exemplo, suponha que, em sua configuração de replicação, você especifique o prefixo de objeto TaxDocs para dizer ao Amazon S3 para replicar somente objetos com o prefixo de nome de chave TaxDocs. Todos os objetos dos quais você fizer upload e tiverem esse prefixo de nome de chave — por exemplo, TaxDocs/document1.pdf — serão replicados. Para qualquer solicitação de objeto com esse prefixo de nome de chave, o Amazon S3 retorna o cabeçalho x-amz-replication-status com um dos seguintes valores para o status de replicação de objeto: PENDING, COMPLETED ou FAILED.

### Note

Se a replicação do objeto falhar depois de você fazer upload de um objeto, não será possível tentar novamente a replicação. É preciso fazer upload do objeto novamente.

- Ao solicitar um objeto no bucket de destino, se o objeto da sua solicitação for uma réplica criada pelo Amazon S3, o Amazon S3 retornará o cabeçalho x-amz-replication-status com valor REPLICATED.

Descubra o status de replicação do objeto no console, com AWS Command Line Interface (AWS CLI) ou com o AWS SDK.

- Console: Escolha o objeto e, em seguida, Properties (Propriedades) para ver as propriedades de objeto, incluindo o status de replicação.
- AWS CLI: Use o comando da AWS CLI head-object para recuperar metadados do objeto:

```
aws s3api head-object --bucket source-bucket --key object-key --version-id object-version-id
```

O comando retorna os metadados do objeto, incluindo `ReplicationStatus`, conforme exibido na resposta de exemplo a seguir:

```
{  
    "AcceptRanges": "bytes",  
    "ContentType": "image/jpeg",  
    "LastModified": "Mon, 23 Mar 2015 21:02:29 GMT",  
    "ContentLength": 3191,  
    "ReplicationStatus": "COMPLETED",  
    "VersionId": "jfnW.HIMOFYiD_9rGbSkmroXsFj3fqZ.",  
    "ETag": "\"6805f2cfc46c0f04559748bb039d69ae\"",  
    "Metadata": {}  
}
```

- AWS SDKs: Os fragmentos de código a seguir obtêm status de replicação com AWS SDK for Java e AWS SDK para .NET, respectivamente.
- AWS SDK for Java

```
GetObjectMetadataRequest metadataRequest = new GetObjectMetadataRequest(bucketName,  
    key);  
ObjectMetadata metadata = s3Client.getObjectMetadata(metadataRequest);  
  
System.out.println("Replication Status : " +  
    metadata.getRawHeaderValue(Headers.OBJECT_REPLICATION_STATUS));
```

- AWS SDK para .NET

```
GetObjectMetadataRequest getmetadataRequest = new GetObjectMetadataRequest  
{  
    BucketName = sourceBucket,  
    Key        = objectKey  
};  
  
GetObjectMetadataResponse getmetadataResponse =  
    client.GetObjectMetadata(getmetadataRequest);  
Console.WriteLine("Object replication status: {0}",  
    getmetadataResponse.ReplicationStatus);
```

#### Note

Antes de excluir um objeto de um bucket de origem com a replicação habilitada, verifique o status de replicação dele para garantir que o objeto tenha sido replicado.  
Se a configuração de ciclo de vida estiver habilitada no bucket de origem, o Amazon S3 suspenderá as ações de ciclo de vida até que o status dos objetos seja `COMPLETED` ou `FAILED`.

## Tópicos relacionados

[Replicação entre regiões \(p. 544\)](#)

## Solucionar problemas da replicação entre regiões

Se as réplicas dos objetos não aparecerem no bucket de destino depois de configurar a replicação entre regiões, use as dicas a seguir para identificar e corrigir os problemas.

- A replicação de objeto costuma levar alguns minutos, mas algumas vezes pode demorar horas e, raramente, até um ou mais dias. O tempo que o Amazon S3 leva para replicar um objeto depende de vários fatores, como da origem e do destino do par de regiões e do tamanho do objeto. Para objetos grandes, a replicação pode levar várias horas. Se o objeto que estiver sendo replicado for grande, aguarde um pouco antes de conferir se ele está sendo exibido no bucket de destino. Você também pode conferir o status de replicação do objeto de origem. Se o status de replicação do objeto for pending, você saberá que o Amazon S3 não concluiu a replicação. Se o status de replicação do objeto for failed, confira a configuração de replicação definida no bucket de origem.
- Na configuração de replicação do bucket de origem, verifique o seguinte:
  - O nome de recurso da Amazon (ARN) do bucket de destino está correto.
  - O prefixo do nome de chave está correto. Por exemplo, se você definiu a configuração para replicar objetos com o prefixo Tax, apenas objetos com nomes de chaves como Tax/document1 ou Tax/document2 serão replicados. Um objeto com o nome de chave document3 não será replicado.
  - O status é enabled.
- Se o bucket de destino pertencer a outra conta da AWS, verifique se o proprietário do bucket tem uma política de bucket no bucket de destino que permite que o proprietário do bucket de origem replique objetos. Para ver um exemplo, consulte [Exemplo 2: Configurar CRR quando os buckets de origem e de destino forem de propriedade de contas da AWS diferentes \(p. 575\)](#).
- Se a réplica do objeto não aparecer no bucket de destino, a replicação poderá ter sido evitada pelo seguinte:
  - O Amazon S3 não replica um objeto em um bucket de origem que seja uma réplica criada por outra configuração de replicação. Por exemplo, se você definir a configuração de replicação do bucket A para o bucket B e para o bucket C, o Amazon S3 não replicará réplicas de objeto no bucket B para o bucket C.
  - O proprietário de bucket de origem pode conceder a outras contas da AWS permissão para fazer upload de objetos. Por padrão, o proprietário do bucket de origem não tem nenhuma permissão para os objetos criados por outras contas. A configuração de replicação vai replicar somente os objetos para os quais o proprietário do bucket de origem tem permissões de acesso. O proprietário do bucket de origem pode conceder a outras contas da AWS permissões para criar objetos condicionalmente exigindo permissões explícitas de acesso nesses objetos. Para ver um exemplo de política, consulte [Conceder permissões entre contas para fazer upload de objetos garantindo que o proprietário do bucket tenha controle total \(p. 364\)](#).
- Vamos supor que, na configuração da replicação, você adicione uma regra para replicar um subgrupo de objetos com uma tag específica. Neste caso, atribua a chave da tag específica e o valor no momento de criar o objeto para o Amazon S3 replicar o objeto. Se você primeiro criar um objeto e depois adicionar a tag ao objeto existente, o Amazon S3 não vai replicar o objeto.

## Tópicos relacionados

[Replicação entre regiões \(p. 544\)](#)

## Outras considerações sobre replicação entre regiões

O Amazon S3 também oferece suporte às configurações do bucket para o seguinte:

- Versionamento. Para obter mais informações, consulte [Usar versionamento \(p. 448\)](#).
- Hospedagem de sites. Para obter mais informações, consulte [Hospedagem de um site estático no Amazon S3 \(p. 494\)](#).

- Acesso ao bucket por meio de uma política ou lista de controle de acesso (ACL). Para obter mais informações, consulte [Uso de políticas de bucket e políticas de usuário \(p. 326\)](#) e [Gerenciar o acesso com ACLs \(p. 390\)](#).
- Armazenamento de logs. Para obter mais informações, [Registro em log de acesso ao servidor Amazon S3 \(p. 625\)](#).
- Gerenciamento do ciclo de vida para os objetos dentro de um bucket. Para obter mais informações, consulte [Gerenciamento do ciclo de vida de objetos \(p. 122\)](#).

Este tópico explica como a configuração de replicação do bucket afeta o comportamento dessas configurações do bucket.

## Configuração de ciclo de vida e réplicas de objeto

O tempo que o Amazon S3 leva para replicar um objeto depende do tamanho do objeto. Para objetos grandes, pode levar várias horas. Embora possa demorar um pouco até a réplica ser disponibilizada no bucket de destino, demora o mesmo tempo para criar a réplica que demorou para criar o objeto correspondente no bucket de origem. Se uma política de ciclo de vida estiver habilitada no bucket de destino, as regras de ciclo de vida honram o tempo original de criação de objeto, não o momento em que a réplica foi disponibilizada no bucket de destino.

Se você tiver uma política de ciclo de vida de expiração do objeto em um bucket sem versão e quiser manter o mesmo comportamento de exclusão permanente quando habilitar o versionamento, precisará adicionar uma política de expiração de versão desatualizada para gerenciar as exclusões das versões de objetos desatualizadas no bucket habilitado por versão.

A configuração de replicação requer que o bucket seja ativado por versionamento. Ao habilitar o versionamento em um bucket, lembre-se de:

- Se você tiver uma política de ciclo de vida de expiração de um objeto, depois de habilitar o versionamento, adicione uma política de `NonCurrentVersionExpiration` para manter o mesmo comportamento de exclusão permanente que antes de habilitar o versionamento.
- Se você tiver uma política de ciclo de vida de transição, depois de habilitar o versionamento, considere adicionar a política `NonCurrentVersionTransition`.

## Configuração do versionamento e configuração de replicação

Os buckets de origem e de destino devem ter versionamento habilitado quando você configura replicação em um bucket. Depois que você habilitar o versionamento nos buckets de origem e de destino e configurar a replicação no bucket de origem, vai encontrar os seguintes problemas:

- Se você tentar desabilitar o versionamento do bucket de origem, o Amazon S3 retornará um erro. É necessário remover a configuração de replicação antes de desabilitar o versionamento o bucket de origem.
- Se você desabilitar o versionamento o bucket de destino, ocorrerá falha na replicação. O objeto de origem tem o status de replicação `Failed`.

## Configuração de log e de replicação

Se o Amazon S3 entregar logs em um bucket com a replicação habilitada, ele vai replicar os objetos do log.

Se os logs de acesso ao servidor ([Registro em log de acesso ao servidor Amazon S3 \(p. 625\)](#)) ou logs do AWS CloudTrail ([Registro em log de chamadas à API do Amazon S3 usando o AWS CloudTrail \(p. 606\)](#)) estiverem ativados no seu bucket de origem ou de destino, o Amazon S3 incluirá nos logs as solicitações relacionadas à CRR. Por exemplo, o Amazon S3 registra cada objeto que ele replica.

## CRR e região de destino

Na configuração de CRR, os buckets de origem e destino devem estar em regiões da AWS diferentes. Você pode escolher a região do seu bucket de destino com base nas suas necessidades comerciais ou nas considerações de custo. Por exemplo, as cobranças de transferência de dados entre regiões variam dependendo das regiões que você escolher. Vamos supor que você escolhe Leste dos EUA (Norte da Virgínia) (us-east-1) como a região para o bucket de origem. Se você escolher Oeste dos EUA (Oregon) (us-west-2) como a região do bucket de destino, pagará mais do que se escolher Leste dos EUA (Ohio) (us-east-2). Para obter informações sobre preços, consulte a seção "Definição de preço da transferência de dados" em [Definição de preço do Amazon S3](#).

## Pausar a configuração de replicação

Para pausar temporariamente a replicação, desabilite a regra em questão na configuração da replicação.

Se a replicação estiver habilitada e você remover a função do IAM que concede ao Amazon S3 as permissões necessárias, a replicação falhará. O Amazon S3 vai reportar o status da replicação para os objetos afetados como `Failed`.

## Tópicos relacionados

[Replicação entre regiões \(p. 544\)](#)

# Roteamento de solicitação

## Tópicos

- [Redirecionamento de solicitação e a API REST \(p. 590\)](#)
- [Considerações de DNS \(p. 594\)](#)

Os programas que fazem solicitações em buckets criados usando a API <CreateBucketConfiguration> devem oferecer suporte a redirecionamentos. Além disso, alguns clientes que não respeitam TTLs DNS podem encontrar problemas.

Esta seção descreve problemas de roteamento e DNS a serem considerados ao projetar seu serviço ou aplicativo para uso com o Amazon S3.

## Redirecionamento de solicitação e a API REST

O Amazon S3 usa o Domain Name System (DNS) para rotear solicitações para instalações capazes de processá-las. Esse sistema funciona com eficiência, mas podem ocorrer erros de roteamento temporários. Se uma solicitação chega na localização errada do Amazon S3, o Amazon S3 responde com um redirecionamento temporário pedindo que o solicitante reenvie a solicitação para um novo endpoint. Se uma solicitação é formada de maneira incorreta, o Amazon S3 usa redirecionamentos constantes para fornecer direções sobre como executar a solicitação corretamente.

### Important

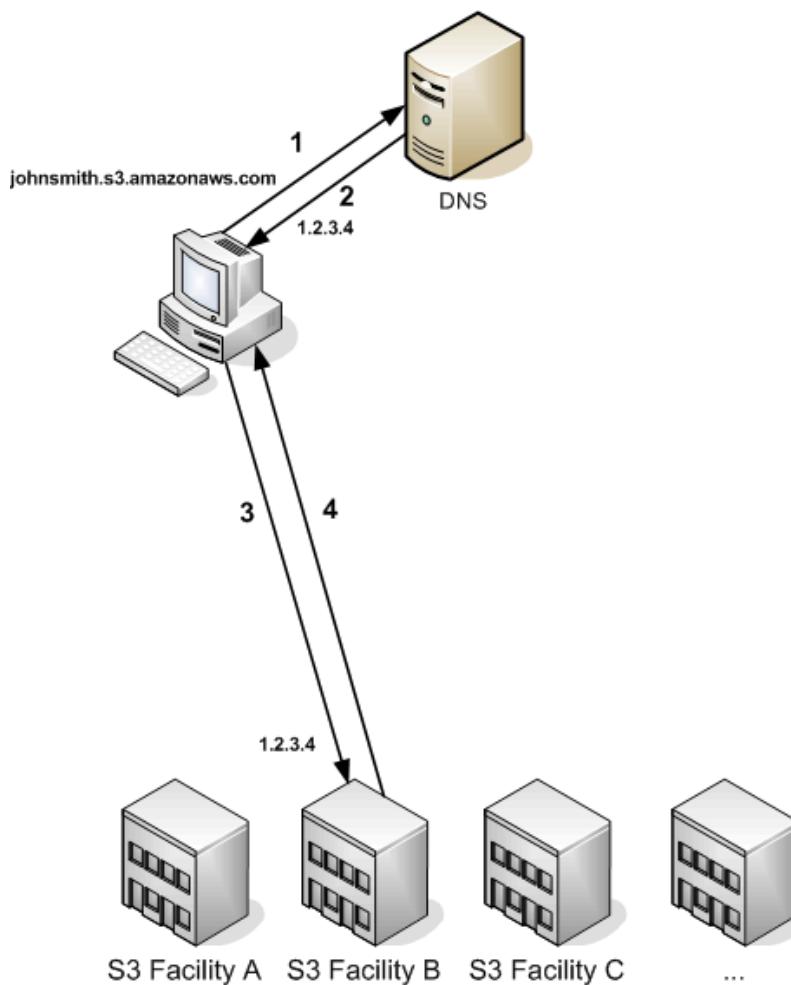
Para usar esse recurso, você deve ter um aplicativo que possa lidar com respostas de redirecionamento do Amazon S3. A única exceção é para aplicativos que funcionam exclusivamente com buckets criados sem <CreateBucketConfiguration>. Para obter mais informações sobre restrições de localização, consulte [Acesso a um bucket \(p. 56\)](#).

## Tópicos

- [Roteamento de DNS \(p. 590\)](#)
- [Redirecionamento de solicitação temporário \(p. 591\)](#)
- [Redirecionamento permanente de solicitação \(p. 593\)](#)
- [Exemplos de redirecionamento de solicitação \(p. 593\)](#)

## Roteamento de DNS

O roteamento de DNS encaminha solicitações para instalações apropriadas do Amazon S3. A figura e o procedimento a seguir mostram um exemplo de roteamento de DNS.



#### Etapas de solicitação de roteamento DNS

1. O cliente faz uma solicitação de DNS para obter um objeto armazenado no Amazon S3.
2. O cliente recebe um ou mais endereços IP para instalações capazes de processar a solicitação. Neste exemplo, o endereço IP é para a instalação B.
3. O cliente faz uma solicitação à instalação B do Amazon S3.
4. A instalação B retorna uma cópia do objeto ao cliente.

## Redirecionamento de solicitação temporário

Um redirecionamento temporário é um tipo de resposta de erro que indica que o solicitante deve reenviar a solicitação para um endpoint diferente. Devido à natureza distribuída do Amazon S3, as solicitações podem ser roteadas temporariamente para a instalação errada. É mais provável que isso aconteça imediatamente após a criação ou exclusão de buckets.

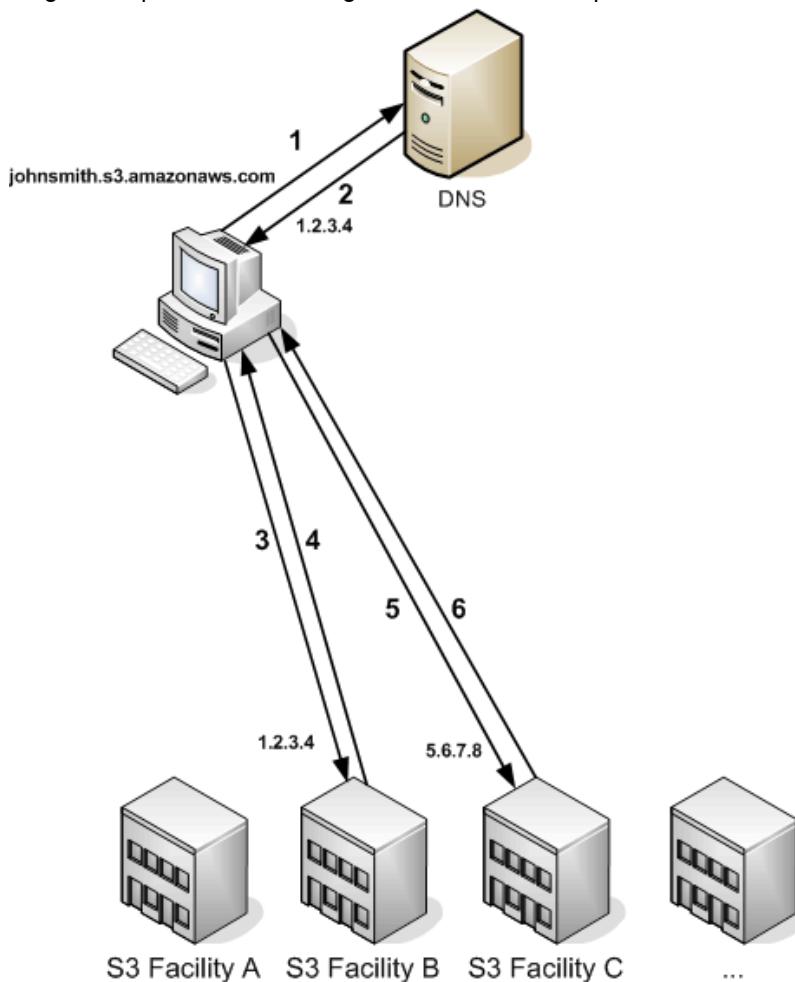
Por exemplo, se você criar um bucket novo e, em seguida, fizer uma solicitação a esse bucket, é possível que receba um redirecionamento temporário dependendo da restrição de localização do bucket. Se você criou o bucket na região da AWS Leste dos EUA (Norte da Virgínia), você não verá o redirecionamento já que esse também é o endpoint padrão do Amazon S3.

No entanto, se o bucket for criado em qualquer outra região, todas as solicitações feitas ao bucket vão para o endpoint padrão enquanto a entrada do DNS do bucket é propagada. O endpoint padrão redireciona a solicitação para o endpoint correto com uma resposta HTTP 302. Os redirecionamentos temporários contêm um URI para a instalação correta, que pode ser usado para reenviar a solicitação imediatamente.

**Important**

Não reutilize um endpoint fornecido por uma resposta de redirecionamento anterior. Ele pode parecer funcionar (até mesmo durante longos períodos), mas pode fornecer resultados imprevisíveis e eventualmente falhará sem aviso.

A figura e o procedimento a seguir mostram um exemplo de um redirecionamento temporário.



**Etapas de redirecionamento de solicitação temporário**

1. O cliente faz uma solicitação de DNS para obter um objeto armazenado no Amazon S3.
2. O cliente recebe um ou mais endereços IP para instalações capazes de processar a solicitação.
3. O cliente faz uma solicitação à instalação B do Amazon S3.
4. A instalação B retorna um redirecionamento indicando que o objeto está disponível na localização C.
5. O cliente reenvia a solicitação para a instalação C.
6. A instalação C retorna uma cópia do objeto.

## Redirecionamento permanente de solicitação

Um redirecionamento permanente indica que a solicitação abordou um recurso de maneira inapropriada. Por exemplo, redirecionamentos permanentes ocorrem se você usar uma solicitação no estilo de caminho para acessar um bucket criado com <CreateBucketConfiguration>. Para obter mais informações, consulte [Acesso a um bucket \(p. 56\)](#).

Para ajudar a encontrar esses erros durante o desenvolvimento, esse tipo de redirecionamento não contém um cabeçalho HTTP de localização que permite o acompanhamento automático da solicitação para a localização correta. Consulte o documento de erros XML resultante para obter ajuda no uso do endpoint correto do Amazon S3.

## Exemplos de redirecionamento de solicitação

Veja a seguir exemplos de respostas de redirecionamento de solicitação temporário.

### API REST de resposta de redirecionamento temporário

```
HTTP/1.1 307 Temporary Redirect
Location: http://johnsmith.s3-gztb4pa9sq.amazonaws.com/photos/puppy.jpg?rk=e2c69a31
Content-Type: application/xml
Transfer-Encoding: chunked
Date: Fri, 12 Oct 2007 01:12:56 GMT
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>TemporaryRedirect</Code>
  <Message>Please re-send this request to the specified temporary endpoint.
  Continue to use the original request endpoint for future requests.</Message>
  <Endpoint>johnsmith.s3-gztb4pa9sq.amazonaws.com</Endpoint>
</Error>
```

### API SOAP de resposta de redirecionamento temporário

#### Note

O suporte de SOAP via HTTP está obsoleto, mas continua disponível via HTTPS. Novos recursos do Amazon S3 não serão compatíveis com o SOAP. Recomendamos que você use o REST API ou os SDKs da AWS.

```
<soapenv:Body>
<soapenv:Fault>
  <Faultcode>soapenv:Client.TemporaryRedirect</Faultcode>
  <Faultstring>Please re-send this request to the specified temporary endpoint.
  Continue to use the original request endpoint for future requests.</Faultstring>
  <Detail>
    <Bucket>images</Bucket>
    <Endpoint>s3-gztb4pa9sq.amazonaws.com</Endpoint>
  </Detail>
</soapenv:Fault>
</soapenv:Body>
```

## Considerações de DNS

Um dos requisitos de design do Amazon S3 é uma disponibilidade extremamente alta. Uma das maneiras em que cumprimos esse requisito é atualizando os endereços IP associados com o endpoint do Amazon S3 no DNS, conforme necessário. Essas alterações são refletidas automaticamente em clientes com tempo de vida curto, mas não em alguns clientes com tempo de vida longo. Os clientes com tempo de vida longo precisarão realizar uma ação especial para resolver de novo o endpoint do Amazon S3 periodicamente para se beneficiar com essas alterações. Para obter mais informações sobre Virtual Machines (VMs – Máquinas virtuais), consulte o seguinte:

- Para Java, por padrão, o JVM do Sun armazena em cache as pesquisas de DNS para sempre; acesse a seção "InetAddress Caching" da [documentação do InetAddress](#) para obter informações sobre como alterar esse comportamento.
- Para PHP, a VM PHP persistente que é executada nas configurações de implantação mais populares armazena em cache as pesquisas do DNS até a VM ser reiniciada. Acesse [os docs getHostByName PHP](#).

# Otimização do desempenho

Esta seção aborda as práticas recomendadas do Amazon S3 para otimizar o desempenho nos tópicos a seguir.

## Tópicos

- [Orientações sobre desempenho e taxa de solicitações \(p. 595\)](#)
- [Escalabilidade da janela de TCP \(p. 595\)](#)
- [Reconhecimento seletivo de TCP \(p. 596\)](#)

## Note

Para obter mais informações sobre ajuste de alto desempenho, consulte [Enabling High Performance Data Transfers](#) no site do Pittsburgh Supercomputing Center (PSC).

## Orientações sobre desempenho e taxa de solicitações

O Amazon S3 escala automaticamente para taxas de solicitações elevadas. Por exemplo, seu aplicativo pode alcançar pelo menos 3.500 solicitações PUT/POST/DELETE e 5.500 solicitações GET por segundo por prefixo em um bucket. Não há limite para o número de prefixos em um bucket. É simples aumentar exponencialmente o desempenho de leitura ou gravação. Por exemplo, se você cria 10 prefixos em um bucket do Amazon S3 para paralelizar leituras, é possível escalar o desempenho de leitura para 55.000 solicitações de leitura por segundo.

Se a carga de trabalho do Amazon S3 usa criptografia no lado do servidor com o AWS Key Management Service (SSE-KMS), consulte [Limites do AWS KMS](#) no AWS Key Management Service Developer Guide para obter informações sobre as taxas de solicitações compatíveis com seu caso de uso.

## Cargas de trabalho que usam muito GET

Se sua carga de trabalho estiver enviando principalmente solicitações GET, além das diretrizes anteriores, você deve considerar usar o Amazon CloudFront para otimização de desempenho. Ao integrar o CloudFront ao Amazon S3, é possível distribuir conteúdo aos usuários com baixa latência e uma alta taxa de transferência de dados. Você também envia menos solicitações diretas para o Amazon S3, o que reduz os custos.

Por exemplo, suponha que você tenha alguns objetos que são muito populares. O CloudFront busca esses objetos do Amazon S3 e os armazena em cache. Em seguida, o CloudFront pode atender a futuras solicitações para os objetos de seu cache, reduzindo o número de solicitações GET enviadas para o Amazon S3. Para obter mais informações, consulte a página de detalhes de produtos do [Amazon CloudFront](#).

## Escalabilidade da janela de TCP

A escalabilidade da janela de TCP permite que você melhore o desempenho de throughput de rede entre o sistema operacional e a camada de aplicativo e o Amazon S3 com suporte para janelas com mais

de 64 KB. No início da sessão TCP, um cliente apresenta o fator WSCALE da janela de recebimento suportado, e o Amazon S3 responde com o fator WSCALE da janela de recebimento suportado para o sentido ascendente.

Embora a escalabilidade da janela de TCP possa melhorar o desempenho, pode ser difícil definir adequadamente. Ajuste as configurações no nível de aplicativo e de kernel. Para obter mais informações sobre a escalabilidade da janela de TCP, consulte a documentação do sistema operacional e acesse [RFC 1323](#).

## Reconhecimento seletivo de TCP

O reconhecimento seletivo de TCP foi criado para aumentar o tempo de recuperação após um grande número de perdas de pacotes. Ele é compatível com a maioria dos sistemas operacionais mais novos, mas talvez precise ser habilitado. Para obter mais informações sobre reconhecimentos seletivos de TCP, consulte a documentação fornecida com o sistema operacional e acesse [RFC 2018](#).

# Monitoramento do Amazon S3

O monitoramento é uma parte importante da manutenção da confiabilidade, da disponibilidade e da performance do Amazon S3 e das suas soluções da AWS. Você deve coletar dados de monitoramento de todas as partes de sua solução da AWS para ser mais fácil realizar a depuração de uma falha de vários pontos (caso ocorra). Porém, para começar a monitorar o Amazon S3, é necessário criar um plano de monitoramento que inclua respostas às seguintes perguntas:

- Quais são seus objetivos de monitoramento?
- Quais recursos você vai monitorar?
- Com que frequência você vai monitorar esses recursos?
- Quais ferramentas de monitoramento você usará?
- Quem realizará o monitoramento das tarefas?
- Quem deve ser notificado quando algo der errado?

## Tópicos

- [Ferramentas de monitoramento \(p. 597\)](#)
- [Métricas de monitoramento com o Amazon CloudWatch \(p. 598\)](#)
- [Configurações de métricas para buckets \(p. 604\)](#)
- [Registro em log de chamadas à API do Amazon S3 usando o AWS CloudTrail \(p. 606\)](#)

## Ferramentas de monitoramento

A AWS fornece várias ferramentas que você pode usar para monitorar o Amazon S3. É possível configurar algumas dessas ferramentas para fazer o monitoramento em seu lugar, e, ao mesmo tempo, algumas das ferramentas exigem intervenção manual. Recomendamos que as tarefas de monitoramento sejam automatizadas ao máximo possível.

## Ferramentas de monitoramento automatizadas

Use as seguintes ferramentas de monitoramento automatizado para observar o Amazon S3 e gerar relatórios quando algo estiver errado:

- Alarmes do Amazon CloudWatch – observe uma única métrica ao longo de um período que você especificar e realize uma ou mais ações com base no valor da métrica em relação a um determinado limite ao longo de vários períodos. A ação é uma notificação enviada a um tópico do Amazon Simple Notification Service (Amazon SNS) ou a uma política do Amazon EC2 Auto Scaling. Os alarmes do CloudWatch não invocam ações simplesmente porque estão em um estado específico. O estado deve ter sido alterado e mantido por um número específico de períodos. Para obter mais informações, consulte [Métricas de monitoramento com o Amazon CloudWatch \(p. 598\)](#).
- AWS CloudTrail Log Monitoring (Monitoramento do log de CTI) – Compartilhe arquivos de log entre contas, monitore os arquivos de log do CloudTrail em tempo real enviando-os para o CloudWatch Logs, grave aplicativos de processamento de logs em Java e confirme se os arquivos de log não foram

alterados após a entrega pelo CloudTrail. Para obter mais informações, consulte [Registro em log de chamadas à API do Amazon S3 usando o AWS CloudTrail \(p. 606\)](#).

## Ferramentas de monitoramento manual

Outra parte importante do monitoramento do Amazon S3 envolve o monitoramento manual desses itens que os alarmes do CloudWatch não abrangem. Os painéis dos consoles da AWS Amazon S3, CloudWatch, Trusted Advisor e outros fornecem uma visão rápida do estado do ambiente da AWS. Você pode permitir o registro de acessos ao servidor que acompanha as solicitações de acesso ao seu bucket. Cada registro de log de acesso fornece detalhes sobre uma única solicitação de acesso, como solicitante, nome do bucket, horário da solicitação, ação da solicitação, status de resposta e código de erro, se houver. Para obter mais informações, consulte [Registro em log de acesso ao servidor Amazon S3 \(p. 625\)](#) no Guia do desenvolvedor do Amazon Simple Storage Service.

- O painel da Amazon S3 mostra:
  - Seus buckets, objetos e propriedades que contêm.
- A página inicial do CloudWatch mostra:
  - Alarmes e status atual.
  - Gráficos de alarmes e recursos.
  - Estado de integridade do serviço.

Além disso, você pode usar o CloudWatch para fazer o seguinte:

- Criar [painéis personalizados](#) para monitorar os serviços com os quais você se preocupa.
- Colocar em gráfico dados de métrica para solucionar problemas e descobrir tendências.
- Pesquisar e procurar todas as métricas de recursos da AWS.
- Criar e editar alarmes para ser notificado sobre problemas.
- O AWS Trusted Advisor pode ajudar a monitorar os recursos da AWS para melhorar o desempenho, a confiabilidade, a segurança e a economia. Quatro verificações do Trusted Advisor estão disponíveis a todos os usuários; mais de 50 verificações estão disponíveis para usuários com um plano de suporte Business ou Enterprise. Para obter mais informações, consulte [AWS Trusted Advisor](#).

O Trusted Advisor tem essas verificações que se referem ao Amazon S3:

- Verificações de configuração de log dos buckets do Amazon S3.
- Verificações de segurança para buckets do Amazon S3 que têm permissões de acesso livre.
- Verificações de tolerância a falhas para os buckets do Amazon S3 que não têm versionamento ativado ou têm versionamento suspenso.

## Métricas de monitoramento com o Amazon CloudWatch

As métricas do Amazon CloudWatch para o Amazon S3 podem ajudá-lo a entender e melhorar o desempenho de aplicativos que usam o Amazon S3. Há duas maneiras de se usar o CloudWatch com o Amazon S3.

- Métricas diárias de armazenamento para buckets - Você pode monitorar o armazenamento de bucket com o CloudWatch, que coleta e processa dados de armazenamento do Amazon S3 em métricas diárias legíveis. Essas métricas de armazenamento para o Amazon S3 são relatadas uma vez por dia e fornecidas a todos os clientes sem qualquer custo adicional.
- Métricas de solicitações - Você pode escolher monitorar as solicitações do Amazon S3 para identificar e atuar rapidamente em problemas operacionais. As métricas estão disponíveis em intervalos de 1 minuto

após alguma latência para processar. Essas métricas do CloudWatch são faturadas na mesma taxa que as métricas personalizadas do Amazon CloudWatch. Para obter informações sobre definição de preços do CloudWatch, consulte [Definição de preços do Amazon CloudWatch](#). Para saber mais sobre como optar por obter essas métricas, consulte [Configurações de métricas para buckets \(p. 604\)](#).

Quando ativadas, as métricas de solicitações são relatadas para todas as operações de objeto. Por padrão, essas métricas de 1 minuto estão disponíveis no nível do bucket do Amazon S3. Você também pode definir um filtro para as métricas coletadas – usando um prefixo compartilhado ou tag de objeto – permitindo que você alinhe filtros de métricas para aplicativos de negócios, fluxos de trabalho ou organizações internas específicos.

Todas as estatísticas do CloudWatch ficam retidas por um período de 15 meses, para que você possa acessar informações históricas e obter uma perspectiva melhor sobre como o serviço ou o aplicativo web estão se saindo. Para obter mais informações sobre o CloudWatch consulte [O que são Amazon CloudWatch, Eventos do Amazon CloudWatch e Amazon CloudWatch Logs?](#) no Guia do usuário do Amazon CloudWatch.

## Métricas e dimensões

As métricas e dimensões de armazenamento que o Amazon S3 envia ao CloudWatch estão listadas abaixo.

### Métricas diárias de armazenamento por buckets do CloudWatch do Amazon S3

O namespace AWS/S3 inclui as métricas de armazenamento diário a seguir para buckets.

Métrica	Descrição
BucketSizeBytes	A quantidade de dados em bytes armazenados em um bucket nas classes de armazenamento STANDARD, INTELLIGENT_TIERING, Standard - Infrequent Access (STANDARD_IA), OneZone - Infrequent Access (ONEZONE_IA), Reduced Redundancy Storage (RRS) ou Glacier (GLACIER).  Filtros de tipo de armazenamento válidos: StandardStorage, IntelligentTieringStorage, GlacierS3ObjectOverhead, StandardIAStorage, StandardIAObjectOverhead, StandardIAStorage, StandardIAObjectOverhead, OneZoneIAStorage, OneZoneIAObjectOverhead, ReducedRedundancyStorage, GlacierStorage e GlacierObjectOverhead (consulte a dimensão StorageType)  Unidade: bytes  Estatística válida: média
NumberOfObjects	O número total de objetos (inclui todos os objetos e todas as suas versões) armazenados em um bucket para todas as classes de armazenamento.  Filtros de tipo de armazenamento válidos: AllStorageTypes (consulte a dimensão StorageType)  Unidade: contagem  Estatística válida: média

## Métricas de solicitação do CloudWatch do Amazon S3

O namespace AWS/S3 inclui as métricas de solicitação a seguir.

Métrica	Descrição
AllRequests	<p>O número total de solicitações HTTP feitas em um bucket do Amazon S3, independentemente do tipo. Se você estiver usando uma configuração de métricas com um filtro, então essa métrica só retornará as solicitações HTTP feitas para os objetos no bucket que atendam aos requisitos do filtro.</p> <p>Unidade: contagem</p> <p>Estatística válida: soma</p>
GetRequests	<p>O número de solicitações HTTP GET feitas para objetos em um bucket do Amazon S3. Isso não inclui operações de lista.</p> <p>Unidade: contagem</p> <p>Estatística válida: soma</p> <p><b>Note</b></p> <p>Solicitações paginadas orientadas a listas, como <a href="#">Listar multipart uploads</a>, <a href="#">Listar partes</a>, <a href="#">Obter versões de objetos do bucket</a> e outras não estão incluídas nessa métrica.</p>
PutRequests	<p>O número de solicitações HTTP PUT feitas para objetos em um bucket do Amazon S3.</p> <p>Unidade: contagem</p> <p>Estatística válida: soma</p>
DeleteRequests	<p>O número de solicitações HTTP DELETE feitas para objetos em um bucket do Amazon S3. Isso também inclui solicitações <a href="#">Excluir vários objetos</a>. Essa métrica exibe o número de solicitações, não o número de objetos excluídos.</p> <p>Unidade: contagem</p> <p>Estatística válida: soma</p>
HeadRequests	<p>O número de solicitações HTTP HEAD feitas para um bucket do Amazon S3.</p> <p>Unidade: contagem</p> <p>Estatística válida: soma</p>
PostRequests	<p>O número de solicitações HTTP POST feitas para um bucket do Amazon S3.</p> <p>Unidade: contagem</p> <p>Estatística válida: soma</p> <p><b>Note</b></p> <p>As solicitações <a href="#">Excluir vários objetos</a> e <a href="#">Conteúdo de objetos do SELECT</a> não estão incluídas nessa métrica.</p>

Métrica	Descrição
SelectRequests	O número de solicitações <a href="#">Conteúdo de objetos do SELECT</a> do Amazon S3 feitas para os objetos em um bucket do Amazon S3.  Unidade: contagem  Estatística válida: soma
SelectScannedBytes	Número de bytes de dados verificados com solicitações <a href="#">Conteúdo de objetos do SELECT</a> do Amazon S3 feitas em um bucket do Amazon S3.  Unidade: bytes  Estatísticas válidas: média (bytes por solicitação), soma (bytes por período), contagem de amostras, mín., máx.
SelectReturnedBytes	O número de bytes de dados retornados com solicitações <a href="#">Conteúdo de objetos do SELECT</a> do Amazon S3 feitas em um bucket do Amazon S3.  Unidade: bytes  Estatísticas válidas: média (bytes por solicitação), soma (bytes por período), contagem de amostras, mín., máx.
ListRequests	O número de solicitações HTTP que listam o conteúdo de um bucket.  Unidade: contagem  Estatística válida: soma
BytesDownloaded	O número de bytes baixados para solicitações feitas para um bucket do Amazon S3, em que a resposta inclui um corpo.  Unidade: bytes  Estatísticas válidas: média (bytes por solicitação), soma (bytes por período), contagem de amostras, mín., máx.
BytesUploaded	O número de bytes que contêm um corpo de solicitação, carregados para um bucket do Amazon S3.  Unidade: bytes  Estatísticas válidas: média (bytes por solicitação), soma (bytes por período), contagem de amostras, mín., máx.
4xxErrors	O número de solicitações de código de status de erro do cliente HTTP 4xx feitas para um bucket do Amazon S3 com um valor de 0 ou 1. A estatística <code>average</code> mostra a taxa de erros, e a estatística <code>sum</code> mostra a contagem desse tipo de erro, durante cada período.  Unidade: contagem  Estatísticas válidas: média (relatórios por solicitação), soma (relatórios por período), mín., máx., contagem de amostras

Métrica	Descrição
<b>5xxErrors</b>	<p>O número de solicitações de código de status de erro do servidor HTTP 5xx feitas para um bucket do Amazon S3 com um valor de 0 ou 1. A estatística <code>average</code> mostra a taxa de erros, e a estatística <code>sum</code> mostra a contagem desse tipo de erro, durante cada período.</p> <p>Unidade: contagens</p> <p>Estatísticas válidas: média (relatórios por solicitação), soma (relatórios por período), mín., máx., contagem de amostras</p>
<b>FirstByteLatency</b>	<p>O tempo por solicitação desde o recebimento da solicitação completa por um bucket do Amazon S3 até o momento em que a resposta começa a ser retornada.</p> <p>Unidade: milissegundos</p> <p>Estatísticas válidas: média, soma, mín., máx., contagem de amostras</p>
<b>TotalRequestLatency</b>	<p>O tempo por solicitação decorrido do primeiro byte recebido até o último byte enviado para um bucket do Amazon S3. Isso inclui o tempo necessário para receber o corpo da solicitação e enviar o corpo da resposta, que não está incluído em <code>FirstByteLatency</code>.</p> <p>Unidade: milissegundos</p> <p>Estatísticas válidas: média, soma, mín., máx., contagem de amostras</p>

## Amazon S3 CloudWatch Dimensões

As seguintes dimensões são usadas para filtrar as métricas do Amazon S3.

Dimensão	Descrição
<b>BucketName</b>	Essa dimensão filtra os dados que você solicita somente para o bucket identificado.
<b>StorageType</b>	Essa dimensão filtra os dados que você armazenou em um bucket pelo tipo de armazenamento. Os tipos são <code>StandardStorage</code> para a classe de armazenamento STANDARD, <code>IntelligentTieringStorage</code> para a classe de armazenamento INTELLIGENT_TIERING, <code>StandardIAStorage</code> para a classe de armazenamento STANDARD_IA, <code>OneZoneIAStorage</code> para a classe de armazenamento ONEZONE_IA, <code>ReducedRedundancyStorage</code> para a classe de armazenamento REDUCED_REDUNDANCY, <code>GlacierStorage</code> para a classe de armazenamento GLACIER e <code>AllStorageTypes</code> . O tipo <code>AllStorageTypes</code> inclui as classes de armazenamento STANDARD, INTELLIGENT_TIERING, STANDARD_IA, ONEZONE_IA, REDUCED_REDUNDANCY e GLACIER.
<b>FilterId</b>	Esta dimensão filtra as configurações de métricas que você especifica para métricas de solicitação em um bucket, por exemplo, um prefixo ou uma tag. Você especifica um ID de filtro ao criar uma

Dimensão	Descrição
	configuração de métricas. Para obter mais informações, consulte <a href="#">Configurações de métricas para buckets</a> .

## Acesso às métricas do CloudWatch

Você pode usar os procedimentos a seguir para visualizar as métricas de armazenamento para o Amazon S3. Observe que, para obter as métricas do Amazon S3 envolvidas, é necessário definir um time stamp de início e de término. Para métricas para qualquer período de 24 horas, configure o período para 86400 segundos, o número de segundos em um dia. Além disso, lembre de configurar as dimensões BucketName e StorageType.

Por exemplo, se você usa a AWS CLI para obter a média do tamanho de um bucket específico, em bytes, você pode usar o seguinte comando:

```
aws cloudwatch get-metric-statistics --metric-name BucketSizeBytes --namespace AWS/S3
--start-time 2016-10-19T00:00:00Z --end-time 2016-10-20T00:00:00Z --statistics Average
--unit Bytes --region us-west-2 --dimensions Name=BucketName,Value=ExampleBucket
Name=StorageType,Value=StandardStorage --period 86400 --output json
```

Este exemplo gera a seguinte saída:

```
{
  "Datapoints": [
    {
      "Timestamp": "2016-10-19T00:00:00Z",
      "Average": 1025328.0,
      "Unit": "Bytes"
    }
  ],
  "Label": "BucketSizeBytes"
}
```

Para visualizar as métricas usando o console do CloudWatch

1. Abra o console do CloudWatch em <https://console.aws.amazon.com/cloudwatch/>.
2. No painel de navegação, selecione Metrics (Métricas).
3. Selecione o namespace S3.
4. (Opcional) Para visualizar uma métrica, digite o nome da métrica no campo de pesquisa.
5. (Opcional) Para filtrar pela dimensão StorageType, digite o nome de classe de armazenamento no campo de pesquisa.

Ver uma lista de métricas válidas armazenadas para sua conta da AWS usando a CLI da AWS

- Em um prompt de comando, use o seguinte comando:

```
aws cloudwatch list-metrics --namespace "AWS/S3"
```

## Recursos relacionados

- [Amazon CloudWatch Logs API Reference](#)
- [Guia do usuário do Amazon CloudWatch](#)
- ação `list-metrics` no AWS CLI Command Reference.
- ação `get-metric-statistics` no AWS CLI Command Reference.
- [Configurações de métricas para buckets \(p. 604\)](#).

## Configurações de métricas para buckets

As métricas de solicitações do CloudWatch para o Amazon S3 permitem que você receba métricas do CloudWatch de 1 minuto, defina alarmes do CloudWatch e acesse painéis do CloudWatch para ver operações em tempo quase real e desempenho de seu armazenamento do Amazon S3. Para aplicativos que dependem de armazenamento em nuvem, essas métricas permitem que você identifique rapidamente os problemas operacionais e aja em relação a eles. Quando ativadas, essas métricas de 1 minuto estão disponíveis por padrão no nível do bucket do Amazon S3.

Você deve criar uma configuração de métricas para um bucket se desejar obter as métricas de solicitações do CloudWatch para objetos nesse bucket. Você também pode definir um filtro para as métricas coletadas – usando um prefixo compartilhado ou tags de objeto – permitindo que você alinhe filtros de métricas para aplicativos de negócios, fluxos de trabalho ou organizações internas específicos.

Para obter mais informações sobre as métricas do CloudWatch que estão disponíveis e as diferenças entre métricas de armazenamento e métricas de solicitação, consulte [Métricas de monitoramento com o Amazon CloudWatch \(p. 598\)](#).

Tenha o seguinte em mente ao usar as configurações de métricas:

- Você pode ter um máximo de 1000 configurações de métricas por bucket.
- Você pode escolher que objetos em um bucket serão incluídos nas configurações de métricas, utilizando filtros. Filtrar com um prefixo compartilhado ou tag de objeto permite que você alinhe filtros de métricas para aplicativos de negócios, fluxos de trabalho ou organizações internas específicos. Para métricas de solicitação para todo o bucket, crie uma configuração de métricas sem um filtro.
- As configurações de métricas são necessárias apenas para permitir métricas de solicitação. As métricas diárias de armazenamento ao nível do bucket- estão sempre ativadas e são fornecidas sem nenhum custo adicional. Atualmente, não é possível obter métricas diárias de armazenamento para um subconjunto filtrado de objetos.
- Cada configuração de métricas permite o conjunto completo de [métricas de solicitações disponíveis \(p. 600\)](#). As métricas específicas de operações (como `PostRequests`) serão relatadas somente se houver solicitações daquele tipo para seu bucket ou filtro.
- As métricas de solicitações são relatadas para operações ao nível do objeto e também são relatadas para operações que listem o conteúdo do bucket, como [GET Bucket \(Listar objetos\)](#), [GET versões de objetos do bucket](#) e [Listar multipart uploads](#), mas não são relatadas para outras operações em buckets.

## Entrega com melhor esforço de métricas do CloudWatch

As métricas do CloudWatch são entregues com base em melhor esforço. A maioria de solicitações para um objeto do Amazon S3 que tenha métricas de solicitações resulta no envio de um ponto de dados ao CloudWatch.

A integralidade e a oportunidade das métricas, no entanto, não são garantidas. O ponto de dados para uma solicitação específica pode ser retornado com um time stamp posterior à solicitação processada. Ou o ponto de dados para um minuto pode sofrer um atraso antes de ficar disponível por meio do CloudWatch, ou pode não ser entregue. As métricas de solicitação do CloudWatch fornecem uma ideia da natureza do tráfego em relação ao seu bucket quase em tempo real. Não se trata de uma contabilidade completa de todas as solicitações.

Devido à natureza de melhor esforço deste recurso, os relatórios disponíveis no [Painel de faturamento e de gerenciamento de custo](#) podem incluir uma ou mais solicitações de acesso que não aparecem nas métricas do bucket.

## Filtrar configurações de métricas

Ao trabalhar com configurações de métricas do CloudWatch, você tem a opção de filtrar a configuração em grupos de objetos relacionados em um único bucket. Você pode filtrar objetos em um bucket para inclusão em uma configuração de métricas baseada em um ou mais dos seguintes elementos:

- Prefixo de nome de chave de objeto – embora o modelo de dados do Amazon S3 seja uma estrutura plana, você pode inferir a hierarquia usando um prefixo. O console do Amazon S3 oferece suporte a esses prefixos com o conceito de pastas. Se você filtrar por prefixo, os objetos com mesmo prefixo serão incluídos na configuração de métricas.
- Tag – Você pode adicionar tags, pares de nome de valor de chave, aos objetos. As tags permitem que você encontre e organize objetos com facilidade. Essas tags também podem ser usadas como um filtro para configurações de métricas.

Se você especificar um filtro, somente solicitações que operem em objetos únicos podem corresponder ao filtro e serem incluídas nas métricas relatadas. As solicitações como [Excluir vários objetos](#) e solicitações de Listar não retornam nenhuma métrica para configurações com filtros.

Para solicitar uma filtragem mais complexa, escolha dois ou mais elementos. Somente os objetos que têm todos esses elementos são incluídos na configuração de métricas. Se você não definir filtros, todos os objetos no bucket estão incluídos na configuração de métricas.

## Como adicionar configurações de métricas

Você pode adicionar configurações de métricas a um bucket com o console do Amazon S3, com a AWS CLI ou com a API REST do Amazon S3. Para obter informações sobre como fazer isso no Console de gerenciamento da AWS, consulte [Como configurar métricas de solicitação para um bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

Adicionar configurações de métricas com a AWS CLI

1. Instalar e configurar a AWS CLI. Para obter instruções, consulte [Configurar a interface da linha de comando da AWS](#) no Guia do usuário do AWS Command Line Interface.
2. Abra um terminal.
3. Execute o seguinte comando para adicionar sua configuração de métrica:

```
aws s3api put-bucket-metrics-configuration --endpoint http://s3-us-west-2.amazonaws.com
--bucket bucket-name --id metrics-config-id --metrics-configuration '{"Id":"'metrics-config-id'","Filter":{"Prefix":"prefix1'}}'
```

4. Para verificar se a configuração foi adicionada, execute o seguinte comando:

```
aws s3api get-bucket-metrics-configuration --endpoint http://s3-us-west-2.amazonaws.com
--bucket bucket-name --id metrics-config-id
```

Isso retorna a seguinte resposta:

```
{  
    "MetricsConfiguration": {  
        "Filter": {  
            "Prefix": "prefix1"  
        },  
        "Id": "metrics-config-id"  
    }  
}
```

Você também pode adicionar configurações de métricas de maneira programática com a API REST do Amazon S3. Para obter mais informações, consulte os seguintes tópicos no Amazon Simple Storage Service API Reference:

- [Configuração de métrica PUT Bucket](#)
- [Configuração de métrica GET Bucket](#)
- [Configuração de métrica List Bucket](#)
- [Configuração de métrica DELETE Bucket](#)

## Registro em log de chamadas à API do Amazon S3 usando o AWS CloudTrail

O Amazon S3 é integrado com o AWS CloudTrail, um serviço que fornece um registro das ações executadas por um usuário, função ou um serviço do AWS em Amazon S3 faz. O CloudTrail captura um subconjunto das chamadas à API do Amazon S3 como eventos, incluindo as chamadas do console do Amazon S3 e as chamadas de código das APIs do Amazon S3. Se você criar uma trilha, poderá habilitar a entrega contínua de eventos do CloudTrail para um bucket do Amazon S3, incluindo eventos para o Amazon S3. Se não configurar uma trilha, você ainda poderá visualizar os eventos mais recentes no console do CloudTrail em Event history. Com as informações coletadas pelo CloudTrail, determine a solicitação feita para o Amazon S3, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para saber mais sobre o CloudTrail, incluindo como configurá-lo e habilitá-lo, consulte o [AWS CloudTrail User Guide](#).

## Informações do Amazon S3 no CloudTrail

O CloudTrail está habilitado na sua conta da AWS ao criá-la. Quando a atividade do evento com suporte ocorre no Amazon S3, ela é registrada em um evento do CloudTrail junto com outros eventos de serviços da AWS em Event history (Histórico de eventos). Você pode visualizar, pesquisar e fazer download de eventos recentes em sua conta da AWS. Para obter mais informações, consulte [Visualizar eventos com o histórico de eventos do CloudTrail](#).

Para obter um registro contínuo de eventos em sua conta da AWS, incluindo eventos para o Amazon S3, crie uma trilha. Uma trilha permite CloudTrail para fornecer arquivos de log a um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, ela é aplicada a todas as regiões. A trilha registra eventos em log de todas as regiões na partição da AWS e entrega os arquivos de log ao bucket do Amazon S3 que você especificar. Além disso, é possível configurar outros serviços da AWS para analisar mais profundamente e agir sobre os dados de evento coletados nos logs do CloudTrail. Para obter mais informações, consulte:

- [Visão geral da criação de uma trilha](#)

- [CloudTrail Serviços compatíveis e integrações do](#)
- [Configuração de notificações do Amazon SNS para o CloudTrail](#)
- [Receber arquivos de log do CloudTrail de várias regiões e receber arquivos de log do CloudTrail de várias contas](#)

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar:

- Se a solicitação foi feita com credenciais de usuário da raiz ou do IAM.
- Se a solicitação foi feita com credenciais de segurança temporárias de uma função ou de um usuário federado.
- Se a solicitação foi feita por outro serviço da AWS.

Para obter mais informações, consulte [Elemento userIdentity do CloudTrail](#).

É possível armazenar os arquivos de log em seu bucket pelo tempo que você desejar, mas também é possível definir regras de ciclo de vida do Amazon S3 para arquivar ou excluir os arquivos de log automaticamente. Por padrão, os arquivos de log são criptografados usando server-side encryption (SSE - criptografia server-side) do Amazon S3.

## Ações do nível do bucket do Amazon S3 rastreadas pelo registro do CloudTrail

Por padrão, o CloudTrail registra ações no nível do bucket. Os registros do Amazon S3 são gravados juntamente com outros registros de serviços da AWS em um arquivo de log. O CloudTrail determina quando criar e gravar em um novo arquivo com base no período e no tamanho do arquivo.

As tabelas nesta seção listam as ações no nível do bucket do Amazon S3 que são compatíveis com o registro do CloudTrail.

### Ações do nível do bucket do Amazon S3 rastreadas pelo registro do CloudTrail

Nome da API REST	Nome do Evento da API usado no log do CloudTrail
<a href="#">DELETE bucket</a>	DeleteBucket
<a href="#">DELETE bucket cors</a>	DeleteBucketCors
<a href="#">DELETE Bucket encryption</a>	DeleteBucketEncryption
<a href="#">DELETE Bucket lifecycle</a>	DeleteBucketLifecycle
<a href="#">DELETE política de bucket</a>	DeleteBucketPolicy
<a href="#">DELETE replicação de bucket</a>	DeleteBucketReplication
<a href="#">Atribuição de tags de DELETE Bucket</a>	DeleteBucketTagging
<a href="#">Site de DELETE Bucket</a>	DeleteBucketWebsite
<a href="#">GET Bucket acl</a>	GetBucketAcl
<a href="#">Cors de GET Bucket</a>	GetBucketCors
<a href="#">GET Bucket encryption</a>	GetBucketEncryption
<a href="#">Ciclo de vida de GET Bucket</a>	GetBucketLifecycle

Nome da API REST	Nome do Evento da API usado no log do CloudTrail
GET localização do bucket	GetBucketLocation
Registro de GET Bucket	GetBucketLogging
Notificação de GET Bucket	GetBucketNotification
GET política de bucket	GetBucketPolicy
Replicação do GET Bucket	GetBucketReplication
GET requestPayment de bucket	GetBucketRequestPay
GET marcação de bucket	GetBucketTagging
Versionamento de GET Bucket	GetBucketVersioning
GET em site de bucket	GetBucketWebsite
GET Service (Relacionar todos os buckets)	ListBuckets
PUT bucket	CreateBucket
acl de PUT Bucket	PutBucketAcl
PUT bucket cors	PutBucketCors
PUT Bucket encryption	PutBucketEncryption
Ciclo de vida de PUT Bucket	PutBucketLifecycle
PUT registro de bucket	PutBucketLogging
Notificação de PUT Bucket	PutBucketNotification
PUT política de bucket	PutBucketPolicy
Replicação do PUT Bucket	PutBucketReplication
PUT requestPayment de bucket	PutBucketRequestPay
PUT marcação de bucket	PutBucketTagging
Versionamento de PUT Bucket	PutBucketVersioning
PUT em site de bucket	PutBucketWebsite

Além dessas operações de API, também é possível usar a ação **OPTIONS object** do nível do objeto. Essa ação é tratada como uma ação do nível do bucket no registro do CloudTrail, pois verifica a configuração cors de um bucket.

## Ações no nível do objeto do Amazon S3 rastreadas pelo registro do CloudTrail

Você também pode obter logs do CloudTrail para ações do nível do objeto do Amazon S3. Para fazer isso, especifique o objeto do Amazon S3 para sua trilha. Quando uma ação do nível do objeto ocorre em sua conta, o CloudTrail avalia suas configurações de trilha. Se o evento corresponder ao objeto que você especificou em uma trilha, o evento será registrado. Para obter mais informações, consulte [Como habilitar](#)

[o registro em log no nível do objeto para um bucket do S3 com eventos de dados do AWS CloudTrail?](#) no Guia do usuário do console do Amazon Simple Storage Service e [Eventos de dados no AWS CloudTrail User Guide](#). A tabela a seguir lista as ações do nível do objeto que CloudTrail pode registrar:

Nome da API REST	Nome do Evento da API usado no log do CloudTrail
Anular multipart upload	AbortMultipartUpload
Concluir multipart upload	CompleteMultipartUpload
Objeto DELETE	DeleteObject
Objeto GET	GetObject
ACL de objeto GET	GetObjectAcl
GET atribuição de tags de objeto	GetObjectTagging
GET torrent de objeto	GetObjectTorrent
Objeto HEAD	HeadObject
Iniciar multipart upload	CreateMultipartUpload
Listar partes	ListParts
Objeto POST	PostObject
POST restauração de objeto	RestoreObject
Objeto PUT	PutObject
PUT Object acl	PutObjectAcl
PUT atribuição de tags de objeto	PutObjectTagging
Objeto PUT - Copiar	CopyObject
Conteúdo de objetos do SELECT	SelectObjectContent
Upload de parte	UploadPart
Upload de parte - Copiar	UploadPartCopy

Além dessas operações, você pode usar as seguintes operações no nível do bucket para obter logs do CloudTrail como ações no nível do objeto do Amazon S3 sob certas condições:

- [GET Bucket \(Listar objetos\) versão 2](#) – selecione um prefixo especificado na trilha.
- [Versões de objeto de GET Bucket](#) – selecione um prefixo especificado na trilha.
- [HEAD Bucket](#) – especifique um bucket e um prefixo vazio.
- [Excluir vários objetos](#) – especifique um bucket e um prefixo vazio.

## Ações do nível do objeto em cenários entre contas

Veja a seguir casos de uso especiais envolvendo APIs no nível do objeto em cenários em várias contas e como os logs do CloudTrail são relatados. O CloudTrail sempre entrega os logs ao solicitante (que fez a chamada de API). Para estabelecer acesso entre contas, considere os exemplos nesta seção.

Note

Os exemplos supõem que os logs do CloudTrail estejam configurados adequadamente.

**Exemplo 1: O CloudTrail entrega logs de acesso ao proprietário do bucket**

O CloudTrail só entrega os logs de acesso ao proprietário do bucket se o proprietário do bucket tiver permissão para a mesma API de objeto. Considere o seguinte cenário entre contas:

- A conta A possui o bucket.
- A conta-B (o solicitante) tenta acessar um objeto nesse bucket.

O CloudTrail entrega sempre os logs de acesso da API do nível do objeto ao solicitante. Além disso, o CloudTrail também entrega os mesmos logs ao proprietário do bucket somente se o proprietário do bucket tiver permissões para as mesmas ações da API sobre esse objeto.

Note

Se o proprietário do bucket também for o proprietário do objeto, o proprietário do bucket obtém os logs de acesso do objeto. Caso contrário, o proprietário do bucket deve obter permissões, por meio da ACL do objeto, para a mesma API do objeto para obter os logs da API de acesso aos objetos.

**Exemplo 2: O CloudTrail não prolifera os endereços de e-mail usados nas configurações das ACLs de objeto**

Considere o seguinte cenário entre contas:

- A conta A possui o bucket.
- A Conta-B (solicitante) envia uma solicitação para definir uma concessão de ACL de objeto usando um endereço de e-mail. Para obter informações sobre ACLs, consulte [Visão geral da Lista de controle de acesso \(ACL\) \(p. 390\)](#).

A solicitação obtém os logs junto com as informações do e-mail. Contudo, o proprietário do bucket — se for qualificado para receber logs como no exemplo 1 — recebe o log do CloudTrail que relata o evento. Contudo, o proprietário do bucket não obtém as informações de configuração da ACL, especificamente o e-mail do favorecido e a concessão. A única informação que o log dá ao proprietário do bucket é que a chamada da API da ACL foi feita pela Conta-B.

## Rastreamento do CloudTrail com chamadas da API SOAP do Amazon S3

O CloudTrail rastreia as chamadas da API SOAP do Amazon S3. O suporte de SOAP via HTTP do Amazon S3 está obsoleto, mas continua disponível via HTTPS. Para obter mais informações sobre o suporte SOAP do Amazon S3, consulte [Apêndice A: uso da API SOAP \(p. 653\)](#).

Important

Os novos recursos do Amazon S3 não são compatíveis com o SOAP. Recomendamos que você use a API REST ou os SDKs da AWS.

### Ações do SOAP do Amazon S3 rastreadas pelo registro do CloudTrail

Nome da API SOAP	Nome do Evento da API usado no log do CloudTrail
<a href="#">ListAllMyBuckets</a>	ListBuckets

Nome da API SOAP	Nome do Evento da API usado no log do CloudTrail
CreateBucket	CreateBucket
DeleteBucket	DeleteBucket
GetBucketAccessControlPolicy	GetBucketAcl
SetBucketAccessControlPolicy	PutBucketAcl
GetBucketLoggingStatus	GetBucketLogging
SetBucketLoggingStatus	PutBucketLogging

## Usar os logs do CloudTrail com os logs de acesso ao servidor do Amazon S3 e com o CloudWatch Logs

Você pode usar os logs do AWS CloudTrail junto com os logs de acesso a servidores para o Amazon S3. Os logs do CloudTrail fornecem rastreamento detalhado da API para operações no nível do bucket e no nível do objeto do Amazon S3, enquanto os logs de acesso a servidores para o Amazon S3 fornecem visibilidade em operações no nível do objeto com seus dados no Amazon S3. Para obter mais informações sobre logs de acesso ao servidor, consulte [Registro em log de acesso ao servidor Amazon S3 \(p. 625\)](#).

Também é possível usar os logs do CloudTrail junto com o CloudWatch para o Amazon S3. A integração do CloudTrail com os logs do CloudWatch fornece a atividade de API no nível do bucket do S3 capturada pelo CloudTrail para um fluxo de log do CloudWatch no grupo de logs do CloudWatch que você especificar. Você pode criar alarmes do CloudWatch para monitoramento de atividade específica de API e receber notificações por e-mail quando a atividade específica de API ocorrer. Para obter mais informações sobre os alarmes do CloudWatch para monitoramento de atividade específica de API, consulte [AWS CloudTrail User Guide](#). Para obter mais informações sobre como usar o CloudWatch com o Amazon S3, consulte [Métricas de monitoramento com o Amazon CloudWatch \(p. 598\)](#).

## Exemplo: entradas do arquivo de log do Amazon S3

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log em um bucket do Amazon S3 que você especificar. Os arquivos de log do CloudTrail contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer origem e inclui informações sobre a ação solicitada, a data e hora da ação, parâmetros de solicitação, e assim por diante. Arquivos de log do CloudTrail não são um rastreamento de pilha ordenada das chamadas da API pública. Assim, elas não são exibidas em nenhuma ordem específica.

O seguinte exemplo mostra uma entrada de log do CloudTrail que demonstra as ações da política [DELETE Bucket](#), [acl de PUT Bucket](#) e [versionamento de GET Bucket](#).

```
{  
    "Records": [  
        {  
            "eventVersion": "1.03",  
            "userIdentity": {  
                "type": "IAMUser",  
                "principalId": "111122223333",  
                "arn": "arn:aws:iam::111122223333:user/myUserName",  
                "accountId": "111122223333",  
                "accessKeyId": "AKIAIOSFODNN7EXAMPLE",  
                "userName": "myUserName"  
            }  
        }  
    ]  
}
```

```
        },
        "eventTime": "2015-08-26T20:46:31Z",
        "eventSource": "s3.amazonaws.com",
        "eventName": "DeleteBucketPolicy",
        "awsRegion": "us-west-2",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "[ ]",
        "requestParameters": {
            "bucketName": "myawsbucket"
        },
        "responseElements": null,
        "requestID": "47B8E8D397DCE7A6",
        "eventID": "cdc4b7ed-e171-4cef-975a-ad829d4123e8",
        "eventType": "AwsApiCall",
        "recipientAccountId": "111122223333"
    },
    {
        "eventVersion": "1.03",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "111122223333",
            "arn": "arn:aws:iam::111122223333:user/myUserName",
            "accountId": "111122223333",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "userName": "myUserName"
        },
        "eventTime": "2015-08-26T20:46:31Z",
        "eventSource": "s3.amazonaws.com",
        "eventName": "PutBucketAcl",
        "awsRegion": "us-west-2",
        "sourceIPAddress": "",
        "userAgent": "[ ]",
        "requestParameters": {
            "bucketName": "",
            "AccessControlPolicy": {
                "AccessControlList": {
                    "Grant": {
                        "Grantee": {
                            "xsi:type": "CanonicalUser",
                            "xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",
                            "ID": "d25639fbe9c19cd30a4c0f43fbf00e2d3f96400a9aa8dabfbbebe1906Example"
                        },
                        "Permission": "FULL_CONTROL"
                    }
                },
                "xmlns": "http://s3.amazonaws.com/doc/2006-03-01/",
                "Owner": {
                    "ID": "d25639fbe9c19cd30a4c0f43fbf00e2d3f96400a9aa8dabfbbebe1906Example"
                }
            }
        },
        "responseElements": null,
        "requestID": "BD8798EACDD16751",
        "eventID": "607b9532-1423-41c7-b048-ec2641693c47",
        "eventType": "AwsApiCall",
        "recipientAccountId": "111122223333"
    },
    {
        "eventVersion": "1.03",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "111122223333",
            "arn": "arn:aws:iam::111122223333:user/myUserName",
            "accountId": "111122223333",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
```

```
        "userName": "myUserName"
    },
    "eventTime": "2015-08-26T20:46:31Z",
    "eventSource": "s3.amazonaws.com",
    "eventName": "GetBucketVersioning",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "",
    "userAgent": "[ ]",
    "requestParameters": {
        "bucketName": "myawsbucket"
    },
    "responseElements": null,
    "requestID": "07D681279BD94AED",
    "eventID": "f2b287f3-0df1-4961-a2f4-c4bdfed47657",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
]
```

## Recursos relacionados

- [AWS CloudTrail User Guide](#)
- [Referência a eventos do CloudTrail](#)

# Usar o BitTorrent com o Amazon S3

## Tópicos

- [Como você será cobrado pela entrega de BitTorrent \(p. 614\)](#)
- [Uso do BitTorrent para recuperar objetos armazenados no Amazon S3 \(p. 615\)](#)
- [Publicação de conteúdo com o Amazon S3 e o BitTorrent \(p. 616\)](#)

O BitTorrent é um protocolo ponto a ponto aberto para distribuição de arquivos. Você pode usar o protocolo BitTorrent para recuperar qualquer objeto publicamente acessível no Amazon S3. Esta seção descreve como você pode desejar usar o BitTorrent para distribuir seus dados fora do Amazon S3 e como fazer isso.

O Amazon S3 oferece suporte ao protocolo BitTorrent para que os desenvolvedores possam economizar custos ao distribuir conteúdo na primeira escala. O Amazon S3 é útil para armazenamento simples e confiável para quaisquer dados. O mecanismo de distribuição padrão para dados do Amazon S3 é via download de cliente/servidor. Na distribuição de cliente/servidor, o objeto inteiro é transferido ponto a ponto do Amazon S3 para cada usuário autorizado que solicitar esse objeto. Embora a entrega de cliente/servidor seja adequada para uma grande variedade de casos de uso, ela não é ideal para todos. Especificamente, os custos da distribuição de cliente/servidor aumentam linearmente conforme aumenta o número de usuários que fazem download de objetos. Isso pode torná-lo caro para distribuição de objetos populares.

O BitTorrent resolve esse problema recrutando os próprios clientes que estão fazendo download do objeto como distribuidores: cada cliente faz download de algumas partes do objeto do Amazon S3 e algumas de outros clientes, fazendo o upload das partes do mesmo objeto para outros "elementos" interessados. O benefício para os publicadores é que, para arquivos populares grandes, a quantidade de dados fornecida pelo Amazon S3 pode ser substancialmente menor do que seria ao atender os mesmos clientes via download de cliente/servidor. Menos dados transferidos significa custos reduzidos para o publicador do objeto.

### Note

Você pode obter o torrent apenas para objetos com tamanho menor que 5 GB.

## Como você será cobrado pela entrega de BitTorrent

Não há sobretaxa para uso do BitTorrent com o Amazon S3. A transferência de dados via protocolo BitTorrent é medida na mesma taxa da entrega de cliente/servidor. Para ser exato, sempre que o download de um cliente do BitTorrent solicita um "pedaço" de um objeto do "seeder" do Amazon S3, as cobranças são acumuladas como se uma solicitação anônima desse pedaço fosse feita usando o protocolo REST ou SOAP. Essas cobranças aparecerão na fatura e nos relatórios de uso do Amazon S3 da mesma forma. A diferença é que, se muitos clientes solicitarem o mesmo objeto simultaneamente via BitTorrent, a quantidade de dados que o Amazon S3 deve servir para satisfazer esses clientes será menor do que a da entrega de cliente/servidor. Isso porque os clientes do BitTorrent fazem upload e download entre si simultaneamente.

### Note

O suporte de SOAP via HTTP está obsoleto, mas continua disponível via HTTPS. Novos recursos do Amazon S3 não serão compatíveis com o SOAP. Recomendamos que você use o REST API ou os SDKs da AWS.

A economia de transferência de dados conseguida com o uso do BitTorrent pode variar amplamente dependendo da popularidade do objeto. Os objetos menos populares exigem um uso mais pesado do "seeder" para servir os clientes, e a diferença entre custos de distribuição do BitTorrent e custos de distribuição de cliente/servidor pode ser pequena para tais objetos. Especificamente, se apenas um cliente fizer download de um objeto específico por vez, o custo de entrega do BitTorrent será o mesmo do download direto.

## Uso do BitTorrent para recuperar objetos armazenados no Amazon S3

Os objetos no Amazon S3 que podem ser lidos anonimamente também podem ser baixados via BitTorrent. Fazer isso requer o uso de um aplicativo cliente BitTorrent. A Amazon não distribui um aplicativo cliente BitTorrent, mas há muitos clientes gratuitos disponíveis. A implementação do Amazon S3BitTorrent foi testada para trabalhar com o cliente BitTorrent oficial (acesse <http://www.bittorrent.com/>).

O ponto inicial para um download do BitTorrent é um arquivo .torrent. Esse arquivo pequeno descreve para clientes BitTorrent os dados a serem baixados e onde começar a encontrar esses dados. Um arquivo .torrent é uma pequena fração do tamanho do objeto real a ser baixado. Assim que você executa um arquivo .torrent gerado pelo Amazon S3 do aplicativo cliente BitTorrent, ele deve começar a ser baixado imediatamente do Amazon S3 and nos clientes BitTorrent "pares".

É fácil recuperar um arquivo .torrent para qualquer objeto publicamente disponível. Basta adicionar um parâmetro de string de consulta "?torrent" ao final da solicitação REST GET do objeto. Nenhuma autenticação é necessária. Assim que você tiver um cliente BitTorrent instalado, o download de um objeto usando o download do BitTorrent pode ser tão fácil quanto abrir esse URL em seu navegador da web.

Não há nenhum mecanismo para buscar o .torrent para um objeto do Amazon S3 usando a API SOAP.

### Note

O suporte de SOAP via HTTP está obsoleto, mas continua disponível via HTTPS. Novos recursos do Amazon S3 não serão compatíveis com o SOAP. Recomendamos que você use o REST API ou os SDKs da AWS.

### Example

Este exemplo recupera o arquivo Torrent para o objeto "Nelson" no bucket "quotes".

### Sample Request

```
GET /quotes/Nelson?torrent HTTP/1.0
Date: Wed, 25 Nov 2009 12:00:00 GMT
```

### Sample Response

```
HTTP/1.1 200 OK
x-amz-request-id: 7CD745EBB7AB5ED9
Date: Wed, 25 Nov 2009 12:00:00 GMT
Content-Disposition: attachment; filename=Nelson.torrent;
Content-Type: application/x-bittorrent
Content-Length: 537
Server: AmazonS3

<body: a Bencoded dictionary as defined by the BitTorrent specification>
```

## Publicação de conteúdo com o Amazon S3 e o BitTorrent

Todo objeto legível de maneira anônima armazenado no Amazon S3 está automaticamente disponível para download usando o BitTorrent. O processo para de alteração da ACL em um objeto para permitir operações `READ` anônimas é descrito em [Gerenciamento de permissões de acesso aos recursos do Amazon S3 \(p. 282\)](#).

Você pode direcionar os clientes para seus objetos acessíveis pelo BitTorrent fornecendo a eles o arquivo `.torrent` diretamente ou publicando um link para a URL do `?torrent` do objeto. É importante salientar que o arquivo `.torrent` que descreve um objeto do Amazon S3 é gerado por demanda, na primeira vez que ele é solicitado (por meio do recurso `?torrent REST`). A geração do `.torrent` de um objeto demora um tempo proporcional ao tamanho desse objeto. Para objetos grandes, esse tempo pode ser significativo. Dessa forma, antes de publicar um link do `?torrent`, sugerimos fazer a primeira solicitação para você mesmo. O Amazon S3 pode levar alguns minutos para responder à primeira solicitação ao gerar o arquivo `.torrent`. A menos que você atualize o objeto em questão, as solicitações subsequentes para o `.torrent` serão rápidas. Seguindo este procedimento antes de implantar um link para o `?torrent` garantirá uma experiência de download do BitTorrent suave para seus clientes.

Para parar a distribuição de um arquivo usando o BitTorrent, simplesmente remova o acesso anônimo a ele. Isso pode ser realizado excluindo o arquivo do Amazon S3 ou modificando a política de controle de acesso para proibir leituras anônimas. Depois disso, o Amazon S3 não atuará como um “propagador” na rede do BitTorrent para seu arquivo, e não atenderá o arquivo `.torrent` por meio da API REST `?torrent`. No entanto, após a publicação de um `.torrent` para seu arquivo, essa ação talvez não interrompa downloads públicos do objeto que acontecem exclusivamente usando a rede ponto a ponto do BitTorrent.

# Tratar erros de REST e SOAP

## Tópicos

- [A resposta de erro de REST \(p. 617\)](#)
- [A resposta de erro de SOAP \(p. 619\)](#)
- [Melhores práticas para erros do Amazon S3 \(p. 619\)](#)

Esta seção descreve os erros de REST e de SOAP e como tratá-los.

### Note

O suporte de SOAP via HTTP está obsoleto, mas continua disponível via HTTPS. Novos recursos do Amazon S3 não serão compatíveis com o SOAP. Recomendamos que você use o REST API ou os SDKs da AWS.

## A resposta de erro de REST

### Tópicos

- [Cabeçalhos de resposta \(p. 617\)](#)
- [Resposta de erro \(p. 618\)](#)

Se uma solicitação de REST resultar em um erro, a resposta HTTP terá:

- Um documento de erro XML como o corpo da resposta
- Content-Type: application/xml
- O código de status HTTP 3xx, 4xx ou 5xx apropriado

O seguinte é um exemplo de uma resposta de erro de REST.

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>NoSuchKey</Code>
  <Message>The resource you requested does not exist</Message>
  <Resource>/mybucket/myfoto.jpg</Resource>
  <RequestId>4442587FB7D0A2F9</RequestId>
</Error>
```

Para obter mais informações sobre erros do Amazon S3, visite [ErrorCodeList](#).

## Cabeçalhos de resposta

Os seguintes são cabeçalhos de resposta retornados por todas as operações:

- **x-amz-request-id**: Um ID exclusivo atribuído a cada solicitação pelo sistema. No caso improvável de problemas com o Amazon S3, a Amazon pode usar esse ID para ajudar a resolver o problema.
- **x-amz-id-2**: Um token especial que nos ajudará a solucionar problemas.

## Resposta de erro

### Tópicos

- [Código de erro \(p. 618\)](#)
- [Mensagem de erro \(p. 618\)](#)
- [Detalhes adicionais \(p. 618\)](#)

Quando uma solicitação do Amazon S3 está em erro, o cliente recebe uma resposta de erro. O formato exato de resposta de erro é específico à API: por exemplo, a resposta de erro de REST difere da resposta de erro de SOAP. Contudo, todas as respostas de erro têm elementos comuns.

#### Note

O suporte de SOAP via HTTP está obsoleto, mas continua disponível via HTTPS. Novos recursos do Amazon S3 não serão compatíveis com o SOAP. Recomendamos que você use o REST API ou os SDKs da AWS.

## Código de erro

O código de erro é uma sequência que identifica exclusivamente uma condição de erro. O objetivo desse código é ser lido e compreendido pelos programas que detectam e tratam erros por tipo. Muitos códigos de erro são comuns entre as APIs SOAP e REST, mas alguns são específicos à API. Por exemplo, `NoSuchKey` é universal, mas `UnexpectedContent` pode ocorrer apenas em resposta a uma solicitação inválida do REST. Em todos os casos, os códigos com falha de SOAP têm um prefixo, conforme indicado na tabela de códigos de erro, para que um erro de `NoSuchKey` seja realmente retornado em SOAP como `Client.NoSuchKey`.

#### Note

O suporte de SOAP via HTTP está obsoleto, mas continua disponível via HTTPS. Novos recursos do Amazon S3 não serão compatíveis com o SOAP. Recomendamos que você use o REST API ou os SDKs da AWS.

## Mensagem de erro

A mensagem de erro contém uma descrição genérica da condição do erro em inglês. Ela é destinada ao público humano. Programas simples exibem a mensagem diretamente ao usuário final se encontrarem uma condição de erro que não conhecem ou não tratam. Programas sofisticados com tratamento de erro mais exaustivo e internacionalização própria são mais prováveis de ignorar a mensagem de erro.

## Detalhes adicionais

Muitas respostas de erro contêm dados estruturados adicionais para serem lidos e compreendidos pelo desenvolvedor que diagnostica erros de programação. Por exemplo, se você enviar um cabeçalho `Content-MD5` com uma solicitação PUT de REST que não corresponde ao resumo calculado no servidor, você receberá um erro `BadDigest`. A resposta do erro também inclui como elementos de detalhes o resumo que calculamos, e o resumo que você nos informou para esperar. Durante o desenvolvimento, você pode usar essas informações para diagnosticar o erro. Em produção, um programa bem-comportado pode incluir essas informações em seu log de erros.

## A resposta de erro de SOAP

### Note

O suporte de SOAP via HTTP está obsoleto, mas continua disponível via HTTPS. Novos recursos do Amazon S3 não serão compatíveis com o SOAP. Recomendamos que você use o REST API ou os SDKs da AWS.

Em SOAP, um resultado de erro será retornado ao cliente como uma falha de SOAP, com o código de resposta HTTP 500. Se você não receber uma falha de SOAP, sua solicitação terá sido bem-sucedida. O código de falha de SOAP do Amazon S3 é composto de um código de falha padrão de SOAP 1.1 ("servidor" ou "cliente") concatenado com o código de erro específico do Amazon S3. Por exemplo: "Server.InternalError" ou "Client.NoSuchBucket". O elemento da sequência da falha de SOAP contém uma mensagem de erro genérica, legível pelo usuário em inglês. Finalmente, o elemento de detalhes da falha de SOAP contém informações diversas relevantes para o erro.

Por exemplo, se você tentar excluir o objeto "Fred", que não existe, o corpo da resposta de SOAP conterá uma falha de SOAP "NoSuchKey".

### Example

```
<soapenv:Body>
<soapenv:Fault>
  <Faultcode>soapenv:Client.NoSuchKey</Faultcode>
  <Faultstring>The specified key does not exist.</Faultstring>
  <Detail>
    <Key>Fred</Key>
  </Detail>
</soapenv:Fault>
</soapenv:Body>
```

Para obter mais informações sobre erros do Amazon S3, visite [ErrorCodeList](#).

## Melhores práticas para erros do Amazon S3

Ao desenvolver um aplicativo para uso com o Amazon S3, é importante tratar os erros do Amazon S3 de maneira adequada. Esta seção descreve os problemas a serem considerados ao desenvolver um aplicativo.

### Tentar InternalErrors novamente

Os erros internos são erros que ocorrem no ambiente do Amazon S3.

Solicitações que recebem uma resposta de InternalError podem não ter sido processadas. Por exemplo, se uma solicitação PUT retornar um InternalError, um GET subsequente poderá recuperar o valor antigo ou o valor atualizado.

Se o Amazon S3 retornar uma resposta de InternalError, tente a solicitação novamente.

### Ajuste o aplicativo para erros repetidos de SlowDown

Como com qualquer sistema distribuído, o S3 tem mecanismos de proteção que detectam consumo excessivo intencional ou involuntário de recursos e reagem adequadamente. Os erros de SlowDown podem ocorrer quando uma alta taxa de solicitações aciona um desses mecanismos. Reduzir a taxa de solicitações reduzirá ou eliminará erros desse tipo. De modo geral, a maioria dos usuários

não experimentará esses erros regularmente. Contudo, se você desejar mais informações ou estiver recebendo erros de SlowDown inesperados ou em quantidade, faça uma postagem em nosso Fórum de desenvolvimento do Amazon S3 ou cadastre-se no AWS Premium Support<https://forums.aws.amazon.com/https://aws.amazon.com/premiumsupport/>.

## Erros isolados

### Note

O suporte de SOAP via HTTP está obsoleto, mas continua disponível via HTTPS. Novos recursos do Amazon S3 não serão compatíveis com o SOAP. Recomendamos que você use o REST API ou os SDKs da AWS.

O Amazon S3 fornece um conjunto de códigos de erro que são usados pelas APIs SOAP e REST. A API SOAP retorna códigos de erro padrão do Amazon S3. A API REST é desenvolvida para parecer um servidor HTTP padrão e interagir com clientes HTTP existentes (por exemplo, navegadores, bibliotecas de cliente HTTP, proxies, caches etc.) Para garantir que os clientes HTTP tratem erros corretamente, mapeamos cada erro do Amazon S3 para um código de status HTTP.

Os códigos de status HTTP são menos expressivos que os códigos de erro do Amazon S3 e contêm menos informações sobre o erro. Por exemplo, os erros `NoSuchKey` e `NoSuchBucket` do Amazon S3 são mapeados para o código de status `HTTP 404 Not Found`.

Embora os códigos de status HTTP contenham menos informações sobre o erro, os clientes que entendem HTTP, mas não a API do Amazon S3, geralmente tratam o erro corretamente.

Portanto, para lidar com erros ou relatar erros do Amazon S3 para os usuários finais, use o código de erro do Amazon S3 em vez do código de status HTTP, uma vez que ele contém a maioria das informações sobre o erro. Além disso, ao depurar o aplicativo, você também deve consultar o elemento `<Details>` legível pelo usuário da resposta de erro XML.

# Solução de problemas do Amazon S3

Esta seção descreve como solucionar problemas do Amazon S3 e explica como obter os IDs de solicitação necessários quando você entrar em contato com o AWS Support.

## Tópicos

- [Solucionar problemas do Amazon S3 por sintoma \(p. 621\)](#)
- [Obter os IDs da solicitação do Amazon S3 para o AWS Support \(p. 622\)](#)
- [Tópicos relacionados \(p. 624\)](#)

## Solucionar problemas do Amazon S3 por sintoma

Os tópicos a seguir listam os sintomas para ajudar a solucionar alguns dos problemas que você pode encontrar ao trabalhar com o Amazon S3.

### Sintomas

- [Aumentos significativos em respostas HTTP 503 para solicitações do Amazon S3 para buckets com o versionamento habilitado \(p. 621\)](#)
- [Comportamento inesperado ao acessar buckets definidos com CORS \(p. 621\)](#)

## Aumentos significativos em respostas HTTP 503 para solicitações do Amazon S3 para buckets com o versionamento habilitado

Se você perceber um aumento significativo do número de respostas HTTP 503 recebidas com lentidão do Amazon S3 de solicitações PUT ou DELETE de objetos a um bucket que tenha versionamento habilitado, talvez tenha um ou mais objetos no bucket para os quais há milhões de versões. Quando você tem objetos com milhões de versões, o Amazon S3 limita automaticamente as solicitações ao bucket para proteger o cliente de uma quantidade excessiva de tráfego de solicitação, o que pode potencialmente impedir outras solicitações feitas ao mesmo bucket.

Para determinar quais objetos do S3 têm milhões de versões, use a ferramenta de inventário do Amazon S3. A ferramenta de inventário gera um relatório que fornece uma lista de arquivos simples dos objetos em um bucket. Para obter mais informações, consulte [Inventário do Amazon S3 \(p. 273\)](#).

A equipe do Amazon S3 incentiva os clientes a investigarem os pedidos que sobrescrevem repetidamente o mesmo objeto do S3, criando potencialmente milhões de versões desse objeto, para determinar se o aplicativo está funcionando conforme o esperado. Se você tiver um caso de uso que exija milhões de versões para um ou mais objetos do S3, entre em contato com a equipe de suporte da AWS no [AWS Support](#) para discutir seu caso de uso e nos ajudar a dar uma assistência ótima para o cenário do seu caso de uso.

## Comportamento inesperado ao acessar buckets definidos com CORS

Se você encontrar comportamento inesperado ao acessar buckets definidos com o compartilhamento de recurso entre origens (CORS), consulte [Solução de problemas do CORS \(p. 165\)](#).

# Obter os IDs da solicitação do Amazon S3 para o AWS Support

Sempre que você precisar entrar em contato com o AWS Support devido à erros ou comportamento inesperado no Amazon S3, você precisará obter os IDs das solicitações associadas à ação com falha. A obtenção desses IDs permite que o AWS Support ajude você a resolver os problemas que está enfrentando. Os IDs da solicitação são fornecidos em pares, são retornados em cada resposta que o Amazon S3 processa (mesmo errôneas) e podem ser acessados por meio de logs detalhados. Há vários métodos comuns para obter os IDs de solicitação.

Depois de recuperar esses logs, copie e mantenha esses dois valores, pois você precisará deles ao entrar em contato com o AWS Support. Para obter informações sobre o AWS Support, consulte [Entre em contato conosco](#).

## Tópicos

- [Usar o HTTP para obter IDs de solicitação \(p. 622\)](#)
- [Usar um navegador da web para obter IDs de solicitação \(p. 622\)](#)
- [Usar os AWS SDKs para obter IDs de solicitação \(p. 623\)](#)
- [Usar o AWS CLI para obter IDs de solicitação \(p. 624\)](#)

## Usar o HTTP para obter IDs de solicitação

Você pode obter os IDs da solicitação, `x-amz-request-id` e `x-amz-id-2`, registrando os bits de uma solicitação HTTP antes que ela chegue ao aplicativo de destino. Há várias ferramentas de terceiros que podem ser usadas para recuperar logs detalhados de solicitações HTTP. Escolha uma de sua confiança e execute-a, ouvindo na porta em que o tráfego do Amazon S3 viaja ao enviar outra solicitação HTTP do Amazon S3.

Para solicitações HTTP, o par de IDs da solicitação será parecido com os exemplos a seguir.

```
x-amz-request-id: 79104EXAMPLEB723
x-amz-id-2: IOWQ4fDEXAMPLEQM+ey7N9WgVhSnQ6JEXAMPLEZb7hSQDASK+Jd1vEXAMPLEa3Km
```

### Note

As solicitações HTTP estão criptografadas e ocultas na maioria das capturas de pacotes.

## Usar um navegador da web para obter IDs de solicitação

A maioria dos navegadores da web têm ferramentas de desenvolvedor que permitem que você visualize os cabeçalhos das solicitações.

Para solicitações baseadas em navegador da web que retornam um erro, o par de IDs de solicitações será parecido com os exemplos a seguir.

```
<Error><Code>AccessDenied</Code><Message>Access Denied</Message>
<RequestId>79104EXAMPLEB723</RequestId><HostId>IOWQ4fDEXAMPLEQM
+ey7N9WgVhSnQ6JEXAMPLEZb7hSQDASK+Jd1vEXAMPLEa3Km</HostId></Error>
```

Para obter o par de IDs de solicitações bem-sucedidas, você precisa usar as ferramentas de desenvolvedor para ver os cabeçalhos das respostas HTTP. Para obter informações sobre as ferramentas

de desenvolvedor para navegadores específicos, consulte Solução de problemas do Amazon S3 - como recuperar os IDs de solicitações do S3 nos Fóruns de desenvolvimento da AWS.

## Usar os AWS SDKs para obter IDs de solicitação

As seções a seguir incluem informações para configuração do log usando o AWS SDK. Embora seja possível habilitar log detalhado em cada solicitação e resposta, você não deve habilitar o log em sistemas de produção uma vez que solicitações/respostas grandes podem provocar lentidão significativa em um aplicativo.

Para solicitações do AWS SDK, o par de IDs da solicitação será parecido com os exemplos a seguir.

```
Status Code: 403, AWS Service: Amazon S3, AWS Request ID: 79104EXAMPLEB723
AWS Error Code: AccessDenied AWS Error Message: Access Denied
S3 Extended Request ID: IOWQ4fDEXAMPLEQM+ey7N9WgVhSnQ6JEXAMPLEb7hSQDASK+Jd1vEXAMPLEa3Km
```

### Usar o SDK para PHP para obter IDs de solicitação

Você pode configurar o log usando o PHP. Para obter mais informações, consulte [Como posso ver quais dados são enviados pela rede?](#) nas Perguntas frequentes sobre o AWS SDK para PHP.

### Usar o SDK para Java para obter IDs de solicitação

Você pode habilitar o log para solicitações ou respostas específicas, para permitir a captura e o retorno apenas de cabeçalhos relevantes. Para fazer isso, importe a classe `com.amazonaws.services.s3.S3ResponseMetadata`. Mais tarde, você pode armazenar a solicitação em uma variável antes de executar a solicitação real. Chame `getCachedResponseMetadata(AmazonWebServiceRequest request).getRequestID()` para obter a solicitação ou a resposta registrada em log.

#### Example

```
PutObjectRequest req = new PutObjectRequest(bucketName, key, createSampleFile());
s3.putObject(req);
S3ResponseMetadata md = s3.getCachedResponseMetadata(req);
System.out.println("Host ID: " + md.getHostId() + " RequestID: " + md.getRequestId());
```

Como alternativa, você pode usar log detalhado de cada solicitação e resposta de Java. Para obter mais informações, consulte [Log detalhado da conexão](#) no tópico Registrar chamadas do AWS SDK para Java no AWS SDK for Java Developer Guide.

### Usar o AWS SDK para .NET para obter IDs de solicitação

Você pode configurar o log no AWS SDK para .NET usando a ferramenta incorporada de log `System.Diagnostics`. Para obter mais informações, consulte a postagem [Registrar em log com o AWS SDK para .NET](#) no blog de desenvolvimento da AWS.

#### Note

Por padrão, o log retornado contém somente informações de erros. O arquivo de configuração precisa ter `AWSLogMetrics` (e, opcionalmente, `AWSResponseLogging`) adicionado para obter os IDs de solicitação.

### Usar o SDK for Python para obter IDs de solicitação

Você pode configurar o log no Python adicionando as seguintes linhas ao código para obter a saída das informações de depuração em um arquivo.

```
import logging
logging.basicConfig(filename="mylog.log", level=logging.DEBUG)
```

Se estiver usando a interface do Boto Python para a AWS, você poderá definir o nível de depuração como dois conforme a documentação do Boto, [aqui](#).

## Usar o SDK para Ruby para obter IDs de solicitação

Você pode obter os IDs de solicitação usando o SDK para Ruby - versão 1, 2 ou 3.

- Usar o SDK para Ruby - versão 1 – você pode habilitar o log da conexão HTTP globalmente com a linha de código a seguir.

```
s3 = AWS::S3.new(:logger => Logger.new($stdout), :http_wire_trace => true)
```

- Usar o SDK para Ruby - versão 2 ou 3 – você pode habilitar o log da conexão HTTP globalmente com a linha de código a seguir.

```
s3 = Aws::S3::Client.new(:logger => Logger.new($stdout), :http_wire_trace => true)
```

## Usar o AWS CLI para obter IDs de solicitação

Você pode obter os IDs de solicitação na AWS CLI adicionando --debug ao comando.

## Tópicos relacionados

Para obter outros tópicos de como solucionar problemas e obter suporte, consulte o seguinte:

- [Solução de problemas do CORS \(p. 165\)](#)
- [Tratar erros de REST e SOAP \(p. 617\)](#)
- [Documentação do AWS Support](#)

Para obter informações de como solucionar problemas relacionados a ferramentas de terceiros, consulte [Obter os IDs de solicitação do Amazon S3 nos Fóruns de desenvolvimento da AWS](#).

# Registro em log de acesso ao servidor Amazon S3

O registro em log de acesso ao servidor fornece detalhes sobre as solicitações que são feitas a um bucket. Os logs de acesso ao servidor são úteis para muitos aplicativos. Por exemplo, as informações do log de acesso podem ser úteis em auditorias de segurança e acesso. Isso também pode ajudar a conhecer sua base de clientes e entender sua fatura do Amazon S3.

## Tópicos

- [Como habilitar o registro em log de acesso ao servidor \(p. 625\)](#)
- [Formato da chave de objeto de log \(p. 626\)](#)
- [Como os logs são entregues? \(p. 627\)](#)
- [Entrega de logs pelo servidor de melhor esforço \(p. 627\)](#)
- [As alterações do status do registro de bucket em logs entram em vigor ao longo do tempo \(p. 627\)](#)
- [Habilitar o registro usando o console \(p. 627\)](#)
- [Habilitar o log por programação \(p. 628\)](#)
- [Formato do log de acesso ao servidor \(p. 631\)](#)
- [Excluir arquivos de log do Amazon S3 \(p. 638\)](#)

## Como habilitar o registro em log de acesso ao servidor

Para acompanhar as solicitações de acesso ao seu bucket, habilite o registro em log de acesso ao servidor. Cada registro do log de acesso fornece detalhes sobre uma única solicitação de acesso, como solicitante, nome do bucket, horário da solicitação, ação da solicitação, status da resposta e um código de erro, se relevante.

### Note

Não existe cobrança adicional para habilitar o registro de log de acesso ao servidor de um bucket do Amazon S3. No entanto, todos os arquivos de log entregues pelo sistema acumularão cobranças normais de armazenamento. Você pode excluir os arquivos de registro a qualquer momento. Não há cobrança de transferências de dados para a entrega dos arquivos de log, mas o acesso a esses arquivos é cobrado da mesma forma que qualquer outra transferência de dados.

Por padrão, o log está desabilitado. Quando o registro de log está habilitado, os logs são salvos em um bucket na mesma região da AWS em que o bucket de origem está.

Para habilitar o registro de acesso em logs, faça o seguinte:

- Ative a entrega de logs adicionando a configuração de log no bucket para o qual deseja que o Amazon S3 entregue os logs de acesso. Chamaremos esse bucket de bucket de origem.
- Conceda permissão ao grupo de Entrega de logs do Amazon S3 para gravação no bucket em que deseja que os logs de acesso sejam salvos. Chamaremos esse bucket de bucket de destino.

Para ativar a entrega de logs, forneça as seguintes informações de configuração de logs:

- O nome do bucket de destino em que você deseja que o Amazon S3 salve os logs de acesso como objetos. Os logs podem ser entregues a qualquer bucket que você possui e que esteja na mesma região que o bucket de origem, incluindo o próprio bucket de origem.

Recomendamos que você salve os logs de acessos em um bucket diferente para facilitar o gerenciamento dos logs. Se você optar por salvar os logs de acesso no bucket de origem, recomendamos que você especifique um prefixo para todas as chaves de objeto do log, para que os nomes do objeto comecem com uma string em comum e seja fácil identificar esses objetos.

Quando o bucket de origem e o bucket de destino são os mesmos, logs adicionais são criados para os logs que forem gravados no bucket. Esse comportamento pode não ser ideal para o seu caso de uso, pois pode resultar em um pequeno aumento no faturamento de armazenamento. Além disso, com os logs adicionais sobre logs, pode ser mais difícil encontrar o log que você está procurando.

#### Note

Os buckets de origem e de destino devem ser de propriedade da mesma conta da AWS e devem estar na mesma região.

- (Opcional) Um prefixo para o Amazon S3 atribuir a todas as chaves de objeto de log. O prefixo facilita a localização de objetos de log.

Por exemplo, se você especificar o valor do prefixo `logs/`, cada objeto de log criado pelo Amazon S3 começará com o prefixo `logs/` na sua chave, conforme o exemplo a seguir:

```
logs/2013-11-01-21-32-16-E568B2907131C0C0
```

O prefixo da chave ajuda ao excluir os logs. Por exemplo, você pode definir uma regra de configuração de ciclo de vida para que o Amazon S3 exclua objetos com um prefixo de chave específico. Para obter mais informações, consulte [Excluir arquivos de log do Amazon S3 \(p. 638\)](#).

- (Opcional) Permissões para que outros possam acessar os logs gerados. Por padrão, o proprietário do bucket sempre tem acesso completo aos objetos de log. Se quiser, você tem a opção de conceder acesso a outros usuários.

Para obter mais informações sobre a ativação do registro em log de acesso ao servidor, consulte [Habilitar o registro usando o console \(p. 627\)](#) e [Habilitar o log por programação \(p. 628\)](#).

## Formato da chave de objeto de log

O Amazon S3 usa o formato de chave de objeto a seguir para os objetos de log carregados no bucket de destino:

```
TargetPrefixYYYY-mm-DD-HH-MM-SS-UniqueString
```

Na chave, `YYYY`, `mm`, `DD`, `HH`, `MM` e `SS` são os dígitos do ano, mês, dia, hora, minuto e segundos (respectivamente) quando o arquivo de log foi entregue.

Um arquivo de log entregue em um horário específico pode conter registros gravados a qualquer momento até aquele horário. Não há como saber se todos os logs de um certo intervalo de tempo foram entregues ou não.

O componente `UniqueString` da chave existe para impedir que arquivos sejam substituídos por outros. Ele não tem significado, e o software de processamento de logs deve ignorá-lo.

## Como os logs são entregues?

O Amazon S3 coleta periodicamente os registros de log de acesso, consolida-os em arquivos de log e, em seguida, faz upload desses arquivos no bucket de destino como objetos de log. Caso o registro em log esteja habilitado em diversos buckets de origem que identifiquem o mesmo bucket de destino, ele terá logs de acesso de todos os buckets de origem. No entanto, cada objeto de log relata registros de log para um bucket de origem específico.

O Amazon S3 usa uma conta especial de entrega de logs, chamada de grupo de Entrega de logs, para gravar logs de acesso. Essas gravações estão sujeitas a restrições usuais de controle de acesso. Você deve conceder ao grupo de Entrega de logs a permissão para gravação no bucket de destino adicionando uma entrada de concessão na lista de controle de acesso (ACL) do bucket. Se você usar o console do Amazon S3 para habilitar o registro em log em um bucket, o console habilitará o log no bucket de origem e atualizará a ACL no bucket de destino para conceder a permissão para gravação ao grupo de Entrega de logs.

## Entrega de logs pelo servidor de melhor esforço

Os registros de log de acessos do servidor são entregues com base no melhor esforço. A maioria das solicitações para um bucket configurado corretamente para registro em log tem como resultado um registro do log entregue. A maioria dos registros de log é entregue dentro de algumas horas após o tempo em que forem registrados, mas eles podem ser entregues com mais frequência.

A integralidade e a pontualidade do registro em log do servidor não são garantidas. O registro de log de uma solicitação específica pode ser entregue muito depois de a solicitação ter sido realmente processada ou pode nem ser entregue. A finalidade dos logs do servidor é proporcionar uma ideia da natureza do tráfego no bucket. É raro perder registros de log, mas o log dos servidores não tem como objetivo ser uma contabilidade completa de todas as solicitações.

Levando em conta a natureza de melhor esforço do recurso de log do servidor, os relatórios de uso disponíveis no portal da AWS (relatórios do Gerenciamento de custos e faturamento no [Console de gerenciamento da AWS](#)) podem incluir uma ou mais solicitações de acesso que não aparecem em um log do servidor entregue.

## As alterações do status do registro de bucket em logs entram em vigor ao longo do tempo

As alterações no status do log de um bucket levam tempo para realmente afetar a entrega de arquivos de log. Por exemplo, se você habilitar o log para um bucket, algumas solicitações feitas na hora seguinte podem ser registradas, enquanto outras não. Se você alterar o bucket de destino para log do bucket A para o B, alguns logs podem continuar sendo entregues ao bucket A durante a próxima hora, enquanto outros serão entregues ao novo bucket de destino B. Em todo caso, as novas configurações entrarão em vigor posteriormente, sem a necessidade de ações adicionais.

## Habilitar o registro usando o console

Para obter informações sobre a habilitação de [Registro em log de acesso ao servidor Amazon S3 \(p. 625\)](#) no [Console de gerenciamento da AWS](#), consulte [Como habilitar o log de acessos ao servidor para um bucket do S3?](#) no Guia do usuário do console do Amazon Simple Storage Service.

Ao habilitar o registro em log em um bucket, o console habilita o log no bucket de origem e adiciona uma concessão na lista de controle de acesso (ACL) do bucket de destino, concedendo ao grupo de Entrega de logs a permissão para gravação.

Para obter informações sobre como habilitar o log por programação, consulte [Habilitar o log por programação \(p. 628\)](#).

Para obter informações sobre o formato do log, incluindo a lista de campos e suas descrições, consulte [Formato do log de acesso ao servidor \(p. 631\)](#).

## Habilitar o log por programação

Habilite ou desabilite o registro em log por programação usando a API do Amazon S3 ou os SDKs da AWS. Para isso, habilite o log no bucket e conceda permissão ao grupo de Entrega de logs para gravar logs no bucket de destino.

### Tópicos

- [Habilitar registro em log \(p. 628\)](#)
- [Conceder as permissões WRITE e READ\\_ACP ao grupo de Entrega de logs \(p. 629\)](#)
- [Exemplo: AWS SDK para .NET \(p. 629\)](#)
- [Mais informações \(p. 631\)](#)

## Habilitar registro em log

Para habilitar o registro em log, envie uma solicitação [PUT Bucket logging](#) para adicionar a configuração de registro no bucket de origem. A solicitação especifica o bucket de destino e, opcionalmente, o prefixo a ser usado por todas as chaves de objeto dos logs. O exemplo a seguir identifica `logbucket` como o bucket de destino e `logs/` como o prefixo.

```
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <LoggingEnabled>
    <TargetBucket>logbucket</TargetBucket>
    <TargetPrefix>logs/</TargetPrefix>
  </LoggingEnabled>
</BucketLoggingStatus>
```

Os objetos de log são gravados e pertencem à conta de Entrega de logs. O proprietário do bucket tem permissões totais aos objetos de log. Além disso, você tem a opção de conceder permissões a outros usuários para que eles possam acessar os logs. Para obter mais informações, consulte [Registro de bucket em logs PUT](#).

O Amazon S3 também oferece a API [GET Bucket logging](#) para recuperar a configuração de registro em um bucket. Para excluir a configuração de registro, envie a solicitação PUT Bucket logging com um `BucketLoggingStatus` vazio.

```
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">
</BucketLoggingStatus>
```

Use a API do Amazon S3 ou as bibliotecas wrapper do SDK da AWS para habilitar o registro em log em um bucket.

## Conceder as permissões WRITE e READ\_ACP ao grupo de Entrega de logs

O Amazon S3 grava os arquivos de log no bucket de destino como um membro do grupo predefinido de Entrega de logs do Amazon S3. Essas gravações estão sujeitas a restrições usuais de controle de acesso. Você deve conceder as permissões `s3:GetObjectAcl` e `s3:PutObject` a esse grupo adicionando concessões na lista de controle de acesso (ACL) do bucket de destino. O grupo de Entrega de logs é representado pelo seguinte URL.

```
http://acs.amazonaws.com/groups/s3/LogDelivery
```

Para conceder permissões WRITE e READ\_ACP, adicione use as concessões a seguir. Para obter informações sobre ACLs, consulte [Gerenciar o acesso com ACLs \(p. 390\)](#).

```
<Grant>
    <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
        <URI>http://acs.amazonaws.com/groups/s3/LogDelivery</URI>
    </Grantee>
    <Permission>WRITE</Permission>
</Grant>
<Grant>
    <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
        <URI>http://acs.amazonaws.com/groups/s3/LogDelivery</URI>
    </Grantee>
    <Permission>READ_ACP</Permission>
</Grant>
```

Para exemplos de como adicionar concessões na ACL por programação usando os SDKs da AWS, consulte [Gerenciar ACLs usando o AWS SDK for Java \(p. 396\)](#) e [Gerenciar ACLs usando o AWS SDK para .NET \(p. 399\)](#).

## Exemplo: AWS SDK para .NET

O exemplo C# a seguir habilita o log em um bucket. Você deve criar dois buckets, o de origem e o de destino. Primeiro, o exemplo concede as permissões necessárias ao grupo de Entrega de logs para gravação de logs no bucket de destino e, em seguida, habilita o registro em log no bucket de origem. Para obter mais informações, consulte [Habilitar o log por programação \(p. 628\)](#). Para obter instruções sobre como criar e testar um exemplo funcional, consulte [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#).

### Example

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-s3-developer-guide/blob/master/LICENSE-SAMPLECODE.)

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

namespace Amazon.DocSamples.S3
{
    class ServerAccessLoggingTest
    {
        private const string bucketName = "**** bucket name for which to enable logging
****";
```

```
private const string targetBucketName = "*** bucket name where you want access logs
stored ***";
private const string logObjectKeyPrefix = "Logs";
// Specify your bucket region (an example region is shown).
private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
private static IAmazonS3 client;

public static void Main()
{
    client = new AmazonS3Client(bucketRegion);
    EnableLoggingAsync().Wait();
}

private static async Task EnableLoggingAsync()
{
    try
    {
        // Step 1 - Grant Log Delivery group permission to write log to the target
        await GrantPermissionsToWriteLogsAsync();
        // Step 2 - Enable logging on the source bucket.
        await EnableDisableLoggingAsync();
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine("Error encountered on server. Message:{0}' when writing
an object", e.Message);
    }
    catch (Exception e)
    {
        Console.WriteLine("Unknown encountered on server. Message:{0}' when
writing an object", e.Message);
    }
}

private static async Task GrantPermissionsToWriteLogsAsync()
{
    var bucketACL = new S3AccessControlList();
    var aclResponse = client.GetACL(new GetACLRequest { BucketName =
targetBucketName });
    bucketACL = aclResponse.AccessControlList;
    bucketACL.AddGrant(new S3Grantee { URI = "http://acs.amazonaws.com/groups/s3/
LogDelivery" }, S3Permission.WRITE);
    bucketACL.AddGrant(new S3Grantee { URI = "http://acs.amazonaws.com/groups/s3/
LogDelivery" }, S3Permission.READ_ACP);
    var setACLRequest = new PutACLRequest
    {
        AccessControlList = bucketACL,
        BucketName = targetBucketName
    };
    await client.PutACLAsync(setACLRequest);
}

private static async Task EnableDisableLoggingAsync()
{
    var loggingConfig = new S3BucketLoggingConfig
    {
        TargetBucketName = targetBucketName,
        TargetPrefix = logObjectKeyPrefix
    };

    // Send request.
    var putBucketLoggingRequest = new PutBucketLoggingRequest
    {
        BucketName = bucketName,
        LoggingConfig = loggingConfig
    }
}
```

```
        };
        await client.PutBucketLoggingAsync(putBucketLoggingRequest);
    }
}
```

## Mais informações

- [Registro em log de acesso ao servidor Amazon S3 \(p. 625\)](#)
- [AWS::S3::Bucket](#) no Guia do usuário do AWS CloudFormation

## Formato do log de acesso ao servidor

Os arquivos do log de acesso ao servidor consistem em uma sequência de registros do log delimitados por novas linhas. Cada registro do log representa uma solicitação e consiste em campos delimitados por espaço. O seguinte é um log de exemplo que consiste em seis registros em log.

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be mybucket [06/
Feb/2014:00:00:38 +0000] 192.0.2.3
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 3E57427F3EXAMPLE
REST.GET.VERSIONING - "GET /mybucket?versioning HTTP/1.1" 200 - 113 - 7 - "-"
"S3Console/0.4" -
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be mybucket [06/
Feb/2014:00:00:38 +0000] 192.0.2.3
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 891CE47D2EXAMPLE
REST.GET.LOGGING_STATUS - "GET /mybucket?logging HTTP/1.1" 200 - 242 - 11 - "-"
"S3Console/0.4" -
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be mybucket [06/
Feb/2014:00:00:38 +0000] 192.0.2.3
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be A1206F460EXAMPLE
REST.GET.BUCKETPOLICY - "GET /mybucket?policy HTTP/1.1" 404 NoSuchBucketPolicy 297 - 38 -
"-"
"S3Console/0.4" -
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be mybucket [06/
Feb/2014:00:01:00 +0000] 192.0.2.3
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 7B4A0FABBEXAMPLE
REST.GET.VERSIONING - "GET /mybucket?versioning HTTP/1.1" 200 - 113 - 33 - "-"
"S3Console/0.4" -
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be mybucket [06/
Feb/2014:00:01:57 +0000] 192.0.2.3
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be DD6CC733AEXAMPLE
REST.PUT.OBJECT s3-dg.pdf "PUT /mybucket/s3-dg.pdf HTTP/1.1" 200 - - 4406583 41754 28 "-"
"S3Console/0.4" -
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be mybucket [06/
Feb/2014:00:03:21 +0000] 192.0.2.3
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be BC3C074D0EXAMPLE
REST.GET.VERSIONING - "GET /mybucket?versioning HTTP/1.1" 200 - 113 - 28 - "-"
"S3Console/0.4" -
```

### Note

Qualquer campo pode ser definido como – para indicar que os dados eram desconhecidos ou estavam indisponíveis ou que o campo não era aplicável para essa solicitação.

A lista a seguir descreve os campos dos registros em log.

### Proprietário do bucket

O ID canônico do usuário do proprietário do bucket de origem. O ID de usuário canônico é uma outra forma do ID da conta da AWS. Para obter mais informações sobre o ID de usuário canônico, consulte

**Identificadores de conta da AWS.** Para obter informações sobre como encontrar o ID de usuário canônico da conta, consulte [Encontrar o ID de usuário canônico da conta](#).

Entrada de exemplo

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Bucket

O nome do bucket no qual a solicitação foi processada. Se o sistema receber uma solicitação malformada e não puder determinar o bucket, a solicitação não aparecerá em nenhum log de acesso ao servidor.

Entrada de exemplo

```
mybucket
```

Tempo

A hora em que a solicitação foi recebida. O formato que usa a terminologia `strftime()` é o seguinte: `[%d/%b/%Y:%H:%M:%S %z]`

Entrada de exemplo

```
[06/Feb/2014:00:00:38 +0000]
```

IP remoto

O endereço de internet aparente do solicitante. Os proxies e os firewalls intermediários podem obscurecer o endereço real da máquina que faz a solicitação.

Entrada de exemplo

```
192.0.2.3
```

Solicitante

O ID canônico do usuário do solicitante ou um – para solicitações não autenticadas. Se o solicitante for um usuário do IAM, esse campo retorna o nome do usuário do IAM do solicitante junto com a conta raiz da AWS à qual o usuário do IAM pertence. Esse identificador é o mesmo usado para fins de controle de acesso.

Entrada de exemplo

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

ID da solicitação

Uma string gerada pelo Amazon S3 para identificar exclusivamente cada solicitação.

Entrada de exemplo

```
3E57427F33A59F07
```

Operação

A operação listada aqui é declarada como `SOAP.operation`,  
`REST.HTTP_method.resource_type`, `WEBSITE.HTTP_method.resource_type` ou  
`BATCH.DELETE.OBJECT`.

Entrada de exemplo

```
REST.PUT.OBJECT
```

Chave

A parte “chave” da solicitação, codificada pela URL ou “-”, se a operação não usar um parâmetro de chave.

Entrada de exemplo

```
/photos/2014/08/puppy.jpg
```

Request-URI

A parte de Request-URI da mensagem de solicitação HTTP.

Entrada de exemplo

```
"GET /mybucket/photos/2014/08/puppy.jpg?x-foo=bar HTTP/1.1"
```

Status HTTP

O código numérico do status do HTTP da resposta.

Entrada de exemplo

```
200
```

Código de erro

O [Código de erro \(p. 618\)](#) do Amazon S3 ou “-”, se não tiver ocorrido nenhum erro.

Entrada de exemplo

```
NoSuchBucket
```

Bytes enviados

O número de bytes de resposta enviados excluindo a sobrecarga do protocolo HTTP ou “-”, se zero.

Entrada de exemplo

```
2662992
```

Tamanho do objeto

O tamanho total do objeto em questão.

Entrada de exemplo

```
3462992
```

Tempo total

O número de milissegundos em que a solicitação esteve em andamento da perspectiva do servidor.  
Esse valor é medido do momento do recebimento da solicitação até o momento em que o último byte

da resposta é enviado. As medidas feitas da perspectiva do cliente podem ser mais longas devido à latência da rede.

Entrada de exemplo

```
70
```

Tempo de retorno

O número de milissegundos que o Amazon S3 gastou processando a solicitação. Esse valor é medido do momento do recebimento do último byte da solicitação até o momento em que o primeiro byte da resposta foi enviado.

Entrada de exemplo

```
10
```

Indicador

O valor do cabeçalho do indicador HTTP, se presente. Os agentes do usuário HTTP (por exemplo, navegadores) normalmente definem esse cabeçalho como o URL da página de vinculação ou incorporação ao fazer uma solicitação.

Entrada de exemplo

```
"http://www.amazon.com/webservices"
```

Agente de usuário

O valor do cabeçalho do agente de usuário do HTTP.

Entrada de exemplo

```
"curl/7.15.1"
```

Id da versão

O ID da versão na solicitação ou “-”, se a operação não usar um parâmetro `versionId`.

Entrada de exemplo

```
3HL4kqtJvjVBH40Nrjfkd
```

## Informações personalizadas do log de acesso

É possível incluir informações personalizadas a serem armazenadas no registro de log de acesso para uma solicitação adicionando um parâmetro de string de consulta personalizado ao URL da solicitação. O Amazon S3 ignora parâmetros de string de consulta que começam com “x-”, mas os inclui no registro de log de acesso da solicitação, como parte do campo `Request-URI` do registro de log. Por exemplo, uma solicitação `GET` para "s3.amazonaws.com/mybucket/photos/2014/08/puppy.jpg?x-user=johndoe" funciona da mesma maneira que a mesma solicitação para "s3.amazonaws.com/mybucket/photos/2014/08/puppy.jpg", com a exceção de que a string "x-user=johndoe" é incluída no campo `Request-URI` para o registro do log associado. Essa funcionalidade está disponível apenas na interface REST.

## Considerações de programação para o formato do log de acesso ao servidor extensível

De vez em quando, podemos estender o formato do registro em log adicionando novos campos no final de cada linha. O código que analisa os logs de acesso de servidor devem ser escritos para tratar de campos que não comprehende.

### Registro em log adicional para operações de cópia

Uma operação de cópia envolve um `GET` e um `PUT`. Por esse motivo, registramos dois registros em log ao executar uma operação de cópia. A tabela anterior descreve os campos relacionados à parte `PUT` da operação. A lista a seguir descreve os campos no registro que se relacionam à parte `GET` da operação de cópia.

#### Proprietário do bucket

O ID canônico do usuário do bucket que armazena o objeto que está sendo copiado. O ID de usuário canônico é uma outra forma do ID da conta da AWS. Para obter mais informações sobre o ID de usuário canônico, consulte [Identificadores de conta da AWS](#). Para obter informações sobre como encontrar o ID de usuário canônico da conta, consulte [Encontrar o ID de usuário canônico da conta](#).

#### Entrada de exemplo

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

#### Bucket

O nome do bucket que armazena o objeto que está sendo copiado.

#### Entrada de exemplo

```
mybucket
```

#### Tempo

A hora em que a solicitação foi recebida. O formato que usa a terminologia `strftime()` é o seguinte:  
[ %d/%B/%Y:%H:%M:%S %z ]

#### Entrada de exemplo

```
[ 06/Feb/2014:00:00:38 +0000 ]
```

#### IP remoto

O endereço de internet aparente do solicitante. Os proxies e os firewalls intermediários podem obscurecer o endereço real da máquina que faz a solicitação.

#### Entrada de exemplo

```
192.0.2.3
```

#### Solicitante

O ID canônico do usuário do solicitante ou um – para solicitações não autenticadas. Se o solicitante for um usuário do IAM, esse campo retornará o nome do usuário do IAM do solicitante junto com a

conta raiz da AWS à qual o usuário do IAM pertence. Esse identificador é o mesmo usado para fins de controle de acesso.

Entrada de exemplo

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

ID da solicitação

Uma string gerada pelo Amazon S3 para identificar exclusivamente cada solicitação.

Entrada de exemplo

```
3E57427F33A59F07
```

Operação

A operação listada aqui é declarada como `SOAP.operation`,  
`REST.HTTP_method.resource_type`, `WEBSITE.HTTP_method.resource_type` ou  
`BATCH.DELETE.OBJECT`.

Entrada de exemplo

```
REST.COPY.OBJECT_GET
```

Chave

A “chave” do objeto que está sendo copiado ou “-”, se a operação não usar um parâmetro de chave.

Entrada de exemplo

```
/photos/2014/08/puppy.jpg
```

Request-URI

A parte de Request-URI da mensagem de solicitação HTTP.

Entrada de exemplo

```
"GET /mybucket/photos/2014/08/puppy.jpg?x-foo=bar"
```

Status HTTP

O código numérico do status do HTTP da parte GET da operação de cópia.

Entrada de exemplo

```
200
```

Código de erro

O [Código de erro \(p. 618\)](#) do Amazon S3 da parte GET da operação de cópia ou “-”, se nenhum erro tiver ocorrido.

Entrada de exemplo

NoSuchBucket

#### Bytes enviados

O número de bytes de resposta enviados excluindo a sobrecarga do protocolo HTTP ou “-”, se zero.

#### Entrada de exemplo

2662992

#### Tamanho do objeto

O tamanho total do objeto em questão.

#### Entrada de exemplo

3462992

#### Tempo total

O número de milissegundos em que a solicitação esteve em andamento da perspectiva do servidor. Esse valor é medido do momento do recebimento da solicitação até o momento em que o último byte da resposta é enviado. As medidas feitas da perspectiva do cliente podem ser mais longas devido à latência da rede.

#### Entrada de exemplo

70

#### Tempo de retorno

O número de milissegundos que o Amazon S3 gastou processando a solicitação. Esse valor é medido do momento do recebimento do último byte da solicitação até o momento em que o primeiro byte da resposta foi enviado.

#### Entrada de exemplo

10

#### Indicador

O valor do cabeçalho do indicador HTTP, se presente. Os agentes do usuário HTTP (por exemplo, navegadores) normalmente definem esse cabeçalho como o URL da página de vinculação ou incorporação ao fazer uma solicitação.

#### Entrada de exemplo

"http://www.amazon.com/webservices"

#### Agente de usuário

O valor do cabeçalho do agente de usuário do HTTP.

#### Entrada de exemplo

"curl/7.15.1"

#### Id da versão

O ID da versão do objeto que está sendo copiado ou "-", se o cabeçalho `x-amz-copy-source` não especificou um parâmetro `versionId` como parte da origem da cópia.

#### Entrada de exemplo

```
3HL4kqtJvjVBH40Nrjfkd
```

## Excluir arquivos de log do Amazon S3

Um bucket do S3 com registro em log de acesso ao servidor habilitado pode acumular vários objetos de log ao longo do tempo. O aplicativo pode precisar desses logs de acesso durante um período específico após sua criação. Depois disso, você pode excluí-los. Use a configuração do ciclo de vida do Amazon S3 para definir regras de modo que o Amazon S3 organize esses objetos automaticamente em filas para exclusão no fim do seu ciclo de vida.

Você pode definir uma configuração de ciclo de vida para um subconjunto de objetos no seu bucket do S3 usando um prefixo compartilhado (ou seja, objetos com nomes que começam com uma string em comum). Se você especificou um prefixo na sua configuração do registro em log de acesso ao servidor, defina uma regra de configuração do ciclo de vida para excluir objetos de log com esse prefixo. Por exemplo, se os seus objetos de log têm o prefixo `logs/`, você pode definir uma regra de configuração do ciclo de vida para excluir todos os objetos no bucket com o prefixo `/logs` após um período especificado. Para obter mais informações sobre a configuração do ciclo de vida, consulte [Gerenciamento do ciclo de vida de objetos \(p. 122\)](#).

## Mais informações

[Registro em log de acesso ao servidor Amazon S3 \(p. 625\)](#)

# Usar os AWS SDKs, a CLI e os Explorers

Você também pode usar os AWS SDKs ao desenvolver aplicativos com o Amazon S3. Os AWS SDKs simplificam as tarefas de programação integrando a API REST subjacente. Os SDKs do AWS Mobile e a biblioteca Amplify JavaScript da AWS também estão disponíveis para a compilação de aplicativos web e aplicativos para dispositivos móveis conectados ao usar a AWS.

Esta seção fornece uma visão geral dos AWS SDKs para desenvolver aplicativos do Amazon S3. Esta seção também descreve como você pode testar os exemplos de código do SDK da AWS fornecidos neste guia.

## Tópicos

- [Especificar a versão da assinatura na autenticação de solicitações \(p. 640\)](#)
- [Configurar a CLI da AWS \(p. 645\)](#)
- [Usar o AWS SDK for Java \(p. 646\)](#)
- [Usar o AWS SDK para .NET \(p. 647\)](#)
- [Usar o AWS SDK para PHP e executar exemplos do PHP \(p. 649\)](#)
- [Usar o AWS SDK para Ruby - versão 3 \(p. 650\)](#)
- [Usar o AWS SDK for Python \(Boto\) \(p. 651\)](#)
- [Usar os AWS Mobile SDKs para iOS e Android \(p. 651\)](#)
- [Usar a biblioteca JavaScript do AWS Amplify \(p. 651\)](#)

Além dos AWS SDKs, os AWS Explorers estão disponíveis para Visual Studio e Eclipse para Java IDE. Nesse caso, os SDKs e os Explorers estão disponíveis em um pacote como toolkits da AWS.

Você também pode usar a interface de linha de comando (AWS CLI) da AWS para gerenciar buckets e objetos do Amazon S3.

## AWS Toolkit for Eclipse

O AWS Toolkit for Eclipse inclui o AWS SDK for Java e o AWS Explorer para Eclipse. O AWS Explorer para Eclipse é um plug-in de código aberto para o Eclipse para Java IDE que facilita o desenvolvimento, a depuração e a implantação de aplicativos Java para os desenvolvedores que usam a AWS. A GUI fácil de usar permite acessar e administrar a infraestrutura da AWS incluindo o Amazon S3. Você pode realizar operações comuns, como gerenciar os buckets e objetos, além de definir políticas do IAM ao mesmo tempo em que desenvolve aplicativos, tudo no contexto do Eclipse para Java IDE. Para instruções de configuração, consulte [Configurar o toolkit](#). Para obter exemplos do Explorer, consulte [Como acessar o AWS Explorer](#).

## AWS Toolkit for Visual Studio

O AWS Explorer para Visual Studio é uma extensão do Microsoft Visual Studio que facilita o desenvolvimento, a depuração e a implantação de aplicativos .NET para os desenvolvedores que usam a Amazon Web Services. A GUI fácil de usar permite acessar e administrar a infraestrutura da AWS incluindo o Amazon S3. Você pode executar operações comuns, como gerenciar buckets e objetos ou definir políticas do IAM ao desenvolver aplicativos, tudo no contexto do Visual Studio. Para obter instruções de configuração, visite [Configurar o AWS Toolkit for Visual Studio](#). Para obter exemplos sobre como usar o Amazon S3 utilizando o Explorer, consulte [Usar o Amazon S3 no AWS Explorer](#).

## SDKs da AWS

Você pode fazer download somente dos SDKs. Para obter informações sobre download de bibliotecas de SDKs, consulte [Bibliotecas de códigos de exemplo](#).

#### AWS CLI

A AWS CLI é uma ferramenta unificada para gerenciar os serviços da AWS, incluindo o Amazon S3. Para obter informações sobre como fazer download da AWS CLI, consulte [AWS Command Line Interface](#).

## Especificar a versão da assinatura na autenticação de solicitações

O Amazon S3 oferece suporte apenas ao AWS Signature versão 4 na maioria das regiões da AWS. Em algumas das regiões mais antigas da AWS, o Amazon S3 oferece suporte às versões 2 e 4 do Signature. No entanto, o Signature versão 2 será descontinuado. Haverá suporte a ele somente até dia 24 de junho de 2019. Para obter mais informações sobre o fim do suporte ao Signature versão 2, consulte [Substituição do AWS Signature versão 2 para o Amazon S3 \(p. 641\)](#).

Para obter uma lista de todas as regiões do Amazon S3 e saber a quais versões do Signature elas oferecem suporte, consulte [Regiões e endpoints](#) na Referência geral da AWS.

Para todas as regiões da AWS, por padrão, os SDKs da AWS usam o Signature versão 4 para autenticar solicitações. Ao usar os SDKs da AWS liberados antes de maio de 2016, talvez seja necessário solicitar o Signature versão 4 conforme mostrado na tabela a seguir:

SDK	Solicitar o Signature versão 4 para autenticação de solicitações
CLI da AWS	<p>Para o perfil padrão, execute o comando a seguir:</p> <pre>\$ aws configure set default.s3.signature_version s3v4</pre> <p>Para um perfil personalizado, execute o comando a seguir:</p> <pre>\$ aws configure set profile.your_profile_name.s3.signature_version s3v4</pre>
SDK do Java	<p>Adicione o seguinte ao código:</p> <pre>System.setProperty(SDKGlobalConfiguration.ENABLE_S3_SIGV4_SYSTEM_PROPERTY, "true");</pre> <p>Ou, na linha de comando, especifique o seguinte:</p> <pre>-Dcom.amazonaws.services.s3.enableV4</pre>
SDK do JavaScript	<p>Defina o parâmetro <code>signatureVersion</code> como <code>v4</code> ao criar o cliente:</p> <pre>var s3 = new AWS.S3({signatureVersion: 'v4'});</pre>
SDK do PHP	<p>Defina o parâmetro <code>signature</code> como <code>v4</code> ao criar o cliente de serviço do Amazon S3:</p> <pre>&lt;?php</pre>

SDK	Solicitar o Signature versão 4 para autenticação de solicitações
	<pre>\$s3 = new S3Client(['signature' =&gt; 'v4']);</pre>
SDK do Python-Boto	Especifique o seguinte no arquivo de configuração boto padrão:  <pre>[s3] use-sigv4 = True</pre>
SDK do Ruby	SDK do Ruby - versão 1: defina o parâmetro <code>:s3_signature_version</code> como <code>:v4</code> ao criar o cliente:  <pre>s3 = AWS::S3::Client.new(:s3_signature_version =&gt; :v4)</pre> SDK do Ruby - versão 3: defina o parâmetro <code>signature_version</code> como <code>v4</code> ao criar o cliente:  <pre>s3 = Aws::S3::Client.new(signature_version: 'v4')</pre>
SDK do .NET	Adicione o código a seguir antes de criar o cliente do Amazon S3:  <pre>AWSConfig.S3.UseSignatureVersion4 = true;</pre> Ou adicione o seguinte ao arquivo config:  <pre>&lt;appSettings&gt;     &lt;add key="AWS.S3.UseSignatureVersion4" value="true" /&gt; &lt;/appSettings&gt;</pre>

## Substituição do AWS Signature versão 2 para o Amazon S3

O Signature versão 2 será descontinuado para o Amazon S3. Haverá suporte a ele somente até dia 24 de junho de 2019. Depois de 24 de junho de 2019, o Amazon S3 só aceitará solicitações de API assinadas com o Signature versão 4.

Esta seção apresenta respostas às perguntas mais comuns relacionadas ao fim do suporte ao Signature versão 2.

O que é o Signature versão 2/4 e o que “assinar solicitações” quer dizer?

O processo de assinatura com as versões 2 ou 4 do Signature é utilizado para autenticar solicitações de API do Amazon S3. Ao assinar solicitações, o Amazon S3 pode identificar quem está enviando a solicitação e as protege contra agentes mal-intencionados.

Para obter mais informações sobre como assinar solicitações da AWS, consulte [Como assinar solicitações de API da AWS](#) na AWS General Reference.

Qual atualização está sendo feita?

No momento, oferecemos suporte a solicitações de API do Amazon S3 assinadas com as versões 2 e 4 do Signature. Depois de 24 de junho de 2019, o Amazon S3 só aceitará solicitações assinadas com o Signature versão 4.

Para mais informações sobre a assinatura de solicitações da AWS, consulte [Mudanças do Signature versão 4](#) na AWS General Reference.

Por que esta atualização está sendo feita?

O Signature versão 4 oferece maior segurança porque usa uma chave de assinatura em vez de sua chave de acesso secreta. No momento, o Signature versão 4 é compatível com todas as regiões da AWS. O Signature versão 2 só é compatível com as regiões lançadas antes de janeiro de 2014. Essa atualização nos permite oferecer uma experiência mais consistente em todas as regiões.

Como posso ter certeza de que estou usando o Signature versão 4 e quais atualizações são necessárias?

Normalmente, a versão utilizada para assinar suas solicitações é definida pela ferramenta ou pelo SDK no lado do cliente. Por padrão, as versões mais recentes dos SDKs da AWS usam o Signature versão 4. Para software de terceiro, entre em contato com a equipe de suporte dele para confirmar a versão necessária. Para enviar chamadas REST diretas para o Amazon S3, você deverá modificar seu aplicativo para usar o processo do Signature versão 4.

Para obter informações sobre qual versão dos SDKs da AWS deve ser usada depois da transição para o Signature versão 4, consulte [Migração do Signature versão 2 para o Signature versão 4 \(p. 642\)](#).

Para obter informações sobre como usar a versão 4 da assinatura com a API REST do Amazon S3, consulte [Autenticar solicitações \(AWS Signature versão 4\)](#) no Amazon Simple Storage Service API Reference.

O que acontecerá se eu não atualizar?

Depois de 24 de junho de 2019, as solicitações assinadas com o Signature versão 2 apresentarão falha na autenticação com o Amazon S3. Os solicitantes verão erros com a mensagem de que a solicitação deve ser assinada com o Signature versão 4.

Devo fazer alterações mesmo se estiver usando um pre-signed URL que exige a assinatura por mais de sete dias?

Se você estiver usando um pre-signed URL que exige a assinatura por mais de sete dias, não é necessário fazer nada. Você pode continuar usando o AWS Signature versão 2 para assinar e autenticar pre-signed URLs. Vamos fazer um acompanhamento e fornecer mais detalhes sobre como migrar para o Signature versão 4 para pre-signed URL.

## Mais informações

- Para obter mais informações sobre como usar o Signature versão 4, consulte [Assinatura de solicitações de API da AWS](#).
- Consulte a lista de alterações do Signature versão 2 para o Signature versão 4 em [Alterações no Signature versão 4](#).
- Veja a publicação [AWS Signature Version 4 to replace AWS Signature Version 2 for signingAmazon S3 API requests](#) nos fóruns da AWS.
- Em caso de dúvidas, entre em contato com o [AWS Support](#).

## Migração do Signature versão 2 para o Signature versão 4

Se você está usando o Signature versão 2 para autenticar solicitações de API do Amazon S3, precisará migrar para o Signature versão 4. Não haverá mais suporte para o Signature versão 2, como descrito em [Substituição do AWS Signature versão 2 para o Amazon S3 \(p. 641\)](#).

Para obter informações sobre como usar a versão 4 da assinatura com a API REST do Amazon S3, consulte [Autenticar solicitações \(AWS Signature versão 4\)](#) no Amazon Simple Storage Service API Reference.

A tabela a seguir contém os SDKs que exigem a utilização do Signature versão 4 (SigV4).

Se você utilizar pre-signed URLs com os SDKs AWS Java, JavaScript (Node.js) ou Python (Boto/CLI), deverá definir a região da AWS correta e o Signature versão 4 na configuração do cliente. Para obter mais informações sobre como definir o SigV4 na configuração do cliente, consulte [Especificificar a versão da assinatura na autenticação de solicitações \(p. 640\)](#).

Se você usa este SDK/ produto	Atualize para esta versão do SDK	Alteração de código necessária para o cliente usar o Sigv4?	Link para documentação do SDK
AWS SDK for Java v1	Atualizar para Java 1.11.x ou v2 no quarto trimestre de 2018.	Sim	<a href="#">Especificar a versão da assinatura na autenticação de solicitações (p. 640)</a>
AWS SDK for Java v2 (pré-visualização)	Não é necessário atualizar os SDKs.	Não	<a href="#">AWS SDK for Java</a>
AWS SDK para .NET v1	Atualize para o 3.1.10 ou versão superior.	Sim	<a href="#">AWS SDK para .NET</a>
AWS SDK para .NET v2	Atualize para o 3.1.10 ou versão superior.	Não	<a href="#">AWS SDK para .NET v2</a>
AWS SDK para .NET v3	Não é necessário atualizar os SDKs.	Sim	<a href="#">AWS SDK para .NET v3</a>
AWS SDK for JavaScript v1	Nenhuma outra ação é necessária no momento. Atualize para a versão principal V3 no terceiro trimestre de 2019.	Sim	<a href="#">AWS SDK for JavaScript</a>
AWS SDK for JavaScript v2	Atualize para o 2.68.0 ou versão superior.	Sim	<a href="#">AWS SDK for JavaScript</a>
AWS SDK for JavaScript v3	Nenhuma outra ação é	Não	<a href="#">AWS SDK for JavaScript</a>

Se você usa este SDK/ produto	Atualize para esta versão do SDK	Alteração de código necessária para o cliente usar o Sigv4?	Link para documentação do SDK
	necessária no momento. Atualize para a versão principal V3 no terceiro trimestre de 2019.		
AWS SDK para PHP v1	Atualize para a versão principal V3.	Sim	<a href="#">AWS SDK para PHP</a>
AWS SDK para PHP v2	Atualize para a versão principal V3.	Sim	<a href="#">AWS SDK para PHP</a>
AWS SDK para PHP v3	Não é necessário atualizar os SDKs.	Não	<a href="#">AWS SDK para PHP</a>
Boto2	Atualize para Boto2 v2.49.0.	Sim	<a href="#">Atualização do Boto 2</a>
Boto3	Atualize para 1.5.71 (BotoCore), 1.4.6 (Boto3).	Sim	<a href="#">Boto 3 - SDK da AWS para Python</a>
AWS CLI	Atualize para 1.11.108.	Sim	<a href="#">Interface de linha de comando da AWS</a>
AWS CLI v2 (pré-visualização)	Não é necessário atualizar os SDKs.	Não	<a href="#">Interface da linha de comando da AWS, versão 2</a>
AWS SDK para Ruby v1	Atualize para Ruby V3.	Sim	<a href="#">Ruby V3 para AWS</a>
AWS SDK para Ruby v2	Atualize para Ruby V3.	Sim	<a href="#">Ruby V3 para AWS</a>
AWS SDK para Ruby v3	Não é necessário atualizar os SDKs.	Não	<a href="#">Ruby V3 para AWS</a>
Go	Não é necessário atualizar os SDKs.	Não	<a href="#">AWS SDK para Go</a>

Se você usa este SDK/ produto	Atualize para esta versão do SDK	Alteração de código necessária para o cliente usar o Sigv4?	Link para documentação do SDK
C++	Não é necessário atualizar os SDKs.	Não	<a href="#">AWS SDK para C++</a>

AWS Tools para Windows PowerShell ou AWS Tools for PowerShell Core

Se você estiver usando versões de módulo anteriores à 3.3.99, deverá atualizar para a 3.3.99.

Para obter informações sobre a versão, use o cmdlet do `Get-Module`:

```
Get-Module -Name AWSPowershell
Get-Module -Name AWSPowershell.NetCore
```

Para atualizar para a versão 3.3.99, use o cmdlet do `Update-Module`:

```
Update-Module -Name AWSPowershell
Update-Module -Name AWSPowershell.NetCore
```

É possível enviar o tráfego do Signature versão 2 para pre-signed URLs válidas por mais de sete dias.

## Configurar a CLI da AWS

Siga as etapas para fazer download e configurar a interface de linha de comando da AWS (AWS CLI).

### Note

Os serviços na AWS, como o Amazon S3, exigem que credenciais sejam fornecidas ao acessá-los. Esse serviço pode determinar se você tem permissões para acessar os respectivos recursos próprios. O console requer sua senha. Você pode criar chaves de acesso para a conta da AWS a fim de acessar a AWS CLI ou a API. No entanto, não recomendamos que você acesse a AWS usando as credenciais da conta da AWS. Em vez disso, recomendamos o uso de AWS Identity and Access Management (IAM). Crie um usuário do IAM, adicione o usuário a um grupo do IAM com permissões administrativas e, em seguida, conceda permissões administrativas ao usuário do IAM criado. Em seguida, você pode acessar a AWS usando uma URL especial e as credenciais desse usuário do IAM. Para obter instruções, acesse [Criar o primeiro usuário do IAM e grupo de administradores](#) no Guia do usuário do IAM.

Para configurar a AWS CLI

1. Faça download e configure a AWS CLI. Para obter instruções, consulte os seguintes tópicos no Guia do usuário da interface de linha de comando da AWS:
  - [Configurar com a interface de linha de comando da AWS](#)
  - [Configurar a interface de linha de comando da AWS](#)

2. Adicione um perfil nomeado para o usuário administrador no arquivo config da AWS CLI. Você pode usar esse perfil ao executar os comandos da AWS CLI.

```
[adminuser]
aws_access_key_id = adminuser access key ID
aws_secret_access_key = adminuser secret access key
region = aws-region
```

Para obter uma lista das regiões da AWS disponíveis, consulte [Regiões e endpoints](#) no AWS General Reference.

3. Verifique a configuração digitando os comandos a seguir no prompt de comando.

- Teste o comando `help` para verificar se a AWS CLI está instalada no computador:

```
aws help
```

- Teste um comando do S3 para verificar se o usuário consegue acessar o Amazon S3. Esse comando lista os buckets de sua conta. A AWS CLI usa as credenciais `adminuser` para autenticar a solicitação.

```
aws s3 ls --profile adminuser
```

## Usar o AWS SDK for Java

O AWS SDK for Java fornece uma API para as operações de bucket e de objeto do Amazon S3. Para operações de objeto, além de fornecer a API para fazer upload de objetos em uma única operação, o SDK fornece a API para fazer upload de grandes objetos em partes. Para obter mais informações, consulte [Upload de objetos usando a API de multipart upload \(p. 181\)](#).

### Tópicos

- [A organização da API Java \(p. 647\)](#)
- [Testar os exemplos de código Java do Amazon S3 \(p. 647\)](#)

O AWS SDK for Java oferece a opção de usar uma API de alto ou baixo nível.

### API de baixo nível

As APIs de baixo nível correspondem às operações subjacentes do Amazon S3 REST, como operações de criação, atualização e exclusão que se aplicam a buckets e objetos. Quando você faz upload de objetos grandes usando a API de multipart upload de baixo nível, ela proporciona mais controle. Por exemplo, ela permite pausar e retomar multipart uploads, variar tamanhos de parte durante o upload ou começar uploads quando você não sabe o tamanho dos dados com antecedência. Se você não tiver esses requisitos, use a API de alto nível para fazer upload de objetos.

### API de alto nível

Para fazer upload de objetos, o SDK fornece um nível superior de abstração fornecendo a classe `TransferManager`. A API de alto nível é uma API mais simples, na qual em apenas algumas linhas de código, você pode fazer upload de arquivos e fluxos no Amazon S3. Você deve usar essa API para fazer upload dos dados a menos que você precise controlar o upload conforme descrito na seção API de baixo nível anterior.

Para dados menores, a API `TransferManager` faz upload dos dados em uma única operação. No entanto, o `TransferManager` alterna para o uso da API de multipart upload quando os dados atingem um determinado limite. Quando possível, o `TransferManager` usa vários threads para fazer upload das

partes simultaneamente. Se houver falha no upload de uma parte, a API repetirá o upload da parte até três vezes. Contudo, essas são opções configuráveis usando a classe `TransferManagerConfiguration`.

#### Note

Quando você está usando um streaming na fonte dos dados, a classe `TransferManager` não faz uploads simultâneos.

## A organização da API Java

Os seguintes pacotes no AWS SDK for Java fornecem a API:

- `com.amazonaws.services.s3`— fornece as APIs para a criação de clientes do Amazon S3 e o trabalho com buckets e objetos. Por exemplo, ele possibilita a você criar buckets, fazer upload de objetos, obter objetos, excluir objetos e listar chaves.
- `com.amazonaws.services.s3.transfer`— fornece as operações dos dados da API de alto nível.

Essa API de alto nível é projetada para simplificar a transferência de objetos do e para o Amazon S3. Ela inclui a classe `TransferManager`, que fornece métodos assíncronos para trabalhar com, consultar e manipular transferências. Também inclui a classe `TransferManagerConfiguration` que você pode usar para configurar o tamanho mínimo das partes para upload e o limite em bytes de quando usar multipart uploads.

- `com.amazonaws.services.s3.model`— fornece as classes da API de baixo nível para criar solicitações e respostas a processos. Por exemplo, inclui a classe `GetObjectRequest` para descrever sua solicitação para obter objetos, a classe `ListObjectsRequest` para descrever suas solicitações para listar chaves, e a classe `InitiateMultipartUploadRequest` para criar multipart uploads.

Para obter mais informações sobre o recurso de API do AWS SDK for Java, consulte [AWS SDK for Java API Reference](#).

## Testar os exemplos de código Java do Amazon S3

Os exemplos de Java neste guia são compatíveis com o AWS SDK for Java versão 1.11.321. Para obter instruções sobre como configurar e executar exemplos de código, consulte [Conceitos básicos do AWS SDK for Java](#) no AWS SDK for Java Developer Guide.

## Usar o AWS SDK para .NET

O AWS SDK para .NET fornece a API para as operações de bucket e de objeto do Amazon S3. Para operações de objetos, além de prover a API para fazer upload de objetos em uma única operação, o SDK provê a API para fazer upload de grandes objetos em partes (consulte [Upload de objetos usando a API de multipart upload \(p. 181\)](#)).

#### Tópicos

- [A organização da API .NET \(p. 648\)](#)
- [Executar os exemplos de código .NET do Amazon S3 \(p. 648\)](#)

O AWS SDK para .NET oferece a opção de usar uma API de alto ou baixo nível.

#### API de baixo nível

As APIs de baixo nível correspondem às operações subjacentes do Amazon S3 REST incluindo operações de criação, atualização e exclusão que se aplicam a buckets e objetos. Quando você faz upload de objetos grandes usando a API de multipart upload de baixo nível (consulte [Upload de objetos usando a API de multipart upload \(p. 181\)](#)).

[multipart upload \(p. 181\)](#)), ela proporciona mais controle. Por exemplo, ela permite pausar e retomar multipart uploads, variar tamanhos de parte durante o upload ou começar uploads quando você não sabe o tamanho dos dados com antecedência. Se você não tiver essas necessidades, use a API de alto nível para fazer upload de objetos.

#### API de alto nível

Para fazer upload de objetos, o SDK fornece um nível superior de abstração fornecendo a classe `TransferUtility`. A API de alto nível é uma API mais simples, na qual em apenas algumas linhas de código, você pode fazer upload de arquivos e fluxos no Amazon S3. Você deve usar essa API para fazer upload dos dados a menos que você precise controlar o upload conforme descrito na seção API de baixo nível anterior.

Para dados menores, a API `TransferUtility` faz upload dos dados em uma única operação. No entanto, o `TransferUtility` alterna para o uso da API de multipart upload quando os dados atingem um determinado limite. Por padrão, ele usa vários threads para fazer upload das partes simultaneamente. Se houver falha no upload de uma parte, a API repetirá o upload da parte até três vezes. Contudo, essas são opções configuráveis.

#### Note

Quando você está usando um streaming na fonte dos dados, a classe `TransferUtility` não faz uploads simultâneos.

## A organização da API .NET

Ao escrever aplicativos do Amazon S3 usando o AWS SDK para .NET, você usa o `AWSSDK.dll`. Os seguintes namespaces neste assembly fornecem a API de multipart upload:

- `Amazon.S3.Transfer`— fornece a API de alto nível para carregar os dados em partes.  
Inclui a classe `TransferUtility` que permite especificar um arquivo, um diretório ou um fluxo para upload dos dados. Também inclui as classes `TransferUtilityUploadRequest` e `TransferUtilityUploadDirectoryRequest` para definir configurações avançadas, como o número de threads simultâneos, o tamanho da parte, os metadados do objeto, a classe de armazenamento (STANDARD, REDUCED\_REDUNDANCY) e a Access Control List (ACL – Lista de controle de acesso) do objeto.
- `Amazon.S3`— fornece a implementação das APIs de baixo nível.  
Fornece métodos que correspondem à API multipart upload do Amazon S3 REST (consulte [Usar a API REST para multipart upload \(p. 214\)](#)).
- `Amazon.S3.Model`— fornece as classes da API de baixo nível para criar solicitações e respostas a processos. Por exemplo, fornece as classes `InitiateMultipartUploadRequest` e `InitiateMultipartUploadResponse` que você pode usar ao iniciar um multipart upload, e as classes `UploadPartRequest` e `UploadPartResponse` ao fazer o upload das partes.
- `Amazon.S3.Encryption`— fornece `AmazonS3EncryptionClient`.
- `Amazon.S3.Util`— fornece várias classes de utilitários, tais como `AmazonS3Util` e `BucketRegionDetector`.

Para obter mais informações sobre a API do AWS SDK para .NET, consulte [Versão 3 da Referência API de AWS SDK para .NET](#).

## Executar os exemplos de código .NET do Amazon S3

Os exemplos de código .NET neste guia são compatíveis com o AWS SDK para .NET versão 3.0. Para obter informações sobre como configurar e executar exemplos de código, consulte [Conceitos básicos do AWS SDK para .NET](#) no AWS SDK for .NET Developer Guide.

# Usar o AWS SDK para PHP e executar exemplos do PHP

O AWS SDK para PHP fornece acesso à API para as operações de bucket e de objeto do Amazon S3. O SDK fornece a opção de usar a API de baixo nível do serviço ou abstrações de alto nível.

O SDK está disponível no [AWS SDK para PHP](#), que também tem instruções para instalar e começar a usar o SDK.

A configuração para usar o AWS SDK para PHP depende do ambiente e de como você deseja executar seu aplicativo. Para configurar seu ambiente para executar os exemplos desta documentação, consulte [AWS SDK para PHP Getting Started Guide](#).

## Tópicos

- [Níveis do AWS SDK para PHP \(p. 649\)](#)
- [Executar exemplos do PHP \(p. 649\)](#)
- [Recursos relacionados \(p. 650\)](#)

## Níveis do AWS SDK para PHP

O AWS SDK para PHP oferece a opção de usar uma API de alto ou baixo nível.

### API de baixo nível

As APIs de baixo nível correspondem às operações subjacentes do Amazon S3 REST incluindo operações de criação, atualização e exclusão em buckets e objetos. As APIs de baixo nível fornecem maior controle sobre essas operações. Por exemplo, você pode colocar suas solicitações em lotes e executá-las em paralelo. Ou, ao usar a API multipart upload, você pode gerenciar as partes de objetos individualmente. Observe que essas chamadas da API de baixo nível retornam um resultado que inclui todos os detalhes da resposta do Amazon S3. Para obter mais informações sobre a API multipart upload, consulte [Upload de objetos usando a API de multipart upload \(p. 181\)](#).

### Abstrações de alto nível

As abstrações de alto nível têm o objetivo de simplificar casos de uso comuns. Por exemplo, para fazer upload dos objetos grandes usando a API de baixo nível, você deve primeiro chamar `Aws\S3\S3Client::createMultipartUpload()`, em seguida, chamar o método `Aws\S3\S3Client::uploadPart()` para fazer upload das partes dos objetos e, em seguida, chamar o método `Aws\S3\S3Client::completeMultipartUpload()` para concluir o upload. Você pode usar o objeto `Aws\S3\MultipartUploader` de alto nível que simplifica a criação de um multipart upload em vez disso.

Outro exemplo é quando se enumera objetos em um bucket no qual você pode usar o recurso de iteradores do AWS SDK para PHP para retornar todas as chaves de objeto, independentemente de quantos objetos foram armazenados no bucket. Se você usar a API de baixo nível, a resposta retornará, no máximo, 1.000 chaves. Se o bucket contiver mais de 1.000 objetos, o resultado ficará truncado e você terá que gerenciar a resposta e verificar o truncamento.

## Executar exemplos do PHP

Por configurar e usar exemplos do Amazon S3 para a versão 3 do AWS SDK para PHP, consulte [Instalação](#) no AWS SDK para PHP Developer Guide.

## Recursos relacionados

- [AWS SDK para PHP para Amazon S3](#)
- [API do AWS SDK para PHP para Amazon S3](#)

# Usar o AWS SDK para Ruby - versão 3

O AWS SDK para Ruby fornece uma API para operações de bucket e objeto do Amazon S3. Para operações de objeto, você pode usar a API para fazer upload de objetos em uma única operação ou fazer upload de objetos grandes em partes (consulte [Usar o AWS SDK para Ruby para multipart upload \(p. 214\)](#)). Contudo, a API para um único upload de operação também pode aceitar objetos grandes e, em segundo plano, gerenciar o upload em partes para você, reduzindo a quantidade de script que precisa escrever.

## A organização da API Ruby

Ao criar aplicativos do Amazon S3 usando o AWS SDK para Ruby, você deve instalar o SDK para Ruby gem. Para obter mais informações, consulte [AWS SDK para Ruby - versão 3](#). Depois de instalado, você pode acessar a API, incluindo as seguintes classes de chaves:

- Aws::S3::Resource — representa a interface para Amazon S3 para o SDK do Ruby e fornece métodos para criar e enumerar buckets.

A classe S3 fornece o método da instância `#buckets` para acessar buckets existentes ou criar novos.

- Aws::S3::Bucket — representa um bucket do Amazon S3.

A classe Bucket fornece os métodos `#object(key)` e `#objects` para acessar os objetos em um bucket, bem como métodos para excluir um bucket e retornar informações sobre um bucket, como a política do bucket.

- Aws::S3::Object — representa um objeto do Amazon S3 identificado por sua chave.

A classe Object fornece métodos para obter e definir as propriedades de um objeto, especificando a classe storage para armazenar objetos, e definindo permissões de objetos usando listas de controle de acesso. A classe Object também tem métodos para exclusão, upload e cópia de objetos. Ao carregar objetos em partes, essa classe fornece opções para especificar a ordem das partes carregadas e o tamanho das partes.

Para obter mais informações sobre o AWS SDK para a API do Ruby, visite [AWS SDK para Ruby - versão 2](#).

## Testar os exemplos de script do Ruby

A maneira mais fácil de começar a usar os exemplos de script do Ruby é instalar o AWS SDK para Ruby gem. Para obter informações sobre como instalar ou atualizar a gem mais recente, visite [AWS SDK para Ruby - versão 3](#). As tarefas a seguir orientam você na criação e nos testes dos exemplos de script do Ruby pressupondo que você instalou o AWS SDK para Ruby.

Processo geral de criação e testes dos exemplos de script do Ruby

1	Para acessar a AWS, você deve fornecer um conjunto de credenciais para seu aplicativo SDK para Ruby. Para obter mais informações, consulte <a href="#">Configurar o AWS SDK para Ruby</a> .
---	---

2	Crie um script SDK para Ruby novo e adicione as seguintes linhas à parte superior do script.  <pre>#!/usr/bin/env ruby  require 'rubygems' require 'aws-sdk-s3'</pre>
	A primeira linha é a diretiva do intérprete e as duas instruções <code>require</code> importam duas gems necessárias no script.
3	Copie o código da seção que você está lendo no script.
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 script. Verifique as alterações nos buckets e nos objetos usando o Console de gerenciamento da AWS. Para obter mais informações sobre o Console de gerenciamento da AWS, visite <a href="https://aws.amazon.com/console/">https://aws.amazon.com/console/</a> .

### Exemplos do Ruby

Os links a seguir contêm exemplos para ajudar você a começar a usar o SDK para Ruby - versão 3:

- [Uso do AWS SDK para Ruby Versão 3 \(p. 63\)](#)
- [Faça upload de objetos usando o AWS SDK para Ruby \(p. 179\)](#)

## Usar o AWS SDK for Python (Boto)

O Boto é um pacote do Python que fornece interfaces à AWS incluindo o Amazon S3. Para obter mais informações sobre o Boto, visite o [AWS SDK for Python \(Boto\)](#). O link de começar a usar nesta página fornece instruções passo a passo para começar.

## Usar os AWS Mobile SDKs para iOS e Android

Você pode usar os AWS Mobile SDKs para Android e iOS com o [AWS Mobile Hub](#), para integrar de maneira rápida e fácil back-ends de nuvem robusta a aplicativos para dispositivos móveis existentes. Você pode configurar e usar recursos como login de usuário, bancos de dados, notificações por push e muito mais, sem ser um especialista na AWS.

Os AWS Mobile SDKs oferecem acesso fácil ao Amazon S3 e a muitos outros serviços da AWS. Para começar a usar os AWS Mobile SDKs, consulte [Conceitos básicos dos AWS Mobile SDKs](#).

## Mais informações

[Usar a biblioteca JavaScript do AWS Amplify \(p. 651\)](#)

## Usar a biblioteca JavaScript do AWS Amplify

AWS Amplify é uma biblioteca de JavaScript de código aberto para desenvolvedores para web e dispositivos móveis que compilam aplicativos compatíveis com a nuvem. O AWS Amplify fornece

componentes de interface do usuário personalizáveis e uma interface declarativa para trabalhar com um bucket do S3, além de outras categorias de alto nível para serviços da AWS.

Para começar a usar a biblioteca JavaScript do AWS Amplify, escolha um dos links a seguir:

- [Conceitos básicos da biblioteca do AWS Amplify para a web](#)
- [Conceitos básicos da biblioteca do AWS Amplify para React Native](#)

Para obter mais informações sobre o AWS Amplify, consulte [AWS Amplify](#) no [GitHub](#).

## Mais informações

[Usar os AWS Mobile SDKs para iOS e Android \(p. 651\)](#)

# Apêndices

Esse apêndice do Guia do desenvolvedor do Amazon Simple Storage Service inclui as seções a seguir.

## Tópicos

- [Apêndice A: uso da API SOAP \(p. 653\)](#)
- [Apêndice B: autenticação de solicitações \(versão 2 do AWS Signature\) \(p. 656\)](#)

## Apêndice A: uso da API SOAP

### Note

O suporte de SOAP via HTTP está obsoleto, mas continua disponível via HTTPS. Novos recursos do Amazon S3 não serão compatíveis com o SOAP. Recomendamos que você use o REST API ou os SDKs da AWS.

Esta seção contém informações específicas da API SOAP do Amazon S3.

### Note

As solicitações SOAP, autenticadas e anônimas, devem ser enviadas para o Amazon S3 usando SSL. O Amazon S3 retorna um erro quando você envia uma solicitação SOAP via HTTP.

## Elementos comuns da API SOAP

### Note

O suporte de SOAP via HTTP está obsoleto, mas continua disponível via HTTPS. Novos recursos do Amazon S3 não serão compatíveis com o SOAP. Recomendamos que você use o REST API ou os SDKs da AWS.

Você pode interagir com o Amazon S3 usando SOAP 1.1 sobre HTTP. O Amazon S3 WSDL, que descreve a API do Amazon S3 em uma forma legível pela máquina, está disponível em: <http://doc.s3.amazonaws.com/2006-03-01/AmazonS3.wsdl>. O esquema do Amazon S3 está disponível em <http://doc.s3.amazonaws.com/2006-03-01/AmazonS3.xsd>.

A maioria de usuários interagirá com o Amazon S3 usando um toolkit SOAP personalizado para sua linguagem e ambiente de desenvolvimento. Diferentes toolkits expõem a API do Amazon S3 de diferentes maneiras. Consulte a documentação específica do toolkit para entender como usá-la. Esta seção ilustra as operações SOAP do Amazon S3 de um modo independente de toolkit exibindo as solicitações XML e as respostas como elas aparecem "na rede".

## Elementos comuns

Você pode incluir os seguintes elementos relacionados a autorização com qualquer solicitação SOAP:

- **AWSAccessKeyId**: O ID de chave de acesso da AWS do solicitante
- **Timestamp**: A hora atual do seu sistema

- **Signature:** A assinatura da solicitação

## Como autenticar solicitações SOAP

### Note

O suporte de SOAP via HTTP está obsoleto, mas continua disponível via HTTPS. Novos recursos do Amazon S3 não serão compatíveis com o SOAP. Recomendamos que você use o REST API ou os SDKs da AWS.

Cada solicitação não anônima deve conter informações de autenticação para estabelecer a identidade do principal que faz a solicitação. Em SOAP, as informações de autenticação são colocadas nos seguintes elementos da solicitação SOAP:

- Seu ID de chave de acesso da AWS

### Note

Ao fazer solicitações SOAP autenticadas, não há suporte para credenciais de segurança temporárias. Para obter mais informações sobre os tipos de credenciais, consulte [Fazer solicitações \(p. 10\)](#).

- **Timestamp:** Deve ser um dateTime (acesse <http://www.w3.org/TR/xmlschema-2/#dateTime>) o fuso horário universal coordenado (horário médio de Greenwich), como 2009-01-01T12:00:00.000Z. A autorização falhará se esse time stamp tiver mais de 15 minutos de diferença do relógio nos servidores do Amazon S3.
- **Signature:** O resumo RFC 2104 HMAC-SHA1 (acesse <http://www.ietf.org/rfc/rfc2104.txt>) da concatenação de "AmazonS3" + OPERAÇÃO + Timestamp, usando sua chave de acesso secreta da AWS como chave. Por exemplo, na solicitação de exemplo CreateBucket a seguir, o elemento de assinatura conteria o resumo HMAC-SHA1 do valor "AmazonS3CreateBucket2009-01-01T12:00:00.000Z":

Por exemplo, na solicitação de exemplo CreateBucket a seguir, o elemento de assinatura conteria o resumo HMAC-SHA1 do valor "AmazonS3CreateBucket2009-01-01T12:00:00.000Z":

### Example

```
<CreateBucket xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>quotes</Bucket>
  <Acl>private</Acl>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2009-01-01T12:00:00.000Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</CreateBucket>
```

### Note

As solicitações SOAP, autenticadas e anônimas, devem ser enviadas para o Amazon S3 usando SSL. O Amazon S3 retorna um erro quando você envia uma solicitação SOAP via HTTP.

### Important

Devido a interpretações diferentes em relação a como a precisão de tempo extra deve ser aplicada, os usuários de .NET devem tomar cuidado para não enviar ao Amazon S3 datas e horas excessivamente específicas. Isso pode ser realizado criando manualmente objetos `DateTime` com precisão de apenas milissegundos.

## Configurar políticas de acesso padrão com SOAP

### Note

O suporte de SOAP via HTTP está obsoleto, mas continua disponível via HTTPS. Novos recursos do Amazon S3 não serão compatíveis com o SOAP. Recomendamos que você use o REST API ou os SDKs da AWS.

O controle de acesso pode ser definido no momento em que um bucket ou um objeto são gravados ao se incluir o elemento “AccessControlList” com a solicitação para `CreateBucket`, `PutObjectInline` ou `PutObject`. O elemento `AccessControlList` está descrito em [Gerenciamento de permissões de acesso aos recursos do Amazon S3 \(p. 282\)](#). Se nenhuma lista de controle de acesso for especificada com essas operações, o recurso será criado com uma política de acesso padrão que dá ao solicitante acesso `FULL_CONTROL` (esse é o caso mesmo que a solicitação seja uma solicitação `PutObjectInline` ou `PutObject` para um objeto que já exista).

A seguir está uma solicitação que grava dados em um objeto, torna o objeto legível por administradores anônimos e dá ao usuário especificado direitos `FULL_CONTROL` sobre o bucket (a maioria dos desenvolvedores vai querer dar a si mesmo acesso `FULL_CONTROL` a seu próprio bucket).

### Example

A seguir está uma solicitação que grava dados em um objeto e torna o objeto legível por administradores anônimos.

#### Sample Request

```
<PutObjectInline xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>quotes</Bucket>
  <Key>Nelson</Key>
  <Metadata>
    <Name>Content-Type</Name>
    <Value>text/plain</Value>
  </Metadata>
  <Data>aGETaGE=</Data>
  <ContentLength>5</ContentLength>
  <AccessControlList>
    <Grant>
      <Grantee xsi:type="CanonicalUser">
        <ID>75cc57f09aa0c8caeab4f8c24e99d10f8e7faebf76c078efc7c6caea54ba06a</ID>
        <DisplayName>chriscustomer</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
    <Grant>
      <Grantee xsi:type="Group">
        <URI>http://acs.amazonaws.com/groups/global/AllUsers</URI>
      </Grantee>
      <Permission>READ</Permission>
    </Grant>
  </AccessControlList>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2009-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</PutObjectInline>
```

#### Sample Response

```
<PutObjectInlineResponse xmlns="http://s3.amazonaws.com/doc/2006-03-01">
  <PutObjectInlineResponse>
    <ETag>"828ef3dfa96f00ad9f27c383fc9ac7f"</ETag>
```

```
<LastModified>2009-01-01T12:00:00.000Z</LastModified>
</PutObjectInlineResponse>
</PutObjectInlineResponse>
```

A política de controle de acesso pode ser lida ou configurada para um bucket ou objeto existentes usando os métodos `GetBucketAccessControlPolicy`, `GetObjectAccessControlPolicy`, `SetBucketAccessControlPolicy` e `SetObjectAccessControlPolicy`. Para obter mais informações, consulte a explicação detalhada desses métodos.

## Apêndice B: autenticação de solicitações (versão 2 do AWS Signature)

### Important

Esta seção descreve como autenticar solicitações usando o AWS Signature versão 2. O Signature versão 2 será descontinuado. Haverá suporte a ele somente até dia 24 de junho de 2019.

Depois de 24 de junho de 2019, o Amazon S3 só aceitará solicitações de API assinadas com o Signature versão 4. O Signature versão 4 oferece suporte para todas as regiões da AWS; é a única versão compatível com novas regiões. Para obter mais informações, consulte [Autenticação de solicitações \(AWS Signature versão 4\)](#) no Amazon Simple Storage Service API Reference .

### Tópicos

- [Autenticar solicitações usando a API REST \(p. 657\)](#)
- [Assinar e autenticar as solicitações REST \(p. 659\)](#)
- [Uploads baseados no navegador usando POST \(Signature versão 2 da AWS\) \(p. 668\)](#)

## Autenticar solicitações usando a API REST

Ao acessar o Amazon S3 usando REST, você deve fornecer os seguintes itens na solicitação para que ela seja autenticada:

### Elementos da solicitação

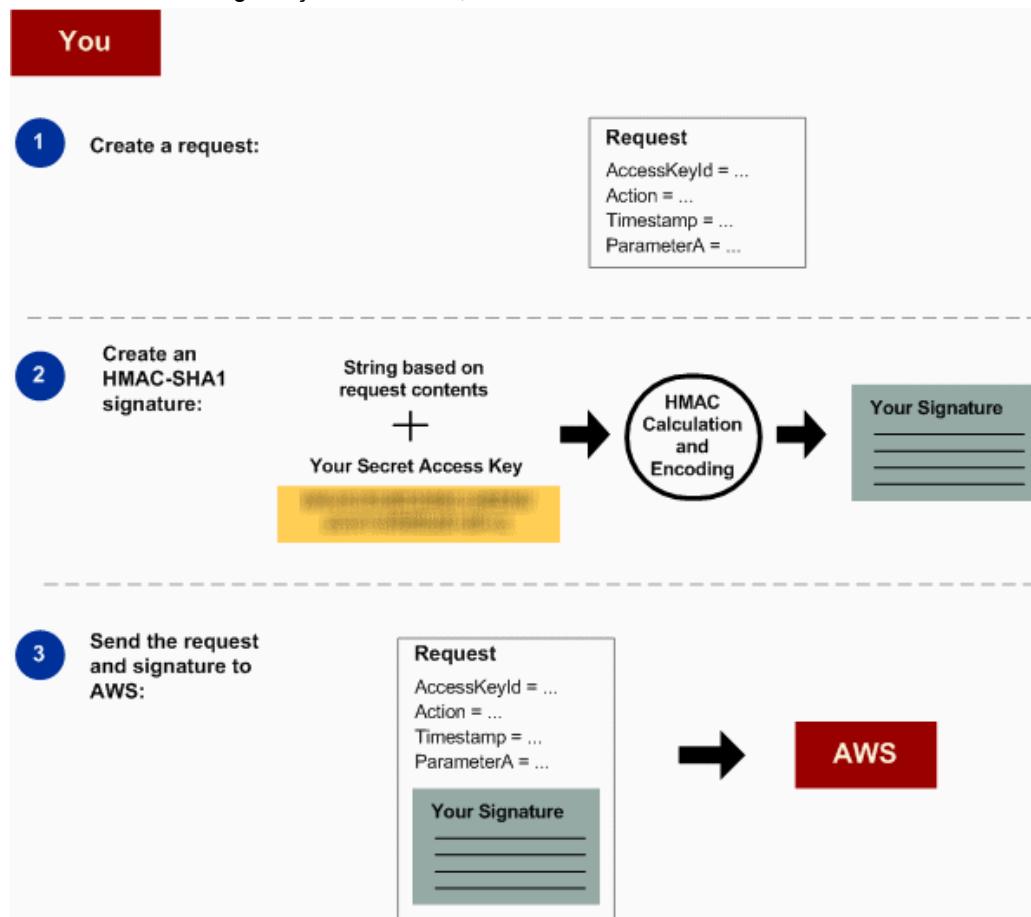
- ID de chave de acesso da AWS – cada solicitação deve conter o ID de chave de acesso da identidade que você estiver usando para enviar a solicitação.
- Assinatura – cada solicitação deve conter uma assinatura de solicitação válida. Do contrário, a solicitação será rejeitada.

Uma assinatura de solicitação é calculada com a chave de acesso secreta, um segredo compartilhado conhecido apenas por você e pela AWS.

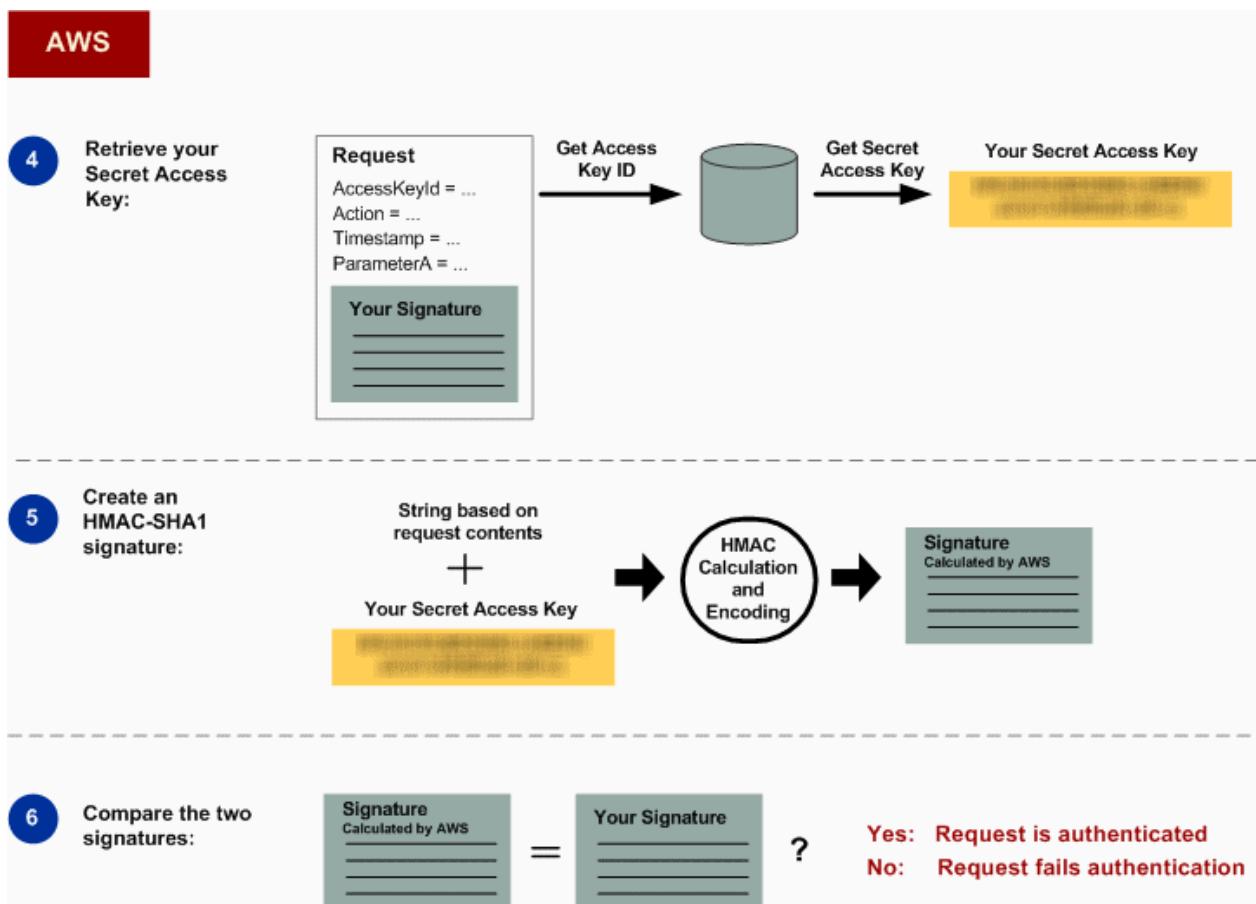
- Time stamp – cada solicitação deve conter a data e a hora de criação da solicitação, representada como uma string no UTC.
- Data – cada solicitação deve conter o time stamp da solicitação.

Dependendo da ação da API que você está usando, é possível fornecer uma data e hora de expiração para a solicitação em vez (ou além) do time stamp. Consulte o tópico de autenticação da respectiva ação para determinar o que é necessário.

Veja a seguir as etapas gerais para autenticar solicitações para o Amazon S3. Pressupõe-se que você tem as credenciais de segurança necessárias, o ID de chave de acesso e a chave de acesso secreta.



1	Crie uma solicitação para a AWS.
2	Calcule a assinatura usando a chave de acesso secreta.
3	Envie a solicitação para o Amazon S3. Inclua o ID de chave de acesso e a assinatura na solicitação. O Amazon S3 realiza as três etapas a seguir.



4	O Amazon S3 usa o ID de chave de acesso para pesquisar a chave de acesso secreta.
5	O Amazon S3 calcula uma assinatura a partir dos dados da solicitação e da chave de acesso secreta usando o mesmo algoritmo usado para calcular a assinatura enviada na solicitação.
6	Se a assinatura gerada pelo Amazon S3 corresponder à enviada na solicitação, ela é considerada autêntica. Se a comparação falhar, a solicitação será descartada e o Amazon S3 retornará uma resposta de erro.

## Informações de autenticação detalhadas

Para obter informações detalhadas sobre a autenticação REST, consulte [Assinar e autenticar as solicitações REST \(p. 659\)](#).

# Assinar e autenticar as solicitações REST

## Tópicos

- [Uso de credenciais de segurança temporárias \(p. 660\)](#)
- [Cabeçalho de autenticação \(p. 660\)](#)
- [Canonização de solicitação para assinatura \(p. 661\)](#)
- [Criar o elemento CanonicalizedResource \(p. 661\)](#)
- [Criar o elemento CanonicalizedAmzHeaders \(p. 662\)](#)
- [Elementos StringToSign de cabeçalho HTTP posicionais versus nomeados \(p. 662\)](#)
- [Requisito de time stamp \(p. 663\)](#)
- [Exemplos de autenticação \(p. 663\)](#)
- [Problemas de assinatura de solicitação REST \(p. 666\)](#)
- [Alternativa de autenticação de solicitação por query string \(p. 666\)](#)

## Note

Este tópico explica como autenticar solicitações usando o Signature Versão 2. O Amazon S3 agora oferece suporte à versão mais recente do Signature, a versão 4. Essa versão mais recente de assinatura é compatível com todas as regiões e qualquer nova região depois de 30 de janeiro de 2014 oferecerá suporte somente ao Signature versão 4. Para obter mais informações, consulte [Autenticar solicitações \(AWS Signature versão 4\)](#) no Amazon Simple Storage Service API Reference.

Autenticação é o processo de provar sua identidade ao sistema. A identidade é um fator importante nas decisões de controle de acesso do Amazon S3. As solicitações são permitidas ou negadas em parte com base na identidade do solicitante. Por exemplo, o direito de criar buckets está reservado a desenvolvedores registrados e (por padrão) o direito de criar objetos em um bucket está reservado para o proprietário do bucket em questão. Como um desenvolvedor, você fará solicitações que invocam esses privilégios e, portanto, precisará provar sua identidade ao sistema, autenticando suas solicitações. Esta seção explica como fazer isso.

## Note

O conteúdo nesta seção não se aplica a HTTP POST. Para obter mais informações, consulte [Uploads baseados no navegador usando POST \(Signature versão 2 da AWS\) \(p. 668\)](#).

A API REST do Amazon S3 usa um esquema HTTP personalizado com base em um HMAC de chave (código de autenticação de mensagem hash) para autenticação. Para autenticar uma solicitação, você primeiro concatena elementos selecionados da solicitação para formar uma string. Depois, você pode usar sua chave de acesso secreta da AWS para calcular o HMAC dessa string. Informalmente, chamamos desse processo de “assinar a solicitação” e chamamos o resultado do algoritmo do HMAC de assinatura, pois ele simula as propriedades de segurança de uma assinatura real. Finalmente, você adiciona esta assinatura como um parâmetro da solicitação usando a sintaxe descrita nesta seção.

Quando o sistema recebe uma solicitação autenticada, ele busca a chave de acesso secreta da AWS que você afirma ter e a usa da mesma forma para computar uma assinatura para a mensagem que recebeu. Então, ele compara a assinatura que calculou com a assinatura apresentada pelo solicitante. Se houver correspondência entre as duas assinaturas, o sistema conclui que o solicitante deve ter acesso à chave de acesso secreta da AWS e, portanto, age com a autoridade do principal para quem a chave foi emitida. Se as duas assinaturas não correspondem, a solicitação é abandonada e o sistema responde com uma mensagem de erro.

## Example Solicitação REST do Amazon S3 autenticada

```
GET /photos/puppy.jpg HTTP/1.1
```

```
Host: johnsmith.s3.amazonaws.com
Date: Mon, 26 Mar 2007 19:37:58 +0000

Authorization: AWS AKIAIOSFODNN7EXAMPLE:frJIUN8DYpKDtOLCwo//yllqDzg=
```

## Uso de credenciais de segurança temporárias

Se você assinar sua solicitação usando credenciais de segurança temporárias (consulte [Fazer solicitações \(p. 10\)](#)), você deverá incluir o token de segurança correspondente em sua solicitação, adicionando o cabeçalho `x-amz-security-token`.

Quando você obtém credenciais de segurança temporárias usando a API do AWS Security Token Service, a resposta inclui credenciais de segurança temporárias e um token de sessão. Você fornece o valor do token de sessão no cabeçalho `x-amz-security-token` quando enviar solicitações para o Amazon S3. Para obter informações da API do AWS Security Token Service fornecida pelo IAM, acesse [Ação](#) no AWS Security Token Service API ReferenceGuide .

## Cabeçalho de autenticação

A API REST do Amazon S3 usa o cabeçalho padrão HTTP `Authorization` para passar informações de autenticação. (O nome do cabeçalho padrão é infeliz porque ele carrega informações de autenticação, não de autorização.) No esquema de autenticação do Amazon S3, o cabeçalho Autorização tem a seguinte forma:

```
Authorization: AWS AWSAccessKeyId:Signature
```

Um ID de chave de acesso da AWS e uma chave de acesso secreta da AWS são emitidos para os desenvolvedores quando eles se registram. Para autenticação de solicitação, o elemento `AWSAccessKeyId` identifica o ID de chave de acesso que foi usado para computar a assinatura e, indiretamente, o desenvolvedor que fez a solicitação.

O elemento `Signature` é o RFC 2104 HMAC-SHA1 dos elementos selecionados da solicitação e, portanto, a parte `Signature` do cabeçalho Autorização variará de uma solicitação para outra. Se a assinatura da solicitação calculada pelo sistema corresponder ao `Signature` incluído na solicitação, o solicitante terá demonstrado a posse da chave de acesso secreta da AWS. A solicitação será então processada na identidade do desenvolvedor para quem a chave foi emitida e com a autoridade dele.

A seguir está uma pseudogramática que ilustra a criação do cabeçalho da solicitação `Authorization`. (No exemplo, `\n` significa o ponto do código Unicode U+000A, geralmente chamado de nova linha).

```
Authorization = "AWS" + " " + AWSAccessKeyId + ":" + Signature;

Signature = Base64( HMAC-SHA1( YourSecretAccessKeyID, UTF-8-Encoding-
Of( StringToSign ) ) );

StringToSign = HTTP-Verb + "\n" +
Content-MD5 + "\n" +
Content-Type + "\n" +
Date + "\n" +
CanonicalizedAmzHeaders +
CanonicalizedResource;

CanonicalizedResource = [ "/" + Bucket ] +
<HTTP-Request-URI, from the protocol name up to the query string> +
[ subresource, if present. For example "?acl", "?location", "?logging", or "?torrent"];
```

CanonicalizedAmzHeaders = <described below>

O HMAC-SHA1 é um algoritmo definido pelo hash de chave [RFC 2104 para autenticação de mensagem](#). O algoritmo recebe como input duas strings de byte, uma chave e uma mensagem. Para a autenticação de solicitação do Amazon S3, use sua chave de acesso secreta da AWS (`YourSecretAccessKeyID`) como a chave e a codificação UTF-8 de `StringToSign` como a mensagem. A saída de HMAC-SHA1 também é uma string de byte, chamada de resumo. O parâmetro de solicitação `Signature` é criado pela codificação Base64 desse resumo.

## Canonização de solicitação para assinatura

Lembre-se de que quando o sistema recebe uma solicitação autenticada, ele compara a assinatura de solicitação computada com a assinatura fornecida na solicitação em `StringToSign`. Por esse motivo, você deve computar a assinatura usando o mesmo método usado pelo Amazon S3. Nós chamamos o processo de colocar uma solicitação em um formulário estabelecido para assinatura de canonização.

## Criar o elemento CanonicalizedResource

`CanonicalizedResource` representa o recurso do Amazon S3 visado pela solicitação. Crie-o para uma solicitação REST como se segue:

Iniciar processo

1	Inicie com uma string vazia ("").
2	<p>Se a solicitação especificar um bucket usando o cabeçalho de host HTTP (estilo hosted virtual), adicione o nome do bucket precedido por uma "/" (por exemplo, "/bucketname"). Para solicitações de estilo de caminho e solicitações que não seja endereçada a um bucket, não faça nada. Para obter mais informações sobre solicitações de estilo hosted virtual, consulte <a href="#">Hospedagem virtual de buckets (p. 46)</a>.</p> <p>Para uma solicitação em estilo hosted virtual "https://johnsmith.s3.amazonaws.com/photos/puppy.jpg", o <code>CanonicalizedResource</code> é "/johnsmith".</p> <p>Para uma solicitação em estilo de caminho, "https://s3.amazonaws.com/johnsmith/photos/puppy.jpg", o <code>CanonicalizedResource</code> é "".</p>
3	<p>Adicione a parte do caminho de um URI de solicitação HTTP descodificado, até a query string, mas sem incluí-la.</p> <p>Para uma solicitação em estilo hosted virtual "https://johnsmith.s3.amazonaws.com/photos/puppy.jpg", o <code>CanonicalizedResource</code> é "/johnsmith/photos/puppy.jpg".</p> <p>Para uma solicitação em estilo de caminho, "https://s3.amazonaws.com/johnsmith/photos/puppy.jpg", o <code>CanonicalizedResource</code> é "/johnsmith/photos/puppy.jpg". Neste ponto, o <code>CanonicalizedResource</code> é o mesmo para a solicitação em estilo hosted virtual e em estilo de caminho.</p> <p>Para uma solicitação que não seja endereçada a um bucket, como <a href="#">GET Service</a>, adicione "/".</p>
4	<p>Se a solicitação endereça um sub-recurso, como <code>?versioning</code>, <code>?location</code>, <code>?acl</code>, <code>?torrent</code>, <code>?lifecycle</code> ou <code>?versionid</code>, adicione o sub-recurso, seu valor, se houver um, e o ponto de interrogação. Observe que, em caso de vários sub-recursos, os sub-recursos devem ser classificados em ordem lexicográfica por nome de sub-recurso e ser separados por '&amp;', por exemplo, <code>?acl&amp;versionId=value</code>.</p> <p>Os sub-recursos que devem ser incluídos ao criar o elemento de <code>CanonicalizedResource</code> são <code>acl</code>, <code>ciclo de vida</code>, <code>local</code>, <code>registro</code>, <code>notificação</code>, <code>partNumber</code>, <code>política</code>, <code>requestPayment</code>, <code>torrent</code>, <code>uploadId</code>, <code>uploads</code>, <code>versionId</code>, <code>versionamento</code>, <code>versões</code> e <code>site</code>.</p>

Se a solicitação especificar os parâmetros de query string que cancelam os valores de cabeçalho de resposta (consulte [Objeto GET](#)), adicione os parâmetros de query string e seus valores. Ao assinar, você não codifica esses valores; contudo, ao fazer a solicitação, você deve codificar esses valores de parâmetros. Os parâmetros de query string em uma solicitação GET incluem `response-content-type`, `response-content-language`, `response-expires`, `response-cache-control`, `response-content-disposition` e `response-content-encoding`.

O parâmetro de query string `delete` deve ser incluído ao criar o CanonicalizedResource para uma solicitação de exclusão de vários objetos.

Os elementos do CanonicalizedResource que vêm da URI da solicitação HTTP devem ser assinados literalmente como aparecem na solicitação HTTP, incluindo metacaracteres de codificação de URL.

O CanonicalizedResource pode ser diferente da URI da solicitação HTTP. Em particular, se sua solicitação usa o cabeçalho HTTP `Host` para especificar um bucket, o bucket não aparece na URI da solicitação HTTP. Contudo, o CanonicalizedResource continua a incluir o bucket. Os parâmetros de query string podem também aparecer na URI da solicitação, mas não estão incluídos em CanonicalizedResource. Para obter mais informações, consulte [Hospedagem virtual de buckets \(p. 46\)](#).

## Criar o elemento CanonicalizedAmzHeaders

Para criar a parte de CanonicalizedAmzHeaders de `StringToSign`, selecione todos os cabeçalhos de solicitações HTTP que comecem com '`x-amz-`' (usando uma comparação que não diferencie maiúsculas e minúsculas) e use o processo a seguir.

### Processo de CanonicalizedAmzHeaders

- |   |  |
|---|--|
| 1 | Converta cada nome de cabeçalho HTTP para minúsculas. Por exemplo, ' <code>X-Amz-Date</code> ' torna-se ' <code>x-amz-date</code> '.   |
| 2 | Classifique a coleção de cabeçalhos por ordem lexicográfica por nome de cabeçalho.   |
| 3 | Combine campos de cabeçalho com o mesmo nome em um par "header-name:comma-separated-value-list" como prescrito por RFC 2616, seção 4.2, sem qualquer espaço entre os valores. Por exemplo, os dois cabeçalhos de metadados ' <code>x-amz-meta-username: fred</code> ' e ' <code>x-amz-meta-username: barney</code> ' seriam combinados em único cabeçalho ' <code>x-amz-meta-username: fred,barney</code> '. |
| 4 | "Desdobre" os cabeçalhos longos que abrangem várias linhas (como permitido por RFC 2616, seção 4.2) substituindo o espaço de dobramento (incluindo a nova linha) por um único espaço.  |
| 5 | Retire qualquer espaço ao redor dos dois pontos no cabeçalho. Por exemplo, o cabeçalho ' <code>x-amz-meta-username: fred, barney</code> ' iria se tornar ' <code>x-amz-meta-username:fred,barney</code> '  |
| 6 | Finalmente, adicione um caractere de nova linha ( <code>\n</code> ) para cada cabeçalho canonizado na lista resultante. Crie o elemento CanonicalizedResource concatenando todos os cabeçalhos dessa lista em uma única string.  |

## Elementos StringToSign de cabeçalho HTTP posicionais versus nomeados

Os primeiros elementos de cabeçalho do `StringToSign` (Content-Type, Date e Content-MD5) são de natureza posicional. `StringToSign` não inclui os nomes desses cabeçalhos, somente seus valores da solicitação. Em contraste, os elementos '`x-amz-`' são nomeados. Os nomes de cabeçalho e os valores de cabeçalho aparecem em `StringToSign`.

Se um cabeçalho posicional chamado para a definição de `StringToSign` não estiver presente na sua solicitação (por exemplo, `Content-Type` ou `Content-MD5` são opcionais para solicitações PUT e sem sentido para solicitações GET), substitua a string vazia ("") para essa posição.

## Requisito de time stamp

Um time stamp válido (usando o cabeçalho HTTP `Date` ou uma alternativa `x-amz-date`) é obrigatório para solicitações autenticadas. Além disso, o time stamp do cliente, incluído com uma solicitação autenticada, não deve exceder 15 minutos do tempo do sistema do Amazon S3 quando a solicitação é recebida. Caso contrário, haverá falha na solicitação com o código de erro `RequestTimeTooSkewed`. A intenção dessas restrições é limitar a possibilidade de que solicitações interceptadas possam ser reenviadas por um adversário. Para uma proteção mais forte contra espionagem, use o transporte HTTPS para solicitações autenticadas.

### Note

A restrição de validação na data da solicitação se aplica somente a solicitações autenticadas que não usem a autenticação por query string. Para obter mais informações, consulte [Alternativa de autenticação de solicitação por query string \(p. 666\)](#).

Algumas bibliotecas de clientes HTTP não expõem a capacidade para configurar o cabeçalho `Date` para uma solicitação. Se tiver problemas para incluir o valor do cabeçalho "Data" nos cabeçalhos canonizados, você pode configurar o time stamp para a solicitação usando um cabeçalho '`x-amz-date`'. O valor do cabeçalho `x-amz-date` deve estar em um dos formatos de RFC 2616 (<http://www.ietf.org/rfc/rfc2616.txt>). Quando um cabeçalho `x-amz-date` está presente em uma solicitação, o sistema ignorará qualquer cabeçalho `Date` ao calcular a assinatura da solicitação. Portanto, se você incluir o cabeçalho `x-amz-date`, use a string vazia para o `Date` quando criar o `StringToSign`. Consulte a próxima seção para ver um exemplo.

## Exemplos de autenticação

Os exemplos nesta seção usam as credenciais (não trabalho) na tabela a seguir.

Parâmetro	Valor
<code>AWSAccessKeyId</code>	<code>AKIAIOSFODNN7EXAMPLE</code>
<code>AWSSecretAccessKey</code>	<code>wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY</code>

No exemplo `StringToSigns`, o formato não é significativo e `\n` significa o ponto de código Unicode U +000A, chamado geralmente de nova linha. Além disso, os exemplos usam "+0000" para designar o fuso horário. Você pode usar "GMT" para designar o fuso horário, mas as assinaturas mostradas nos exemplos serão diferentes.

## Objeto GET

Este exemplo obtém um objeto do bucket `johnsmith`.

Solicitação	<code>StringToSign</code>
<code>GET /photos/puppy.jpg HTTP/1.1</code> <code>Host: johnsmith.s3.amazonaws.com</code> <code>Date: Tue, 27 Mar 2007 19:36:42 +0000</code>  <code>Authorization: AWS</code> <code>AKIAIOSFODNN7EXAMPLE:</code>	<code>GET\n\n\nTue, 27 Mar 2007 19:36:42 +0000\n/johnsmith/photos/puppy.jpg</code>

Solicitação	StringToSign
<code>bWq2s1WEIj+Ydj0vQ697zp+IXMU=</code>	

Observe que o CanonicalizedResource inclui o nome do bucket, mas a URI da solicitação HTTP não o inclui. (O bucket é especificado pelo cabeçalho de host.)

## PUT objeto

Este exemplo põe um objeto no bucket johnsmith.

Solicitação	StringToSign
<pre>PUT /photos/puppy.jpg HTTP/1.1 Content-Type: image/jpeg Content-Length: 94328 Host: johnsmith.s3.amazonaws.com Date: Tue, 27 Mar 2007 21:15:45 +0000  Authorization: AWS AKIAIOSFODNN7EXAMPLE: MyyxerY7whkBe+bq8fHCL/2kKUg=</pre>	<pre>PUT\n \n image/jpeg\n Tue, 27 Mar 2007 21:15:45 +0000\n /johnsmith/photos/puppy.jpg</pre>

Observe o cabeçalho Content-Type na solicitação e em StringToSign. Também observe que o Content-MD5 está em branco em StringToSign porque não está presente na solicitação.

## Lista

Este exemplo lista o conteúdo do bucket johnsmith.

Solicitação	StringToSign
<pre>GET /?prefix=photos&amp;max-keys=50&amp;marker=puppy HTTP/1.1 User-Agent: Mozilla/5.0 Host: johnsmith.s3.amazonaws.com Date: Tue, 27 Mar 2007 19:42:41 +0000  Authorization: AWS AKIAIOSFODNN7EXAMPLE: htDYFYduRNen8P9ZfE/s9SuKy0U=</pre>	<pre>GET\n \n \n Tue, 27 Mar 2007 19:42:41 +0000\n /johnsmith/</pre>

Observe a barra final no CanonicalizedResource e a ausência de parâmetros de query string.

## Fetch

Este exemplo busca o sub-recurso de política de controle de acesso para o bucket 'johnsmith'.

Solicitação	StringToSign
<pre>GET /?acl HTTP/1.1 Host: johnsmith.s3.amazonaws.com Date: Tue, 27 Mar 2007 19:44:46 +0000  Authorization: AWS AKIAIOSFODNN7EXAMPLE: c2WLPFtWHVgbEmEg93a4cG37dM=</pre>	<pre>GET\n \n \n Tue, 27 Mar 2007 19:44:46 +0000\n /johnsmith/?acl</pre>

Observe como o parâmetro de query string do sub-recurso está incluído no CanonicalizedResource.

## Excluir

Este exemplo exclui um objeto do bucket “johnsmith” usando o caminho- estilo e a alternativa de data.

Solicitação	StringToSign
<pre>DELETE /johnsmith/photos/puppy.jpg HTTP/1.1 User-Agent: dotnet Host: s3.amazonaws.com Date: Tue, 27 Mar 2007 21:20:27 +0000 x-amz-date: Tue, 27 Mar 2007 21:20:26 +0000 Authorization: AWS AKIAIOSFODNN7EXAMPLE:lx3byBScXR6KzyMaifNkardMwNk=</pre>	<pre>DELETE\n \n \n Tue, 27 Mar 2007 21:20:26 +0000\n /johnsmith/photos/puppy.jpg</pre>

Observe como usamos o método alternativo 'x-amz- data' de especificar a data (porque a biblioteca de cliente nos impediua de configurar a data, por exemplo). Nesse caso, o x-amz-date tem precedência sobre o cabeçalho Date. Portanto, entrada de data na assinatura deve conter o valor de cabeçalho x-amz-date.

## Carregar

Este exemplo faz upload de um objeto para um bucket de estilo hosted virtual CNAME com metadados.

Solicitação	StringToSign
<pre>PUT /db-backup.dat.gz HTTP/1.1 User-Agent: curl/7.15.5 Host: static.johnsmith.net:8080 Date: Tue, 27 Mar 2007 21:06:08 +0000  x-amz-acl: public-read Content-Type: application/x-download Content-MD5: 4gJE4saaMU4BqNR0kLY+lw== X-Amz-Meta-ReviewedBy: joe@johnsmith.net X-Amz-Meta-ReviewedBy: jane@johnsmith.net X-Amz-Meta-FileChecksum: 0x02661779 X-Amz-Meta-ChecksumAlgorithm: crc32 Content-Disposition: attachment; filename=database.dat Content-Encoding: gzip Content-Length: 5913339  Authorization: AWS AKIAIOSFODNN7EXAMPLE: iIyl83RwaSoYIEdixDQcA4OnAnc=</pre>	<pre>PUT\n 4gJE4saaMU4BqNR0kLY+lw==\n application/x-download\n Tue, 27 Mar 2007 21:06:08 +0000\n  x-amz-acl:public-read\n x-amz-meta-checksumalgorithm:crc32\n x-amz-meta-filechecksum:0x02661779\n x-amz-meta-reviewedby: joe@johnsmith.net,jane@johnsmith.net\n /static.johnsmith.net/db-backup.dat.gz</pre>

Observe como os cabeçalhos 'x-amz-' são classificados, os espaços são excluídos e os cabeçalhos são convertidos para minúsculas. Observe também que vários cabeçalhos com o mesmo nome foram unidos usando vírgulas para separar valores.

Observe como somente os cabeçalhos de entidade HTTP Content-Type e Content-MD5 aparecem em StringToSign. Os outros cabeçalhos de entidade Content-\* não aparecem.

Mais uma vez, observe que o CanonicalizedResource inclui o nome do bucket, mas a URI da solicitação HTTP não o inclui. (O bucket é especificado pelo cabeçalho de host.)

## Relacionar todos os meus buckets

Solicitação	StringToSign
<pre>GET / HTTP/1.1 Host: s3.amazonaws.com Date: Wed, 28 Mar 2007 01:29:59 +0000  Authorization: AWS AKIAIOSFODNN7EXAMPLE:qGdzdERIC03wnaRNKh6OqZehG9s=</pre>	<pre>GET\n \n \n Wed, 28 Mar 2007 01:29:59 +0000\n /</pre>

## Chaves Unicode

Solicitação	StringToSign
<pre>GET /dictionary/fran%C3%A7ais/pr%C3%a9f %c3%a8re HTTP/1.1 Host: s3.amazonaws.com Date: Wed, 28 Mar 2007 01:49:49 +0000 Authorization: AWS AKIAIOSFODNN7EXAMPLE:DNEZGsoieTZ92F3bUfSPQcbGm9e3%a8re</pre>	<pre>GET\n \n \n Wed, 28 Mar 2007 01:49:49 +0000\n /dictionary/fran%C3%A7ais/pr%C3%a9f %c3%a8re</pre>

### Note

Os elementos em StringToSign que foram derivados da URI de solicitação são obtidos literalmente, incluindo a codificação do URL e a capitalização.

## Problemas de assinatura de solicitação REST

Quando a autenticação de solicitação REST falha, o sistema responde à solicitação com um documento de erro em XML. As informações contidas neste documento de erro têm o objetivo de ajudar os desenvolvedores a diagnosticar o problema. Especificamente, o elemento StringToSign do documento de erro SignatureDoesNotMatch diz exatamente que canonização de solicitação o sistema está usando.

Alguns toolkits inserem silenciosamente os cabeçalhos que você não conhece antecipadamente, como a adição do cabeçalho Content-Type durante um PUT. Na maioria desses casos, o valor do cabeçalho inserido permanece constante, permitindo que você descubra os cabeçalhos que faltam, usando ferramentas como Ethereal ou o tcpmon.

## Alternativa de autenticação de solicitação por query string

Você pode autenticar determinados tipos de solicitações passando as informações necessárias como parâmetros de query string em vez de usar o cabeçalho HTTP Authorization. Isso é útil para habilitar o acesso de navegadores de terceiros a seus dados privados do Amazon S3 sem um proxy na solicitação. A ideia é criar uma solicitação "pré-assinada" e codificá-la como um URL que o navegador de um usuário final pode recuperar. Além disso, você pode limitar uma solicitação pré-assinada, especificando um tempo de expiração.

### Note

Para exemplos de uso de AWS SDKs para gerar pre-signed URLs, consulte [Compartilhe um objeto \(p. 172\)](#).

## Criar uma assinatura

A seguir está um exemplo de solicitação REST do Amazon S3 autenticado por query string.

```
GET /photos/puppy.jpg
?AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Expires=1141889120&Signature=vjbyPxybdZaNmGa
%2ByT272YEAiv4%3D HTTP/1.1
Host: johnsmith.s3.amazonaws.com
Date: Mon, 26 Mar 2007 19:37:58 +0000
```

O método de autenticação por query string não requer nenhum cabeçalho especial HTTP. Em vez disso, os elementos de autenticação exigidos são especificados como parâmetros de query string:

Nome de parâmetro de query string	Valor de exemplo	Descrição
AWSAccessKeyId	AKIAIOSFODNN7EXAMPLE	Seu ID de chave de acesso da AWS. Especifica a chave de acesso secreta da AWS usada para assinar a solicitação e, indiretamente, a identidade de desenvolvedor que fez a solicitação.
Expires	1141889120	O tempo quando a assinatura vai expirar, especificado como o número de segundos desde o epoch (00:00:00 UTC em 1º de janeiro de 1970). Uma requisição recebida depois desse tempo (de acordo com o servidor) será rejeitada.
Signature	vjbyPxybdZaNmGa %2ByT272YEAiv4%3D	A codificação do URL da codificação Base64 do HMAC-SHA1 de StringToSign.

O método de autenticação de solicitação por query string difere ligeiramente do método comum, mas somente no formato do parâmetro da solicitação `Signature` e no elemento `StringToSign`. A seguir está a pseudogramática que ilustra o método de autenticação de solicitação por query string.

```
Signature = URL-Encode( Base64( HMAC-SHA1( YourSecretAccessKeyID, UTF-8-Encoding-Of( StringToSign ) ) ) );  
  
StringToSign = HTTP-VERB + "\n" +  
Content-MD5 + "\n" +  
Content-Type + "\n" +  
Expires + "\n" +  
CanonicalizedAmzHeaders +  
CanonicalizedResource;
```

`YourSecretAccessKeyID` é o ID da chave de acesso secreta da AWS que a Amazon atribui a você quando você se cadastra para ser um desenvolvedor da Amazon Web Services. Observe como o

Signature é codificado por URL para ser apropriado para a colocação na query string. Observe também que no StringToSign, o elemento posicional HTTP Date foi substituído por Expires. O CanonicalizedAmzHeaders e o CanonicalizedResource são os mesmos.

**Note**

No método de autenticação por query string, você não utiliza o cabeçalho Date nem o x-amz-date request para calcular o string para assinar.

### Autenticação de solicitação por query string

Solicitação	StringToSign
<pre>GET /photos/puppy.jpg? AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&amp; Signature=NpgCjnDzrM %2BWpzENXmpNDUsSn8%3D&amp; Expires=1175139620 HTTP/1.1 Host: johnsmith.s3.amazonaws.com</pre>	<pre>GET\n \n \n 1175139620\n /johnsmith/photos/puppy.jpg</pre>

Supomos que quando um navegador faz a solicitação GET, ele não fornece um cabeçalho Content-MD5 ou Content-Type, nem define os cabeçalhos x-amz- e assim essas partes de StringToSign são deixadas em branco.

### Usar codificação Base64

As assinaturas de solicitação HMAC devem ser codificadas em Base64. A codificação Base64 converte a assinatura em uma string simples ASCII que pode ser anexada à solicitação. Os caracteres que poderiam aparecer na string da assinatura como mais (+), barra (/) e igual (=) devem ser codificados se forem usados em uma URI. Por exemplo, se o código de autenticação inclui um sinal de mais (+), codifique-o como %2B na solicitação. Codifique uma barra como %2F e o sinal de igual como %3D.

Para obter exemplos de codificação Base64, consulte o Amazon S3 [Exemplos de autenticação \(p. 663\)](#).

## Uploads baseados no navegador usando POST (Signature versão 2 da AWS)

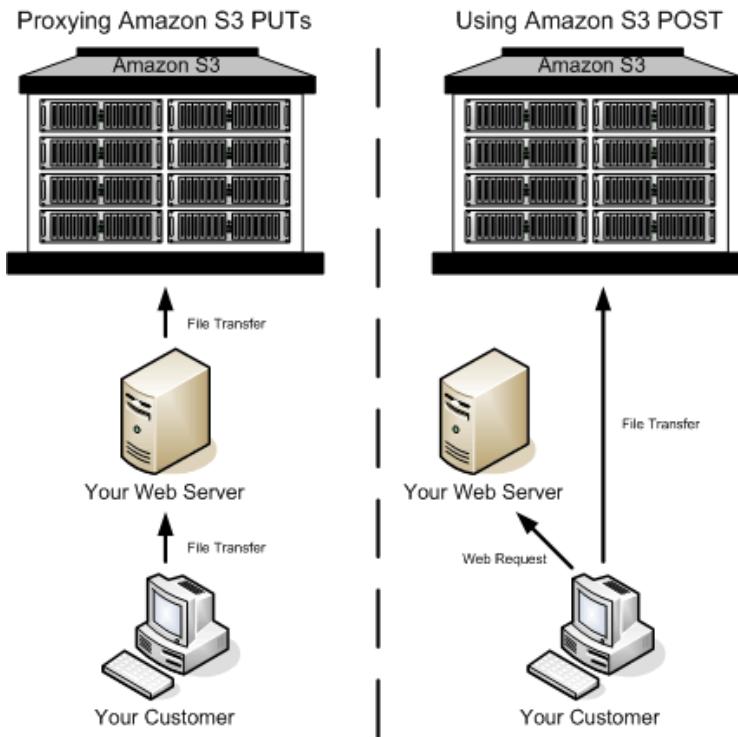
O Amazon S3 oferece suporte a POST, o que permite aos usuários carregar conteúdo diretamente para o Amazon S3. O POST é desenvolvido para simplificar uploads, reduzir a latência do upload e economizar o seu dinheiro nos aplicativos em que os usuários carregam dados para armazenamento no Amazon S3.

**Note**

A autenticação de solicitação discutida nesta seção é baseada no Signature versão 2 da AWS, um protocolo para autenticar solicitações de entrada da API para serviços da AWS.

Agora, o Amazon S3 oferece suporte para o Signature versão 4, um protocolo para autenticação de solicitações de entrada da API para serviços da AWS, em todas as regiões da AWS. Neste momento, as regiões da AWS criadas antes de 30 de janeiro de 2014 continuarão oferecendo suporte ao protocolo anterior, Signature versão 2. Todas as regiões novas a partir de 30 de janeiro de 2014 oferecerão suporte apenas ao Signature versão 4. Portanto, todas as solicitações para essas regiões devem ser feitas com o Signature versão 4. Para obter mais informações, consulte [Autenticar solicitações em uploads baseados no navegador usando POST \(Signature versão 4 da AWS\)](#) no Amazon Simple Storage Service API Reference.

A figura a seguir mostra um upload usando o POST do Amazon S3.



#### Fazer upload usando POST

1	O usuário abre um navegador e acessa sua página da web.
2	A página da web contém um formulário HTTP que contém toda as informações necessárias para que o usuário carregue conteúdo no Amazon S3.
3	O usuário carrega conteúdo diretamente no Amazon S3.

#### Note

Não há suporte para autenticação por query string para POST.

## Formulários HTML (Signature versão 2 da AWS)

### Tópicos

- [Codificação do formulário HTML \(p. 670\)](#)
- [Declaração de formulário HTML \(p. 670\)](#)
- [Campos do formulário HTML \(p. 671\)](#)
- [Criação de política \(p. 673\)](#)
- [Criar uma assinatura \(p. 676\)](#)
- [Redirecionamento \(p. 677\)](#)

Quando você se comunica com o Amazon S3, normalmente usa as APIs REST ou SOAP para executar as operações colocar, obter, excluir, entre outras. Com POST, os usuários carregam dados diretamente no Amazon S3 por meio dos navegadores, que não são capazes de processar a API SOAP ou criar uma solicitação PUT REST.

#### Note

O suporte de SOAP via HTTP está obsoleto, mas continua disponível via HTTPS. Novos recursos do Amazon S3 não serão compatíveis com o SOAP. Recomendamos que você use o REST API ou os SDKs da AWS.

Para permitir que os usuários carreguem conteúdo no Amazon S3 usando os navegadores, use formulários HTML. Os formulários HTML são formados por uma declaração de formulário e campos de formulário. A declaração de formulário contém informações de alto nível sobre a solicitação. Os campos de formulário contêm informações detalhadas sobre a solicitação, bem como a política usada para autenticá-la e garantir que ela satisfaça as condições especificadas.

#### Note

Os dados e os limites do formulário (excluindo o conteúdo do arquivo) não podem exceder 20 KB.

Esta seção explica como usar formulários HTML.

### Codificação do formulário HTML

O formulário e a política devem ser codificados em UTF-8. Aplique a codificação UTF-8 no formulário especificando isso no cabeçalho HTML ou como um cabeçalho de solicitação.

#### Note

A declaração de formulário HTML não aceita parâmetros de autenticação por query string.

A seguir um exemplo de codificação UTF-8 no cabeçalho HTML:

```
<html>
  <head>
    ...
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    ...
  </head>
  <body>
```

A seguir um exemplo de codificação UTF-8 em um cabeçalho de solicitação:

```
Content-Type: text/html; charset=UTF-8
```

### Declaração de formulário HTML

A declaração de formulário tem três componentes: a ação, o método e o tipo de compartimento. Se qualquer um desses valores for definido de maneira incorreta, a solicitação falhará.

A ação especifica o URL que processa a solicitação, que deve ser definido como o URL do bucket. Por exemplo, se o nome do bucket é "johnsmith", o URL é "http://johnsmith.s3.amazonaws.com/".

#### Note

O nome chave é especificado em um campo do formulário.

O método deve ser POST.

O tipo de compartimento (enctype) deve ser especificado e definido como multipart/form-data para uploads de arquivos e de áreas de texto. Para obter mais informações, acesse [RFC 1867](#).

#### Example

O exemplo a seguir é uma declaração de formulário para o bucket "johnsmith".

```
<form action="http://johnsmith.s3.amazonaws.com/" method="post">  
  enctype="multipart/form-data">
```

## Campos do formulário HTML

A tabela a seguir descreve os campos que podem ser usados em um formulário HTML.

### Note

A variável \${filename} é substituída automaticamente pelo nome do arquivo fornecido pelo usuário e é reconhecida por todos os campos do formulário. Se o navegador ou o cliente fornece um caminho completo ou parcial para o arquivo, apenas o texto que vem depois da última barra (/) ou barra invertida (\) será usado. Por exemplo, "C:\Program Files\directory1\file.txt" será interpretado como "file.txt". Se nenhum arquivo ou nome de arquivo for fornecido, a variável será substituída por uma string vazia.

Nome do campo	Descrição	Obrigatório
AWSAccessKeyId	O ID de chave de acesso da AWS do proprietário do bucket que concede acesso a um usuário anônimo para uma solicitação que satisfaz o conjunto de restrições na política. Este campo é necessário se a solicitação inclui um documento de política.	Condisional
acl	<p>Uma lista de controle de acesso (ACL) do Amazon S3. Se uma lista de controle de acesso inválida for especificada, um erro será gerado. Para obter mais informações sobre as ACLs, consulte <a href="#">Listas de controle de acesso (p. 7)</a>.</p> <p>Tipo: string</p> <p>Padrão: privado</p> <p>Valid Values: private   public-read   public-read-write   aws-exec-read   authenticated-read   bucket-owner-read   bucket-owner-full-control</p>	Não
Cache-Control, Content-Type, Content-Disposition, Content-Encoding, Expires	Cabeçalhos específicos para REST. Para obter mais informações, consulte <a href="#">Objeto PUT</a> .	Não
key	<p>O nome da chave carregada.</p> <p>Para usar o nome de arquivo fornecido pelo usuário, use a variável \${filename}. Por exemplo, se o usuário Betty carrega o arquivo lolcatz.jpg e você especifica /user/betty/\${filename}, o arquivo é armazenado como /user/betty/lolcatz.jpg.</p> <p>Para obter mais informações, consulte <a href="#">Chave de objeto e metadados (p. 102)</a>.</p>	Sim
policy	Política de segurança que descreve o que é permitido na solicitação. As solicitações sem	Não

Nome do campo	Descrição	Obrigatório
	<p>uma política de segurança são consideradas anônimas e terão sucesso apenas em buckets com gravação pública.</p>	
<code>success_action_redirect</code> , <code>redirect</code>	<p>A URL para a qual o cliente é redirecionado no upload bem-sucedido. O Amazon S3 anexa o bucket, a chave e os valores de tag como parâmetros de string de consulta à URL.</p> <p>Se <code>success_action_redirect</code> não for especificado, o Amazon S3 retornará o tipo de documento vazio especificado no campo <code>success_action_status</code>.</p> <p>Se o Amazon S3 não conseguir interpretar o URL, o campo será ignorado.</p> <p>Se o upload falhar, o Amazon S3 exibirá um erro e não redirecionará o usuário para um URL.</p> <p>Para obter mais informações, consulte <a href="#">Redirecionamento (p. 677)</a>.</p> <p><b>Note</b></p> <p>O nome do campo de redirecionamento está obsoleto e o suporte para ele será removido no futuro.</p>	Não
<code>success_action_status</code>	<p>O código de status retornado ao cliente após o upload bem-sucedido se <code>success_action_redirect</code> não for especificado.</p> <p>Os valores válidos são 200, 201 ou 204 (padrão).</p> <p>Se o valor está definido como 200 ou 204, o Amazon S3 retorna um documento vazio com um código de status 200 ou 204.</p> <p>Se o valor está definido como 201, o Amazon S3 retorna um documento XML com um código de status 201. Para obter informações sobre o conteúdo do documento XML, consulte <a href="#">Objeto POST</a>.</p> <p>Se o valor não está definido ou é um valor inválido, o Amazon S3 retorna um documento vazio com um código de status 204.</p> <p><b>Note</b></p> <p>Algumas versões do Adobe Flash Player não lidam muito bem com respostas HTTP com um corpo vazio. Para oferecer suporte a uploads por meio do Adobe Flash, recomendamos definir <code>success_action_status</code> como 201.</p>	Não

Nome do campo	Descrição	Obrigatório
signature	<p>A assinatura HMAC criada com a chave de acesso secreta correspondente ao AWSAccessKeyId fornecido. Este campo é necessário se um documento de política estiver incluso na solicitação.</p> <p>Para obter mais informações, consulte <a href="#">Usar acesso de autenticação</a>.</p>	Condisional
x-amz-security-token	<p>Token de segurança usado por credenciais de sessão do</p> <p>Se a solicitação estiver usando o Amazon DevPay, serão necessários dois campos de formulário x-amz-security-token: um para o token de produto e outro para o token de usuário.</p> <p>Se a solicitação estiver usando credenciais de sessão, será necessário um formulário x-amz-security-token. Para obter mais informações, consulte Credenciais de segurança temporárias no .</p>	Não
Outros nomes de campos com o prefixo x-amz-meta-	<p>Metadados especificados pelo usuário.</p> <p>O Amazon S3 não valida ou usa esses dados.</p> <p>Para obter mais informações, consulte <a href="#">Objeto PUT</a>.</p>	Não
file	<p>Conteúdo de arquivo ou texto.</p> <p>O arquivo ou conteúdo deve ser o último campo no formulário. Todos os campos abaixo deles serão ignorados.</p> <p>Não carregue mais de um arquivo por vez.</p>	Sim

## Criação de política

### Tópicos

- [Expiração \(p. 674\)](#)
- [Condições \(p. 674\)](#)
- [Correspondência de condição \(p. 675\)](#)
- [Caracteres de escape \(p. 676\)](#)

A política é um documento JSON codificado em Base64 e UTF-8 que especifica as condições que a solicitação deve satisfazer, sendo usado para autenticar o conteúdo. Dependendo de como os documentos de política forem elaborados, eles podem ser usados por upload, por usuário, para todos os uploads ou de acordo com outros formatos que atendam as suas necessidades.

### Note

Embora o documento de política seja opcional, o recomendamos fortemente em vez de tornar um bucket aberto ao público para gravação.

Veja a seguir um exemplo de um documento de política:

```
{ "expiration": "2007-12-01T12:00:00.000Z",
  "conditions": [
    {"acl": "public-read"},
    {"bucket": "johnsmith"},
    ["starts-with", "$key", "user/eric/"],
  ]
}
```

O documento de política contém a expiração e as condições.

### Expiração

O elemento `expiração` especifica a data de expiração da política no formato de data UTC ISO 8601. Por exemplo, "2007-12-01T12:00:00.000Z" especifica que a política não tem mais validade depois da meia-noite UTC do dia 1º de dezembro de 2007. A expiração é necessária em uma política.

### Condições

As condições no documento de política validam o conteúdo do objeto carregado. Cada campo especificado no formulário (exceto `AWSAccessKeyId`, assinatura, arquivo, política e nomes de campos com o prefixo `x-ignore-`) deve estar incluso na lista de condições.

#### Note

Caso existam vários campos com o mesmo nome, os valores devem ser separados por vírgulas. Por exemplo, se existem dois campos chamados "`x-amz-meta-tag`", o primeiro tem o valor "Ninja" e o segundo tem o valor "Stallman", o documento de política seria definido como `Ninja, Stallman`.

Todas as variáveis dentro do formulário são expandidas antes da validação da política. Portanto, qualquer correspondência de condição deve ser realizada nos campos expandidos. Por exemplo, se o campo chave for definido como `user/betty/${filename}`, a política deve ser `[ "starts-with", "$key", "user/betty/" ]`. Não insira `[ "starts-with", "$key", "user/betty/${filename}" ]`. Para obter mais informações, consulte [Correspondência de condição \(p. 675\)](#).

A tabela a seguir descreve as condições do documento de política.

Nome do elemento	Descrição
<code>acl</code>	Especifica as condições que a ACL deve satisfazer. Oferece suporte à correspondência exata e a <code>starts-with</code> .
<code>content-length-range</code>	Especifica os tamanhos mínimo e máximo permitidos para o conteúdo carregado. Oferece suporte à correspondência por intervalo.
<code>Cache-Control</code> , <code>Content-Type</code> , <code>Content-Disposition</code> , <code>Content-Encoding</code> , <code>Expires</code>	Cabeçalhos específicos para REST. Oferece suporte à correspondência exata e a <code>starts-with</code> .

Nome do elemento	Descrição
chave	O nome da chave carregada. Oferece suporte à correspondência exata e a starts-with.
success_action_redirect, redirect	O URL para o qual o cliente é redirecionado após um upload bem-sucedido. Oferece suporte à correspondência exata e a starts-with.
success_action_status	O código de status retornado ao cliente após o upload bem-sucedido se success_action_redirect não for especificado. Oferece suporte à correspondência exata.
x-amz-security-token	Token de segurança do Amazon DevPay. Cada solicitação que usa o Amazon DevPay requer dois campos de formulário x-amz-security-token: um para o token de produto e outro para o token de usuário. Consequentemente, os valores devem ser separados por vírgulas. Por exemplo, se o token de usuário é eW91dHVIZQ== e o token de produto for b0hnNVNKWVJIQTA=, defina a entrada da política para: { "x-amz-security-token": "eW91dHVIZQ==,b0hnNVNKWVJIQTA=" }.
Outros nomes de campos com o prefixo x-amz-meta-	Metadados especificados pelo usuário. Oferece suporte à correspondência exata e a starts-with.

#### Note

Se o seu toolkit traz campos adicionais (por exemplo, o Flash adiciona nome do arquivo), é necessário adicioná-los ao documento de política. Se essa funcionalidade puder ser controlada, adicione o prefixo x-ignore- ao campo para que o Amazon S3 ignore o recurso e para que futuras versões não sejam afetadas.

#### Correspondência de condição

A tabela a seguir descreve os tipos de correspondência de condição. Embora seja necessário especificar uma condição para cada campo especificado no formulário, é possível criar critérios de correspondência mais complexos especificando várias condições para um campo.

Condição	Descrição
Correspondências exatas	Correspondências exatas verificam se os campos correspondem a valores específicos. Este exemplo indica que a ACL deve ser definida como pública para leitura:  <pre>{ "acl": "public-read" }</pre> <p>Este exemplo é uma forma alternativa para indicar que a ACL deve ser definida como pública para leitura:</p> <pre>[ "eq", "\$acl", "public-read" ]</pre>

Condição	Descrição
Inicia com	Se o valor deve iniciar com um certo valor, use starts-with. Este exemplo indica que a chave deve iniciar com user/betty: <pre>[ "starts-with", "\$key", "user/betty/" ]</pre>
Corresponder qualquer conteúdo	Para configurar a política para permitir qualquer conteúdo em um campo, use starts-with com um valor vazio. Este exemplo permite qualquer success_action_redirect: <pre>[ "starts-with", "\$success_action_redirect", "" ]</pre>
Especificando intervalos	Para os campos que aceitam intervalos, separe os limites superior e inferior do intervalo com uma vírgula. Este exemplo permite um tamanho de arquivo entre 1 e 10 megabytes: <pre>[ "content-length-range", 1048579, 10485760 ]</pre>

## Caracteres de escape

A tabela a seguir descreve os caracteres de escape dentro de um documento de política.

Sequência de escape	Descrição
\\	Barra invertida
\\$	Sinal de dólar
\b	Apagar
\f	Feed do formulário
\n	Nova linha
\r	Carriage return
\t	Guia horizontal
\v	Guia vertical
\uXXXX	Todos os caracteres do Unicode

## Criar uma assinatura

Etapa	Descrição
1	Codifique a política usando UTF-8.
2	Codifique os bytes UTF-8 usando Base64.
3	Assine a política com a chave de acesso secreta usando HMAC SHA-1.
4	Codifique a assinatura SHA-1 usando Base64.

Para obter informações gerais sobre autenticação, consulte [Usar o acesso de autenticação](#).

## Redirecionamento

Esta seção descreve como manipular redirecionamentos.

### Redirecionamento geral

Após a conclusão da solicitação POST, o usuário é redirecionado para o local especificado no campo `success_action_redirect`. Se o Amazon S3 não for capaz de interpretar a URL, o campo `success_action_redirect` será ignorado.

Se `success_action_redirect` não for especificado, o Amazon S3 retornará o tipo de documento vazio especificado no campo `success_action_status`.

Se a solicitação POST falhar, o Amazon S3 exibirá um erro e não fará o redirecionamento.

### Redirecionamento pré-upload

Se o bucket foi criado usando <CreateBucketConfiguration>, os usuários finais poderão exigir um redirecionamento. Se isso ocorrer, alguns navegadores podem manipular o redirecionamento de maneira incorreta. Isso é relativamente raro, mas é mais provável que ocorra logo após a criação do bucket.

## Exemplos de uploads (AWS Signature versão 2)

### Tópicos

- [Upload de arquivo \(p. 677\)](#)
- [Upload de área de texto \(p. 680\)](#)

### Note

A autenticação de solicitação discutida nesta seção é baseada no Signature versão 2 da AWS, um protocolo para autenticar solicitações de entrada da API para serviços da AWS.

Agora, o Amazon S3 oferece suporte para o Signature versão 4, um protocolo para autenticação de solicitações de entrada da API para serviços da AWS, em todas as regiões da AWS. Neste momento, as regiões da AWS criadas antes de 30 de janeiro de 2014 continuarão oferecendo suporte ao protocolo anterior, Signature versão 2. Todas as regiões novas a partir de 30 de janeiro de 2014 oferecerão suporte apenas ao Signature versão 4. Portanto, todas as solicitações para essas regiões devem ser feitas com o Signature versão 4. Para obter mais informações, consulte [Exemplos: upload baseado no navegador usando HTTP POST \(usando o AWS Signature versão 4\)](#) no Amazon Simple Storage Service API Reference.

## Upload de arquivo

Este exemplo mostra o processo completo para criação de uma política e um formulário que pode ser usado para carregar um arquivo anexo.

### Criação de política e formulário

A política a seguir oferece suporte a uploads no Amazon S3 para o bucket `johnsmith`.

```
{ "expiration": "2007-12-01T12:00:00.000Z",
  "conditions": [
    {"bucket": "johnsmith"},
    ["starts-with", "$key", "user/eric/"],
    {"acl": "public-read"},
```

```
{ "success_action_redirect": "http://johnsmith.s3.amazonaws.com/  
successful_upload.html"},  
  [ "starts-with", "$Content-Type", "image/" ],  
  { "x-amz-meta-uuid": "14365123651274" },  
  [ "starts-with", "$x-amz-meta-tag", "" ]  
]  
}
```

Esta política requer o seguinte:

- O upload deve ocorrer antes das 12:00 UTC em 1º de dezembro de 2007.
- O conteúdo deve ser carregado para o bucket johnsmith.
- A chave deve começar com "user/eric/".
- A ACL está definida para leitura pública.
- O success\_action\_redirect está definido como http://johnsmith.s3.amazonaws.com/successful\_upload.html.
- O objeto é um arquivo de imagem.
- A tag x-amz-meta-uuid deve ser definida como 14365123651274.
- A tag x-amz-meta-tag pode conter qualquer valor.

Veja a seguir uma versão codificada em Base64 dessa política.

```
eyAizXhwaxJhdGlvbiI6IClEyMDA3LTEyLTAXVDEyOjAwOjAwMFoiLAogICJjb25kaXRpb25zIjogWwogICAgeyJidWNrZXQiOiA
```

Crie uma assinatura usando suas credenciais. Por exemplo 0RavWzkygo6QX9caELEqKi9kDbU= é a assinatura para o documento de política anterior.

O formulário a seguir oferece suporte a uma solicitação POST para o bucket johnsmith.net que usa essa política.

```
<html>  
  <head>  
    ...  
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />  
    ...  
  </head>  
  <body>  
    ...  
    <form action="http://johnsmith.s3.amazonaws.com/" method="post" enctype="multipart/form-data">  
      Key to upload: <input type="input" name="key" value="user/eric/" /><br />  
      <input type="hidden" name="acl" value="public-read" />  
      <input type="hidden" name="success_action_redirect" value="http://  
johnsmith.s3.amazonaws.com/successful_upload.html" />  
      Content-Type: <input type="input" name="Content-Type" value="image/jpeg" /><br />  
      <input type="hidden" name="x-amz-meta-uuid" value="14365123651274" />  
      Tags for File: <input type="input" name="x-amz-meta-tag" value="" /><br />  
      <input type="hidden" name="AWSAccessKeyId" value="AKIAIOSFODNN7EXAMPLE" />  
      <input type="hidden" name="Policy" value="POLICY" />  
      <input type="hidden" name="Signature" value="SIGNATURE" />  
      File: <input type="file" name="file" /> <br />  
      <!-- The elements after this will be ignored -->  
      <input type="submit" name="submit" value="Upload to Amazon S3" />  
    </form>  
    ...
```

```
</html>
```

## Solicitação de exemplo

Essa solicitação pressupõe que a imagem carregada tem 117.108 bytes; os dados da imagem não estão inclusos.

```
POST / HTTP/1.1
Host: johnsmith.s3.amazonaws.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.10) Gecko/20071115
Firefox/2.0.0.10
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/
plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Content-Type: multipart/form-data; boundary=9431149156168
Content-Length: 118698

--9431149156168
Content-Disposition: form-data; name="key"

user/eric/MyPicture.jpg
--9431149156168
Content-Disposition: form-data; name="acl"

public-read
--9431149156168
Content-Disposition: form-data; name="success_action_redirect"

http://johnsmith.s3.amazonaws.com/successful_upload.html
--9431149156168
Content-Disposition: form-data; name="Content-Type"

image/jpeg
--9431149156168
Content-Disposition: form-data; name="x-amz-meta-uuid"

14365123651274
--9431149156168
Content-Disposition: form-data; name="x-amz-meta-tag"

Some,Tag,For,Picture
--9431149156168
Content-Disposition: form-data; name="AWSAccessKeyId"

AKIAIOSFODNN7EXAMPLE
--9431149156168
Content-Disposition: form-data; name="Policy"

eyAiZXhwaXJhdGlvbiI6ICl4MDA3LTEyLTExVDEyOjAwOjAwLjAwMFOiLAogICJjb25kaXRpb25zIjogWwogICAgeyJidWNrZXQioiA
--9431149156168
Content-Disposition: form-data; name="Signature"

0RavWzkygo6QX9caELEqKi9kDbU=
--9431149156168
Content-Disposition: form-data; name="file"; filename="MyFilename.jpg"
Content-Type: image/jpeg

...file content...
--9431149156168
Content-Disposition: form-data; name="submit"
```

```
Upload to Amazon S3
--9431149156168--
```

## Resposta do exemplo

```
HTTP/1.1 303 Redirect
x-amz-request-id: 1AEE782442F35865
x-amz-id-2: cxzFLJRatFHy+NGtaDFRR8YvI9BHmgLxjvJzNiGGICARZ/mVXHj7T+qQKhdpzHFh
Content-Type: application/xml
Date: Wed, 14 Nov 2007 21:21:33 GMT
Connection: close
Location: http://johnsmith.s3.amazonaws.com/successful_upload.html?
bucket=johnsmith&key=user/eric/
MyPicture.jpg&etag=";39d459dfbc0faabbb5e179358dfb94c3"
Server: AmazonS3
```

## Upload de área de texto

### Tópicos

- [Criação de política e formulário \(p. 680\)](#)
- [Solicitação de exemplo \(p. 681\)](#)
- [Resposta do exemplo \(p. 682\)](#)

O exemplo a seguir mostra o processo completo para criação de uma política e um formulário para carregar uma área de texto. Fazer upload de uma área de texto é útil para o envio de conteúdo criado pelo usuário, como postagens de um blog.

### Criação de política e formulário

A política a seguir oferece suporte a uploads de área de texto no Amazon S3 para o bucket johnsmith.

```
{
  "expiration": "2007-12-01T12:00:00.000Z",
  "conditions": [
    {"bucket": "johnsmith"},
    ["starts-with", "$key", "user/eric/"],
    {"acl": "public-read"},
    {"success_action_redirect": "http://johnsmith.s3.amazonaws.com/new_post.html"},
    ["eq", "$Content-Type", "text/html"],
    {"x-amz-meta-uuid": "14365123651274"},
    ["starts-with", "$x-amz-meta-tag", ""]
  ]
}
```

Esta política requer o seguinte:

- O upload deve ocorrer antes das 12:00 GMT em 1º de dezembro de 2007.
- O conteúdo deve ser carregado para o bucket johnsmith.
- A chave deve começar com "user/eric/".
- A ACL está definida para leitura pública.
- O success\_action\_redirect está definido como [http://johnsmith.s3.amazonaws.com/new\\_post.html](http://johnsmith.s3.amazonaws.com/new_post.html).
- O objeto é texto HTML.
- A tag x-amz-meta-uuid deve ser definida como 14365123651274.
- A tag x-amz-meta-tag pode conter qualquer valor.

Veja a seguir uma versão codificada em Base64 dessa política.

```
eyAizXhwaJhdGlvbiI6ICiYMDA3LTEyLTAxVDEyOjAwOjAwLjAwMFOiLAogICJjb25kaX  
pb25zIjogWwogICAseyJidWNrZXQiOiAiam9obnNtaXRoiIn0sCiAgICBbInN0YXJ0cy13aXRoIiwgIiRrzXkiLC  
AiCAidXNlci9lcmljLy  
LAogICAseyJhY2wiOiAicHVibGlJLXJlYWQifSwKICAgIHsic3VjY2Vzc19hY3Rp  
b25fcmlVkaXJlY3QiOiaHR0cDovL2pvaG5zbWL  
C5zMy5hbWF6b25hd3MuY29tL25ld19wb3N0Lmh0bWwifSwKICAgIFsiZXEilCAiJENvbnRlb  
nQtVHlwZSIsICJ0ZXh0L2h0bWwiXSws  
CAgIHsieC1hbXotbWV0YS1dWlkIjogIjE0MzY1MTIzNjUxMjc0In0sCiAgICBbInN0YXJ0cy13aXRoIiwgIiR4LWFtei  
1tzXrhLXRh  
IsICiixQogIF0KfQo=
```

Crie uma assinatura usando suas credenciais. Por exemplo, qA7FWXKq6VvU68lI9KdveT1cWgF= é a assinatura para o documento de política anterior.

O formulário a seguir oferece suporte a uma solicitação POST para o bucket johnsmith.net que usa essa política.

```
<html>  
  <head>  
    ...  
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />  
    ...  
  </head>  
  <body>  
    ...  
    <form action="http://johnsmith.s3.amazonaws.com/" method="post" enctype="multipart/form-data">  
      Key to upload: <input type="input" name="key" value="user/eric/" /><br />  
      <input type="hidden" name="acl" value="public-read" />  
      <input type="hidden" name="success_action_redirect" value="http://  
johnsmith.s3.amazonaws.com/new_post.html" />  
      <input type="hidden" name="Content-Type" value="text/html" />  
      <input type="hidden" name="x-amz-meta-uuid" value="14365123651274" />  
      Tags for File: <input type="input" name="x-amz-meta-tag" value="" /><br />  
      <input type="hidden" name="AWSAccessKeyId" value="AKIAIOSFODNN7EXAMPLE" />  
      <input type="hidden" name="Policy" value="POLICY" />  
      <input type="hidden" name="Signature" value="SIGNATURE" />  
      Entry: <textarea name="file" cols="60" rows="10">  
  
      Your blog post goes here.  
  
      </textarea><br />  
      <!-- The elements after this will be ignored -->  
      <input type="submit" name="submit" value="Upload to Amazon S3" />  
    </form>  
    ...  
  </html>
```

## Solicitação de exemplo

Essa solicitação pressupõe que a imagem carregada tem 117.108 bytes; os dados da imagem não estão inclusos.

```
POST / HTTP/1.1  
Host: johnsmith.s3.amazonaws.com  
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.10) Gecko/20071115  
Firefox/2.0.0.10  
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/  
plain;q=0.8,image/png,*/*;q=0.5  
Accept-Language: en-us,en;q=0.5  
Accept-Encoding: gzip,deflate  
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7  
Keep-Alive: 300  
Connection: keep-alive  
Content-Type: multipart/form-data; boundary=178521717625888
```

```
Content-Length: 118635
--178521717625888
Content-Disposition: form-data; name="key"
ser/eric/NewEntry.html
--178521717625888
Content-Disposition: form-data; name="acl"
public-read
--178521717625888
Content-Disposition: form-data; name="success_action_redirect"
http://johnsmith.s3.amazonaws.com/new_post.html
--178521717625888
Content-Disposition: form-data; name="Content-Type"
text/html
--178521717625888
Content-Disposition: form-data; name="x-amz-meta-uuid"
14365123651274
--178521717625888
Content-Disposition: form-data; name="x-amz-meta-tag"
Interesting Post
--178521717625888
Content-Disposition: form-data; name="AWSAccessKeyId"
AKIAIOSFODNN7EXAMPLE
--178521717625888
Content-Disposition: form-data; name="Policy"
eyAizXhwaXJhdGlvbiI6IClYMDA3LTEyLTAxVDEyOjAwOjAwLjAwMFoiLAogICJjb25kaXRpb25zIjogWwogICAgeyJidWNrZXQioiA
--178521717625888
Content-Disposition: form-data; name="Signature"
qA7FWXKq6VvU68lI9KdveT1cWgF=
--178521717625888
Content-Disposition: form-data; name="file"
...content goes here...
--178521717625888
Content-Disposition: form-data; name="submit"
Upload to Amazon S3
--178521717625888--
```

## Resposta do exemplo

```
HTTP/1.1 303 Redirect
x-amz-request-id: 1AEE782442F35865
x-amz-id-2: cxzFLJRatFHy+NGtaDFRR8Yvi9BHmgLxjvJzNiGGICARZ/mVXHj7T+qQKhdpzHFh
Content-Type: application/xml
Date: Wed, 14 Nov 2007 21:21:33 GMT
Connection: close
Location: http://johnsmith.s3.amazonaws.com/new_post.html?bucket=johnsmith&key=user/eric/NewEntry.html&etag=40c3271af26b7f1672e41b8a274d28d4
Server: AmazonS3
```

## POST com Adobe Flash

Esta seção descreve como usar o POST com o Adobe Flash.

## Segurança do Adobe Flash Player

Por padrão, o modelo de segurança do Adobe Flash Player proíbe que os Adobe Flash Players realizem conexões de rede com servidores fora do domínio que serve o arquivo SWF.

Para substituir o padrão, é necessário carregar um arquivo crossdomain.xml de leitura pública no bucket que aceitará uploads POST. Veja a seguir um exemplo de arquivo crossdomain.xml.

```
<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy SYSTEM
"http://www.macromedia.com/xml/dtds/cross-domain-policy.dtd">
<cross-domain-policy>
<allow-access-from domain="*" secure="false" />
</cross-domain-policy>
```

### Note

Para obter mais informações sobre o modelo de segurança do Adobe Flash, acesse o site da Adobe.

Adicionar o arquivo crossdomain.xml ao bucket permite que qualquer Adobe Flash Player se conecte ao arquivo crossdomain.xml dentro do bucket; no entanto, isso não concede acesso ao bucket do Amazon S3 em si.

## Considerações do Adobe Flash

A API FileReference no Adobe Flash adiciona o campo de formulário `Filename` à solicitação POST. Ao compilar aplicativos do Adobe Flash que fazem upload no Amazon S3 usando a ação da API `FileReference`, inclua a seguinte condição na política:

```
[ 'starts-with', '$Filename', '' ]
```

Algumas versões do Adobe Flash Player não lidam muito bem com respostas HTTP que têm um corpo vazio. Para configurar o POST para retornar uma resposta que não tenha o corpo vazio, defina `success_action_status` como 201. O Amazon S3 retornará um documento XML com o código de status 201. Para obter informações sobre o conteúdo do documento XML, consulte [Objeto POST](#). Para obter informações campos de formulário, consulte [Campos do formulário HTML \(p. 671\)](#).

# Recursos da Amazon S3

A tabela a seguir lista os recursos relacionados que serão úteis à medida que você utilizar este serviço.

Recurso	Descrição
<a href="#">Guia de conceitos básicos do Amazon Simple Storage Service</a>	O Guia de conceitos básicos fornece um tutorial rápido do serviço com base em um caso de uso simples.
<a href="#">Amazon Simple Storage Service API Reference</a>	A referência da API descreve as operações do Amazon S3 em detalhes.
<a href="#">Perguntas frequentes técnicas sobre o Amazon S3</a>	As perguntas frequentes abordam as dúvidas mais comuns entre os desenvolvedores deste produto.
<a href="#">Centro de recursos do desenvolvedor da AWS</a>	Uma central de ponto de partida para localizar documentação, exemplos de código, notas de release e outras informações para ajudar você a desenvolver aplicativos inovadores com a AWS.
<a href="#">Console de Gerenciamento da AWS</a>	O console permite que você realize a maioria das funções do Amazon S3 sem programação.
<a href="https://forums.aws.amazon.com/">https://forums.aws.amazon.com/</a>	Um fórum comunitário para que os desenvolvedores discutam questões técnicas relacionadas ao AWS.
<a href="#">AWS Support Center</a>	A página inicial para obter suporte técnico AWS, incluindo o acesso aos nossos Fóruns de desenvolvedores, Perguntas frequentes técnicas, Página de status do serviço e Premium Support.
<a href="#">AWS Premium Support</a>	A principal página da web para informações sobre o AWS Premium Support, um canal de suporte de resposta rápida e com atendimento individual, a fim de ajudá-lo a desenvolver e executar aplicativos nos AWS Infrastructure Services.
<a href="#">Informações sobre produtos do Amazon S3</a>	A principal página da web para obter informações sobre o Amazon S3.
<a href="#">Entre em contato conosco</a>	Um ponto central de contato para consultas relativas a faturamento da AWS, conta, eventos, abuso etc.
<a href="#">Condições de uso</a>	Informações detalhadas sobre o uso de direitos autorais e marca registrada na Amazon.com e outros tópicos.

# Referência SQL para o Amazon S3 Select e o Glacier Select

Essa referência contém uma descrição de elementos (SQL) de linguagem de consulta estruturada que são compatíveis com o Amazon S3 Select e o Glacier Select.

## Tópicos

- [Comando SELECT \(p. 685\)](#)
- [Tipos de dados \(p. 692\)](#)
- [Operadores \(p. 693\)](#)
- [Palavras-chave reservadas \(p. 694\)](#)
- [Funções SQL \(p. 698\)](#)

## Comando SELECT

O Amazon S3 Select e o Glacier Select são compatíveis somente com o comando SQL `SELECT`. As seguintes cláusulas padrão ANSI são compatíveis com `SELECT`:

- `SELECT` lista
- Cláusula `FROM`
- Cláusula `WHERE`
- Cláusula `LIMIT` (apenas Amazon S3 Select)

### Note

No momento, as consultas do Amazon S3 Select e do Glacier Select não oferecem suporte a subconsultas ou junções.

## Lista SELECT

A lista `SELECT` nomeia as colunas, as funções e as expressões que a consulta deve retornar. A lista representa o resultado da consulta.

```
SELECT *
SELECT projection [ AS column_alias | column_alias ] [, ...]
```

O primeiro formulário com \* (asterisco) retorna todas as linhas que passaram na cláusula `WHERE`, da maneira como estão. O segundo formulário cria uma linha com expressões escalares de saída definidas pelo usuário `projection` para cada coluna.

## Cláusula FROM

O Amazon S3 Select e o Glacier Select são compatíveis com os seguintes formatos da cláusula `FROM`:

```
FROM table_name
FROM table_name alias
FROM table_name AS alias
```

Em que `table_name` é um `S3Object` (do Amazon S3 Select) ou `ARCHIVE` ou `OBJECT` (para o Glacier Select) referindo-se ao arquivo que está sendo consultado. Os usuários provenientes de bancos de dados relacionais tradicionais podem pensar nisso como um esquema de banco de dados que contém várias visualizações em uma tabela.

Seguindo o SQL padrão, a cláusula `FROM` cria linhas filtradas na cláusula `WHERE` e projetadas na lista `SELECT`.

Para objetos JSON armazenados no Amazon S3 Select, você também pode usar as seguintes formas da cláusula `FROM`:

```
FROM S3Object[*].path
FROM S3Object[*].path alias
FROM S3Object[*].path AS alias
```

Com essa forma da cláusula `FROM`, você pode selecionar entre matrizes ou objetos em um objeto JSON. Você pode especificar esse `path` usando uma das formas a seguir:

- Por nome (em um objeto): `.name` ou `[ 'name' ]`
- Por índice (em uma matriz): `[ index ]`
- Por curinga (em um objeto): `.*`
- Por curinga (em uma matriz): `[ * ]`

#### Note

- Essa forma da cláusula `FROM` funciona apenas com objetos JSON.
- Curingas sempre emitem pelo menos um registro. Se não houver correspondência com nenhum registro, o Amazon S3 Select emitirá o valor `MISSING`. Durante a serialização de saída (após a conclusão da consulta), o Amazon S3 Select substituirá os valores `MISSING` por registros vazios.
- Funções agregadas (`AVG`, `COUNT`, `MAX`, `MIN`, and `SUM`) ignoram valores `MISSING`.
- Se você não fornecer um alias ao usar um curinga, poderá consultar a linha usando o último elemento do caminho. Por exemplo, você pode selecionar todos os preços em uma lista de livros usando a consulta `SELECT price FROM S3Object[*].books[*].price`. Se o caminho terminar com um curinga em vez de um nome, você poderá usar o `valor _1` para consultar a linha. Por exemplo, em vez de `SELECT price FROM S3Object[*].books[*].price`, você pode usar a consulta `SELECT _1.price FROM S3Object[*].books[*]`.
- O Amazon S3 Select sempre trata um documento JSON como uma matriz de valores no nível da raiz. Dessa forma, mesmo se o objeto JSON que você estiver consultando tiver apenas um elemento raiz, a cláusula `FROM` deverá começar com `S3Object[*]`. No entanto, por razões de compatibilidade, o Amazon S3 Select permite omitir o curinga caso você não inclua um caminho. Dessa forma, a cláusula completa `FROM S3Object` é equivalente a `FROM S3Object[*] as S3Object`. Se você incluir um caminho, também deverá usar o curinga. Portanto, `FROM S3Object` e `FROM S3Object[*].path` são cláusulas válidas, mas `FROM S3Object.path` não.

#### Example

Exemplos:

Exemplo 1

Este exemplo mostra resultados usando o seguinte conjunto de dados e consulta:

```
{  
    "Rules": [  
        {"id": "id-1", "condition": "x < 20"},  
        {"condition": "y > x"},  
        {"id": "id-2", "condition": "z = DEBUG"}  
    ]  
},  
{  
    "created": "June 27",  
    "modified": "July 6"  
}
```

```
SELECT id FROM S3Object[*].Rules[*].id
```

```
{"id":"id-1"},  
{},  
{"id":"id-2"},  
{}
```

O Amazon S3 Select produz cada resultado pelos seguintes motivos:

- {"id":"id-1"} — S3Object[0].Rules[0].id produziu uma correspondência.
- {} — S3Object[0].Rules[1].id não correspondeu a um registro, portanto, o Amazon S3 Select emitiu MISSING que, em seguida, foi alterado para um registro vazio durante a serialização de saída e retornou.
- {"id":"id-2"} — S3Object[0].Rules[2].id produziu uma correspondência.
- {} — S3Object[1] não teve correspondência em Rules, portanto, o Amazon S3 Select emitiu MISSING que, em seguida, foi alterado para um registro vazio durante a serialização de saída e retornou.

Se você não quiser que o Amazon S3 Select retorne registros vazios quando ele não encontrar uma correspondência, você poderá testar o valor MISSING. A consulta a seguir retorna os mesmos resultados que a consulta anterior, mas com os valores vazios omitidos:

```
SELECT id FROM S3Object[*].Rules[*].id WHERE id IS NOT MISSING
```

```
{"id":"id-1"},  
{"id":"id-2"}
```

### Exemplo 2

Este exemplo mostra resultados usando o seguinte conjunto de dados e consultas:

```
{  
    "created": "936864000",  
    "dir_name": "important_docs",  
    "files": [  
        {  
            "name": ".  
        },  
        {  
            "name": ".."  
        },  
        {  
            "name": ".aws"  
        },  
        {  
            "name": "downloads"  
        }  
    ]  
}
```

```
        }
    ],
    "owner": "AWS S3"
},
{
    "created": "936864000",
    "dir_name": "other_docs",
    "files": [
        {
            "name": "."
        },
        {
            "name": ".."
        },
        {
            "name": "my stuff"
        },
        {
            "name": "backup"
        }
    ],
    "owner": "User"
}
```

```
SELECT d.dir_name, d.files FROM S3Object[*] d
```

```
{
    "dir_name": "important_docs",
    "files": [
        {
            "name": "."
        },
        {
            "name": ".."
        },
        {
            "name": ".aws"
        },
        {
            "name": "downloads"
        }
    ]
},
{
    "dir_name": "other_docs",
    "files": [
        {
            "name": "."
        },
        {
            "name": ".."
        },
        {
            "name": "my stuff"
        },
        {
            "name": "backup"
        }
    ]
}
```

```
SELECT _1.dir_name, _1.owner FROM S3Object[*]
```

```
{  
    "dir_name": "important_docs",  
    "owner": "AWS S3"  
},  
{  
    "dir_name": "other_docs",  
    "owner": "User"  
}
```

## Cláusula WHERE

A cláusula WHERE segue esta sintaxe:

```
WHERE condition
```

A cláusula WHERE filtra as linhas com base na condição. Uma condição é uma expressão com um valor booliano. Somente linhas para as quais a condição é avaliada como TRUE são retornadas no resultado.

## Cláusula LIMIT (apenas Amazon S3 Select)

A cláusula LIMIT segue esta sintaxe:

```
LIMIT number
```

A cláusula LIMIT limita o número de registros que você deseja que a consulta retorne com base no número.

### Note

O Glacier Select não é compatível com a cláusula LIMIT.

## Acesso de atributo

As cláusulas SELECT e WHERE podem se referir a dados de registro usando um dos métodos nas seções a seguir, dependendo se o arquivo que está sendo consultado está no formato CSV ou JSON.

### CSV

- Números da coluna – Você pode se referir à N<sup>a</sup> coluna de uma linha com o nome \_N, em que N é a posição da coluna. A contagem da posição começa em 1. Por exemplo, a primeira coluna é denominada \_1 e a segunda coluna é denominada \_2.

Você pode se referir a uma coluna como \_N ou alias.\_N. Por exemplo, \_2 e myAlias.\_2 são maneiras válidas de fazer referência a uma coluna na lista SELECT e na cláusula WHERE.

- Cabeçalhos da coluna – Para objetos no formato CSV que possuem uma linha de cabeçalho, os cabeçalhos estão disponíveis para a lista SELECT e a cláusula WHERE. Especificamente, como no SQL tradicional, nas expressões de cláusula SELECT e WHERE, você pode consultar as colunas por alias.column\_name ou column\_name.

### JSON (apenas Amazon S3 Select)

- Documento – Você pode acessar os campos do documentos JSON como alias.name. Os campos aninhados também podem ser acessados, por exemplo, alias.name1.name2.name3

- Lista – Você pode acessar elementos em uma lista JSON usando índices baseados em zero com o operador [ ]. Por exemplo, você pode acessar o segundo elemento de uma lista como alias[1]. Acessar elementos de lista pode ser combinado com campos como alias.name1.name2[1].name3.
- Exemplos: considere esse objeto JSON como um exemplo de conjunto de dados:

```
{"name": "Susan Smith",
"org": "engineering",
"projects":
[
    {
        "project_name": "project1", "completed": false},
        {"project_name": "project2", "completed": true}
    ]
}
```

#### Exemplo 1

A consulta a seguir retorna estes resultados:

```
Select s.name from S3Object s
```

```
{"name": "Susan Smith"}
```

#### Exemplo 2

A consulta a seguir retorna estes resultados:

```
Select s.projects[0].project_name from S3Object s
```

```
{"project_name": "project1"}
```

## Diferenciação de letras maiúsculas e minúsculas de cabeçalho/nomes de atributo

Com o Amazon S3 Select e o Glacier Select, você pode usar aspas duplas para indicar que cabeçalhos de coluna (para objetos CSV) e atributos (para objetos JSON) fazem diferenciação entre letras maiúsculas e minúsculas. Sem as aspas duplas, os cabeçalhos/atributos de objeto não fazem diferenciação entre letras maiúsculas e minúsculas. Um erro ocorre em casos de ambiguidade.

Os exemplos a seguir são 1) objetos do Amazon S3 ou do Glacier no formato CSV com os cabeçalhos de coluna especificados e com `FileHeaderInfo` definido como “Usar” para a solicitação de consulta ou 2) objetos do Amazon S3 no formato JSON com os atributos especificados.

Exemplo 1: O objeto que está sendo consultado tem o cabeçalho/atributo “NAME”.

- A expressão a seguir retorna com êxito valores do objeto (sem aspas: não diferenciando entre letras maiúsculas e minúsculas):

```
SELECT s.name from S3Object s
```

- Os seguintes resultados de expressão em um erro 400 `MissingHeaderName` (aspas: diferenciação entre letras maiúsculas e minúsculas):

```
SELECT s."name" from S3Object s
```

Exemplo 2: O objeto do Amazon S3 que está sendo consultado tem um cabeçalho/atributo com “NAME” e outro cabeçalho/atributo com “name”.

- A seguinte expressão resulta em um erro 400 AmbiguousFieldName (sem aspas: sem diferenciação entre letras maiúsculas e minúsculas, mas há duas correspondências):

```
SELECT s.name from S3Object s
```

- A expressão a seguir retorna com êxito valores do objeto (aspas: diferenciação entre letras maiúsculas e minúsculas, portanto, resolve a ambiguidade):

```
SELECT s."NAME" from S3Object s
```

## Usar palavras-chave reservadas como termos definidos pelo usuário

O Amazon S3 Select e o Glacier Select possuem um conjunto de palavras-chave reservadas que são necessárias para executar as expressões SQL usadas para consultar o conteúdo do objeto. As palavras-chave reservadas incluem nomes de função, tipos de dados, operadores, e assim por diante. Em alguns casos, os termos definidos pelo usuário como os cabeçalhos de coluna (para arquivos CSV) ou os atributos (para objeto JSON) podem entrar em conflito com uma palavra-chave reservada. Quando isso ocorrer, é necessário usar as aspas duplas para indicar que você está usando intencionalmente um termo definido pelo usuário que entra em conflito com uma palavra-chave reservada. Caso contrário, ocorrerá um erro de análise 400.

Para obter a lista completa de palavras-chave, consulte [Palavras-chave reservadas \(p. 694\)](#).

O exemplo a seguir é 1) um objeto do Amazon S3 ou do Glacier no formato CSV com os cabeçalhos de coluna especificados e com `FileHeaderInfo` definido como “Usar” para a solicitação de consulta ou 2) um objeto do Amazon S3 no formato JSON com os atributos especificados.

Exemplo: o objeto que está sendo consultado tem o cabeçalho/atributo nomeado como “CAST”, que é uma palavra-chave reservada.

- A expressão a seguir retorna com êxito valores do objeto (aspas: usar cabeçalho/atributo definido pelo usuário):

```
SELECT s."CAST" from S3Object s
```

- Os seguintes resultados de expressão resultam em um erro de análise 400 (sem aspas: entram em conflito com palavra-chave reservada):

```
SELECT s.CAST from S3Object s
```

## Expressões escalares

Na cláusula WHERE e na lista SELECT, você tem expressões escalares SQL, que são expressões que retornam valores escalares. Elas têm o seguinte formato:

- **literal**

Um literal SQL.

- **column\_reference**

Uma referência a uma coluna no formato `column_name` ou `alias.column_name`.

- **unary\_op expression**

Em que `unary_op` é um operador unário SQL.

- **expression binary\_op expression**

Em que `binary_op` é um operador binário SQL.

- **func\_name**

Em que `func_name` é o nome de uma função escalar a ser invocada.

- **expression [ NOT ] BETWEEN expression AND expression**

- **expression LIKE expression [ ESCAPE expression ]**

## Tipos de dados

O Amazon S3 Select e o Glacier Select são compatíveis com vários tipos de dados primitivos.

## Conversões de tipo de dados

A regra geral é seguir a função `CAST` se definida. Se `CAST` não estiver definido, todos os dados de entrada serão tratados como uma string. Ele deve ser convertido em tipos de dados relevantes quando necessário.

Para obter mais informações sobre a função `CAST`, consulte [CAST \(p. 700\)](#).

## Tipos de dados compatíveis

O Amazon S3 Select e o Glacier Select são compatíveis com o conjunto de tipos de dados primitivos a seguir.

Nome	Descrição	Exemplos
bool	TRUE ou FALSE	FALSE
int, inteiro	Número inteiro assinado de 8 bytes no intervalo de -9.223.372.036.854.775.808 a 9.223.372.036.854.775.807.	100000
string	String de tamanho variável codificada por UTF8. O limite padrão é um caractere. O limite máximo de caracteres é 2.147.483.647.	'xyz'
flutuante	Número de ponto flutuante de 8 bytes.	CAST(0.456 AS FLOAT)
decimal, numérico	Número de base 10, com precisão máxima de 38 (ou seja, a quantidade máxima de dígitos significativos) e com escala no intervalo de $-2^{31}$ a $2^{31}-1$ (ou seja, o expoente de base 10).	123.456
timestamp	Os time stamps representam um momento específico, sempre incluem um deslocamento local e são capazes de oferecer precisão arbitrária.	CAST('2007-04-05T14:30Z' AS TIMESTAMP)

Nome	Descrição	Exemplos
	No formato de texto, os time stamps seguem a <a href="#">nota W3C sobre formatos de data e hora</a> , mas devem terminar com o literal "T", se não for pelo menos a precisão de dia inteiro. As frações de segundos são permitidas, com pelo menos um dígito de precisão e um máximo ilimitado. Os deslocamentos de hora local podem ser representados como deslocamentos de hora:minuto em UTC ou como o literal "Z" para indicar uma hora local em UTC. Eles são necessários em time stamps com hora e não são permitidos em valores de data.	

## Operadores

O Amazon S3 Select e o Glacier Select são compatíveis com os operadores a seguir.

### Operadores lógicos

- AND
- NOT
- OR

### Operadores de comparação

- <
- >
- <=
- >=
- =
- <>
- !=
- BETWEEN
- IN – Por exemplo: IN ('a', 'b', 'c')

### Operadores de correspondência de padrões

- LIKE

### Operadores matemáticos

A adição, a subtração, a multiplicação, a divisão e o módulo são compatíveis.

- +
- -
- \*
- %

## Precedência do operador

A tabela a seguir mostra a precedência dos operadores em ordem decrescente.

Operador/ elemento	Capacidade de associação	Obrigatório
-	direita	menos unário
*, /, %	esquerda	multiplicação, divisão, módulo
+, -	esquerda	adição, subtração
IN		associação de conjunto
BETWEEN		contenção de intervalo
LIKE		correspondência de padrões de string
<>		menor que, maior que
=	direita	igualdade, atribuição
NOT	direita	negação lógica
AND	esquerda	conjunção lógica
OU	esquerda	disjunção lógica

## Palavras-chave reservadas

Veja abaixo a lista de palavras-chave reservadas do Amazon S3 Select e do Glacier Select. Estes incluem os nomes de função, tipos de dados, operadores, etc., necessários para executar as expressões SQL usadas para consultar o conteúdo do objeto.

```
absolute
action
add
all
allocate
alter
and
any
are
as
asc
assertion
at
authorization
avg
```

```
bag
begin
between
bit
bit_length
blob
bool
boolean
both
by
cascade
cascaded
case
cast
catalog
char
char_length
character
character_length
check
clob
close
coalesce
collate
collation
column
commit
connect
connection
constraint
constraints
continue
convert
corresponding
count
create
cross
current
current_date
current_time
current_timestamp
current_user
cursor
date
day
deallocate
dec
decimal
declare
default
deferrable
deferred
delete
desc
describe
descriptor
diagnostics
disconnect
distinct
domain
double
drop
else
end
end-exec
escape
```

```
except
exception
exec
execute
exists
external
extract
false
fetch
first
float
for
foreign
found
from
full
get
global
go
goto
grant
group
having
hour
identity
immediate
in
indicator
initially
inner
input
insensitive
insert
int
integer
intersect
interval
into
is
isolation
join
key
language
last
leading
left
level
like
limit
list
local
lower
match
max
min
minute
missing
module
month
names
national
natural
nchar
next
no
not
```

```
null
nullif
numeric
octet_length
of
on
only
open
option
or
order
outer
output
overlaps
pad
partial
pivot
position
precision
prepare
preserve
primary
prior
privileges
procedure
public
read
real
references
relative
restrict
revoke
right
rollback
rows
schema
scroll
second
section
select
session
session_user
set
sexp
size
smallint
some
space
sql
sqlcode
sqlerror
sqlstate
string
struct
substring
sum
symbol
system_user
table
temporary
then
time
timestamp
timezone_hour
timezone_minute
to
```

```
trailing
transaction
translate
translation
trim
true
tuple
union
unique
unknown
unpivot
update
upper
usage
user
using
value
values
varchar
varying
view
when
whenever
where
with
work
write
year
zone
```

## Funções SQL

O Amazon S3 Select e o Glacier Select são compatíveis com várias funções SQL.

### Tópicos

- [Funções agregadas \(apenas Amazon S3 Select\) \(p. 698\)](#)
- [Funções condicionais \(p. 699\)](#)
- [Funções da conversão \(p. 700\)](#)
- [Funções de data \(p. 701\)](#)
- [Funções de string \(p. 707\)](#)

## Funções agregadas (apenas Amazon S3 Select)

O Amazon S3 Select é compatível com as seguintes funções agregadas.

#### Note

O Glacier Select não é compatível com as funções agregadas.

Função	Tipo de argumento	Tipo de retorno
AVG(expression)	INT, FLOAT, DECIMAL	DECIMAL para um argumento INT, FLOAT para um argumento de ponto

Função	Tipo de argumento	Tipo de retorno
	–	flutuante, caso contrário, é igual ao tipo de dados do argumento.
COUNT	–	INT
MAX(expression)	INT, DECIMAL	O mesmo que o tipo de argumento.
MIN(expression)	INT, DECIMAL	O mesmo que o tipo de argumento.
SUM(expression)	INT, FLOAT, DOUBLE, DECIMAL	INT para argumento INT, FLOAT para um argumento de ponto flutuante, caso contrário, é igual ao tipo de dados do argumento.

## Funções condicionais

O Amazon S3 Select e o Glacier Select são compatíveis com as seguintes funções condicionais.

### Tópicos

- [COALESCE \(p. 699\)](#)
- [NULLIF \(p. 700\)](#)

## COALESCE

Avalia os argumentos na ordem e retorna o primeiro não desconhecido, ou seja, o primeiro que não for nulo ou ausente. Essa função não propaga nulos e ausentes.

### Sintaxe

```
COALESCE ( expression, expression, ... )
```

### Parâmetros

expressão

A expressão de destino na qual a função opera.

### Exemplos

```
COALESCE(1)
```

```
-- 1
```

```
COALESCE(null)          -- null
COALESCE(null, null)    -- null
COALESCE(missing)       -- null
COALESCE(missing, missing) -- null
COALESCE(1, null)        -- 1
COALESCE(null, null, 1)  -- 1
COALESCE(null, 'string') -- 'string'
COALESCE(missing, 1)     -- 1
```

## NULLIF

Dadas as duas expressões, retorna NULL se as duas forem avaliadas para o mesmo valor. Caso contrário, retorna o resultado da avaliação da primeira expressão.

### Sintaxe

```
NULLIF ( expression1, expression2 )
```

### Parâmetros

expression1, expression2

As expressões de destino nas quais a função opera.

### Exemplos

```
NULLIF(1, 1)          -- null
NULLIF(1, 2)          -- 1
NULLIF(1.0, 1)        -- null
NULLIF(1, '1')        -- 1
NULLIF([1], [1])      -- null
NULLIF(1, NULL)       -- 1
NULLIF(NULL, 1)        -- null
NULLIF(null, null)    -- null
NULLIF(missing, null)  -- null
NULLIF(missing, missing) -- null
```

## Funções da conversão

O Amazon S3 Select e o Glacier Select são compatíveis com as seguintes funções de conversão.

### Tópicos

- [CAST \(p. 700\)](#)

## CAST

A função CAST converte uma entidade, como uma expressão que retorna um único valor, de um tipo em outro.

### Sintaxe

```
CAST ( expression AS data_type )
```

## Parâmetros

### expressão

Uma combinação de um ou mais valores, operadores e funções SQL que retornam um valor.

### data\_type

O tipo de dados de destino, como `INT`, no qual a expressão será convertida. Para obter uma lista dos tipos de dados compatíveis, consulte [Tipos de dados \(p. 692\)](#).

## Exemplos

```
CAST('2007-04-05T14:30Z' AS TIMESTAMP)  
CAST(0.456 AS FLOAT)
```

## Funções de data

O Amazon S3 Select e o Glacier Select são compatíveis com as seguintes funções de data.

### Tópicos

- [DATE\\_ADD \(p. 701\)](#)
- [DATE\\_DIFF \(p. 702\)](#)
- [EXTRACT \(p. 703\)](#)
- [TO\\_STRING \(p. 703\)](#)
- [TO\\_TIMESTAMP \(p. 706\)](#)
- [UTCNOW \(p. 707\)](#)

## DATE\_ADD

Dada uma parte da data, uma quantidade e um time stamp, retorna um time stamp atualizado, alterando a parte da data pela quantidade.

### Sintaxe

```
DATE_ADD( date_part, quantity, timestamp )
```

## Parâmetros

### date\_part

Especifica que parte da data deve ser modificada. Pode ser uma das partes a seguir:

- year
- mês
- dia
- hora
- minuto
- segundos

quantity (quantidade)

O valor a ser aplicado a um time stamp atualizado. Os valores positivos para a quantidade são adicionados à date\_part do time stamp e os valores negativos são subtraídos.

timestamp

O time stamp de destino no qual a função opera.

## Exemplos

```
DATE_ADD(year, 5, `2010-01-01T`)
DATE_ADD(month, 1, `2010T`)
    necessary)
DATE_ADD(month, 13, `2010T`)
DATE_ADD(day, -1, `2017-01-10T`)
DATE_ADD(hour, 1, `2017T`)
DATE_ADD(hour, 1, `2017-01-02T03:04Z`)
DATE_ADD(minute, 1, `2017-01-02T03:04:05.006Z`)
DATE_ADD(second, 1, `2017-01-02T03:04:05.006Z`) -- 2015-01-01 (equivalent to 2015-01-01T)
-- 2010-02T (result will add precision as
-- 2011-02T
-- 2017-01-09 (equivalent to 2017-01-09T)
-- 2017-01-01T01:00-00:00
-- 2017-01-02T04:04Z
-- 2017-01-02T03:05:05.006Z
-- 2017-01-02T03:04:06.006Z
```

## DATE\_DIFF

Dada uma parte da data e dois time stamps, retorna a diferença nas partes da data. O valor de retorno é um inteiro negativo quando o valor date\_part do timestamp1 for maior que o valor date\_part do timestamp2. O valor de retorno é um inteiro positivo quando o valor date\_part do timestamp1 for menor que o valor date\_part do timestamp2.

### Sintaxe

```
DATE_DIFF( date_part, timestamp1, timestamp2 )
```

### Parâmetros

date\_part

Especifica que parte dos time stamps deve ser comparada. Para a definição de date\_part, consulte [DATE\\_ADD \(p. 701\)](#).

timestamp1

O primeiro time stamp a ser comparado.

timestamp2

O segundo time stamp a ser comparado.

## Exemplos

```
DATE_DIFF(year, `2010-01-01T`, `2011-01-01T`)
DATE_DIFF(year, `2010T`, `2010-05T`)
    2010-01-01T00:00:00.000Z) -- 1
-- 4 (2010T is equivalent to
DATE_DIFF(month, `2010T`, `2011T`)
DATE_DIFF(month, `2011T`, `2010T`)
DATE_DIFF(day, `2010-01-01T23:00T`, `2010-01-02T01:00T`)
    apart to be 1 day apart) -- 12
-- -12
-- 0 (need to be at least 24h
```

## EXTRACT

Dada uma parte da data e um time stamp, retorna o valor da parte da data do time stamp.

### Sintaxe

```
EXTRACT( date_part FROM timestamp )
```

### Parâmetros

date\_part

Especifica que parte dos time stamps deve ser extraída. Pode ser uma das partes a seguir:

- year
- mês
- dia
- hora
- minuto
- segundos
- timezone\_hour
- timezone\_minute

timestamp

O time stamp de destino no qual a função opera.

### Exemplos

```
EXTRACT(YEAR FROM `2010-01-01T`)  
EXTRACT(MONTH FROM `2010T`)  
2010-01-01T00:00:00.000Z) -- 2010  
-- 1 (equivalent to  
EXTRACT(MONTH FROM `2010-10T`)  
EXTRACT(HOUR FROM `2017-01-02T03:04:05+07:08`)  
EXTRACT(MINUTE FROM `2017-01-02T03:04:05+07:08`)  
EXTRACT(TIMEZONE_HOUR FROM `2017-01-02T03:04:05+07:08`)  
EXTRACT(TIMEZONE_MINUTE FROM `2017-01-02T03:04:05+07:08`)
```

## TO\_STRING

Dado um time stamp e um padrão de formato, retorna uma representação de string do time stamp no formato especificado.

### Sintaxe

```
TO_STRING ( timestamp time_format_pattern )
```

### Parâmetros

timestamp

O time stamp de destino no qual a função opera.

#### time\_format\_pattern

Uma string que possui as seguintes interpretações especiais de caracteres.

Formato	Exemplo	Descrição
yy	69	Ano com dois dígitos
y	1969	Ano com quatro dígitos
yyyy	1969	Ano com 4 dígitos preenchido com zeros
M	1	Mês do ano
MM	01	Mês do ano preenchido com zeros
MMM	Jan	Nome do ano referente ao mês abreviado
MMMM	January	Nome completo do mês do ano
MMMMM	J	Primeira letra do mês do ano (NOTA: inválido para uso com a função to_timestamp)
d	2	Dia do mês (1 a 31)
dd	02	Dia do mês preenchido com zeros (01 a 31)
a	AM	H do dia
h	3	Hora do dia (1 a 24)
hh	03	Hora do dia preenchida com zeros (01 a 24)
H	3	Hora do dia (0 a 23)
HH	03	Hora do dia preenchida com zeros (00 a 23)

Formato	Exemplo	Descrição
m	4	Minuto da hora (0 a 59)
mm	04	Minuto da hora preenchido com zeros (00 a 59)
s	5	Segundo do minuto (0 a 59)
ss	05	Segundo do minuto preenchido com zeros (00 a 59)
S	0	Fração de segundos (precisão: 0,1, intervalo: 0,0 a 0,9)
SS	6	Fração de segundos (precisão: 0,01, intervalo: 0,0 a 0,99)
SSS	60	Fração de segundos (precisão: 0,001, intervalo: 0,0 a 0,999)
...	...	...
SSSSSSSS	60000000	Fração de segundos (precisão máxima: 1 nanosegundo, intervalo: 0,0 a 0,999999999)
n	60000000	Nano de segundo
x	+07 or Z	Deslocamento em horas ou "Z" se o deslocamento for 0

Formato	Exemplo	Descrição
<b>XX or XXXX</b>	<b>+0700 or Z</b>	Deslocamento em horas e minutos ou "Z" se o deslocamento for 0
<b>XXX or XXXXX</b>	<b>+07:00 or Z</b>	Deslocamento em horas e minutos ou "Z" se o deslocamento for 0
<b>x</b>	<b>7</b>	Deslocamento em horas
<b>xx or xxxx</b>	<b>700</b>	Deslocamento em horas e minutos
<b>xxx or xxxxx</b>	<b>+07:00</b>	Deslocamento em horas e minutos

## Exemplos

```

TO_STRING(`1969-07-20T20:18Z`, 'MMMM d, y')           -- "July 20, 1969"
TO_STRING(`1969-07-20T20:18Z`, 'MMM d, yyyy')         -- "Jul 20, 1969"
TO_STRING(`1969-07-20T20:18Z`, 'M-d-yy')              -- "7-20-69"
TO_STRING(`1969-07-20T20:18Z`, 'MM-d-y')              -- "07-20-1969"
TO_STRING(`1969-07-20T20:18Z`, 'MMMM d, y h:m a')     -- "July 20, 1969 8:18 PM"
TO_STRING(`1969-07-20T20:18Z`, 'y-MM-dd''T''H:m:ssX')   -- "1969-07-20T20:18:00Z"
TO_STRING(`1969-07-20T20:18+08:00Z`, 'y-MM-dd''T''H:m:ssX') -- "1969-07-20T20:18:00Z"
TO_STRING(`1969-07-20T20:18+08:00`, 'y-MM-dd''T''H:m:ssXXXX') -- "1969-07-20T20:18:00+0800"
TO_STRING(`1969-07-20T20:18+08:00`, 'y-MM-dd''T''H:m:ssXXXX') -- "1969-07-20T20:18:00+08:00"

```

## TO\_TIMESTAMP

Dada uma string, converte-a em um time stamp. Esta é a operação inversa de TO\_STRING.

### Sintaxe

```
TO_TIMESTAMP ( string )
```

### Parâmetros

**string**

A string de destino na qual a função opera.

## Exemplos

```
TO_TIMESTAMP('2007T') -- `2007T`  
TO_TIMESTAMP('2007-02-23T12:14:33.079-08:00') -- `2007-02-23T12:14:33.079-08:00`
```

## UTCNOW

Retorna o tempo atual em UTC como um time stamp.

### Sintaxe

```
UTCNOW()
```

### Parâmetros

none

## Exemplos

```
UTCNOW() -- 2017-10-13T16:02:11.123Z
```

## Funções de string

O Amazon S3 Select e o Glacier Select são compatíveis com as funções de string a seguir.

### Tópicos

- [CHAR\\_LENGTH, CHARACTER\\_LENGTH \(p. 707\)](#)
- [LOWER \(p. 708\)](#)
- [SUBSTRING \(p. 708\)](#)
- [TRIM \(p. 709\)](#)
- [UPPER \(p. 709\)](#)

## CHAR\_LENGTH, CHARACTER\_LENGTH

Conta o número de caracteres na string especificada.

### Note

CHAR\_LENGTH e CHARACTER\_LENGTH são sinônimos.

### Sintaxe

```
CHAR_LENGTH ( string )
```

### Parâmetros

string

A string de destino na qual a função opera.

## Exemplos

```
CHAR_LENGTH('')      -- 0
CHAR_LENGTH('abcdefg') -- 7
```

## LOWER

Dada uma string, converte todos os caracteres maiúsculos em minúsculos. Todos os caracteres minúsculos permanecem inalterados.

### Sintaxe

```
LOWER( string )
```

### Parâmetros

string

A string de destino na qual a função opera.

## Exemplos

```
LOWER('AbCdEfG!@#$') -- 'abcdefg!@#$'
```

## SUBSTRING

Dada uma string, um índice inicial e, opcionalmente, um tamanho, retorna a substring do índice inicial até o final da string ou até o tamanho fornecido.

Note

O primeiro caractere da string de entrada tem o índice 1. Se `start` for < 1, será definido como 1.

### Sintaxe

```
SUBSTRING( string FROM start [ FOR length ] )
```

### Parâmetros

string

A string de destino na qual a função opera.

start

A posição inicial da string.

length

O tamanho da substring a ser retornada. Se não estiver presente, prossiga para o final da string.

## Exemplos

```
SUBSTRING("123456789", 0)      -- "123456789"
SUBSTRING("123456789", 1)      -- "123456789"
SUBSTRING("123456789", 2)      -- "23456789"
```

```
SUBSTRING("123456789", -4)      -- "123456789"  
SUBSTRING("123456789", 0, 999) -- "123456789"  
SUBSTRING("123456789", 1, 5)    -- "12345"
```

## TRIM

Corta os caracteres iniciais ou finais de uma string. O caractere padrão a ser removido é ''.

### Sintaxe

```
TRIM ( [[LEADING | TRAILING | BOTH remove_chars] FROM] string )
```

### Parâmetros

string

A string de destino na qual a função opera.

LEADING | TRAILING | BOTH

Se é necessário cortar os caracteres iniciais ou finais, ou ambos.

remove\_chars

O conjunto de caracteres a ser removido. Observe que `remove_chars` pode ser uma string com tamanho > 1. Essa função retorna a string com qualquer caractere de `remove_chars` encontrado no início ou final da string que foi removida.

## Exemplos

```
TRIM('      foobar      ')          -- 'foobar'  
TRIM(' \tfoobar\t      ')          -- '\tfoobar\t'  
TRIM(LEADING FROM '      foobar      ') -- 'foobar      '  
TRIM(TRAILING FROM '      foobar      ') -- '      foobar'  
TRIM(BOTH FROM '      foobar      ') -- 'foobar'  
TRIM(BOTH '12' FROM '1112211foobar22211122') -- 'foobar'
```

## UPPER

Dada uma string, converte todos os caracteres minúsculos em maiúsculos. Todos os caracteres maiúsculos permanecem inalterados.

### Sintaxe

```
UPPER ( string )
```

### Parâmetros

string

A string de destino na qual a função opera.

## Exemplos

```
UPPER('AbCdEfG!@#$') -- 'ABCDEFG!@#$'
```

# Histórico do documento

- Última atualização da documentação: 4 de dezembro de 2018
- Versão atual da API: 2006-03-01

A tabela a seguir descreve as alterações importantes em cada versão do Guia do desenvolvedor do Amazon Simple Storage Service, de 19 de junho de 2018 em diante. Para receber notificações sobre atualizações dessa documentação, você pode se inscrever em um feed RSS.

update-history-change	update-history-description	update-history-date
<a href="#">Suporte para arquivos de inventário do Amazon S3 no formato Parquet (p. 710)</a>	O Amazon S3 agora dá suporte ao formato <a href="#">Apache Parquet (Parquet)</a> além dos formatos de arquivo <a href="#">Apache optimized row columnar (ORC)</a> e Comma-Separated Values (CSV – Valores separados por vírgulas) para arquivos de saída de inventário. Para obter mais informações, consulte <a href="#">Inventário do Amazon S3</a> .	December 4, 2018
<a href="#">Restaurar notificações de evento (p. 710)</a>	As notificações de evento do Amazon S3 agora dão suporte a eventos de início e conclusão durante a restauração de objetos na classe de armazenamento GLACIER. Para obter mais informações, consulte <a href="#">Notificações de evento</a> .	November 26, 2018
<a href="#">Restaurar atualização rápida (p. 710)</a>	Usando a atualização rápida de restauração do Amazon S3, altere a velocidade de uma restauração da classe de armazenamento GLACIER para mais rapidez durante o andamento da restauração. Para obter mais informações, consulte <a href="#">Restaurar objetos arquivados</a>	November 26, 2018
<a href="#">PUT diretamente para a classe de armazenamento GLACIER (p. 710)</a>	A operação PUT do Amazon S3 agora dá suporte à especificação do GLACIER como a classe de armazenamento durante a criação de objetos. Anteriormente, você tinha que fazer a transição de objetos para a classe de armazenamento GLACIER de outra classe de armazenamento do Amazon S3. Além disso, ao usar a Cross Region Replication (CRR –	November 26, 2018

Replicação entre regiões), você já especifica GLACIER como a classe de armazenamento para objetos replicados. Para obter mais informações sobre a classe de armazenamento GLACIER, consulte [Classes de armazenamento](#). Para obter mais informações sobre como especificar a classe de armazenamento para objetos replicados, [Visão geral da configuração de replicação](#). Para obter mais informações sobre as alterações de PUT para GLACIER REST API, consulte [Histórico do documento: PUT diretamente para GLACIER](#).

<a href="#">Nova classe de armazenamento (p. 710)</a>	O Amazon S3 agora oferece uma nova classe de armazenamento chamada INTELLIGENT_TIERING projetada para dados duradouros com padrões de acesso alternados ou desconhecidos. Para obter mais informações, consulte <a href="#">Classes de armazenamento</a> .	November 26, 2018
<a href="#">Bloqueio de objetos do S3 (p. 710)</a>	O Amazon S3 agora oferece funcionalidade do Object Lock que fornece proteções de gravação única, leitura múltipla, para objetos do Amazon S3. Para obter mais informações, consulte <a href="#">Bloquear objetos</a> .	November 26, 2018
<a href="#">Amazon S3 Block Public Access (p. 710)</a>	O Amazon S3 agora inclui a possibilidade de bloquear acesso público a buckets e objetos por bucket ou conta. Para obter mais informações, consulte <a href="#">Usar o Amazon S3 Block Public Access</a> .	November 15, 2018

Filtrar melhorias nas regras de replicação entre regiões (CRR) (p. 710)	Na configuração da regra de CRR, você pode especificar o filtro de um objeto para escolher um subgrupo de objetos aos quais a regra deve ser aplicada. Antes, você poderia filtrar somente por um prefixo de chaves de objeto. Nesta versão, você pode filtrar usando um prefixo de chaves de objeto, uma ou mais tags de objeto ou ambos. Para obter mais informações, consulte <a href="#">Configuração da CRR: Visão geral da configuração da replicação</a> .	September 19, 2018
Novos recursos do Amazon S3 Select (p. 710)	O Amazon S3 Select agora é compatível com entradas do Apache Parquet, consultas em objetos JSON aninhados e duas novas métricas de monitoramento do Amazon CloudWatch ( <code>SelectScannedBytes</code> e <code>SelectReturnedBytes</code> ).	September 5, 2018
Atualizações agora disponíveis em RSS (p. 710)	Agora você pode assinar um RSS Feed para receber notificações sobre atualizações para Guia do desenvolvedor do Amazon Simple Storage Service.	June 19, 2018

## Atualizações anteriores

A tabela a seguir descreve as alterações importantes em cada versão do Guia do desenvolvedor do Amazon Simple Storage Service antes de 19 junho de 2018.

Alteração	Descrição	Data
Atualização de exemplos de código	<p>Exemplos de código atualizados:</p> <ul style="list-style-type: none"><li>C# – atualizamos todos os exemplos para usar o padrão assíncrono baseado em tarefas. Para obter mais informações, consulte <a href="#">APIs assíncronas de Amazon Web Services para .NET</a> no Guia do desenvolvedor do AWS SDK para .NET. Agora os exemplos de código são compatíveis com a versão 3 do AWS SDK para .NET.</li><li>Java – atualizamos todos os exemplos a fim de usar o modelo de criador do cliente. Para obter mais informações sobre o modelo de criador do cliente, consulte <a href="#">Criar clientes de serviço</a>.</li><li>PHP — Atualizados todos os exemplos para usar o AWS SDK para PHP 3.0. Para obter mais informações sobre o AWS SDK para PHP 3.0, consulte o <a href="#">AWS SDK para PHP</a>.</li></ul>	30 de abril de 2018

Alteração	Descrição	Data
	<ul style="list-style-type: none"> <li>Ruby — atualizamos o código de exemplo para que os exemplos funcionem com a versão 3 do AWS SDK para Ruby.</li> </ul>	
Agora o Amazon S3 relata classes de armazenamento do GLACIER e do ONEZONE_IA para as métricas de armazenamento do Amazon CloudWatch Logs	<p>Além de relatar bytes reais, essas métricas de armazenamento incluem sobrecarga de bytes por objeto para classes de armazenamento aplicáveis (ONEZONE_IA, STANDARD_IA, e GLACIER):</p> <ul style="list-style-type: none"> <li>Para objetos da classe de armazenamento ONEZONE_IA e STANDARD_IA, o Amazon S3 relata objetos com menos de 128 KB como tendo 128 KB. Para obter mais informações, consulte <a href="#">Classes de armazenamento (p. 107)</a>.</li> <li>Para objetos da classe de armazenamento GLACIER, as métricas de armazenamento relatam as seguintes sobrecargas: <ul style="list-style-type: none"> <li>Uma sobrecarga de 32 KB por objeto, cobrada de acordo com a definição de preço da classe de armazenamento GLACIER</li> <li>Uma sobrecarga de 8 KB por objeto, cobrada de acordo com a definição de preço da classe de armazenamento STANDARD</li> </ul> </li> </ul> <p>Para obter mais informações, consulte <a href="#">Fazer a transição de objetos (p. 124)</a>.</p> <p>Para obter mais informações sobre métricas de armazenamento, consulte <a href="#">Métricas de monitoramento com o Amazon CloudWatch (p. 598)</a>.</p>	30 de abril de 2018
Nova classe de armazenamento	O Amazon S3 agora oferece uma nova classe de armazenamento, ONEZONE_IA (IA significa, em inglês, acesso com pouca frequência), para armazenar objetos. Para obter mais informações, consulte <a href="#">Classes de armazenamento (p. 107)</a> .	4 de abril de 2018
Amazon S3 Select	O Amazon S3 agora oferece suporte à recuperação de conteúdo de objetos com base em uma expressão SQL. Para obter mais informações, consulte <a href="#">Selecionar conteúdo de objetos (p. 255)</a> .	4 de abril de 2018
Região Ásia-Pacífico (Osaka-Local)	<p>Agora, o Amazon S3 está disponível na região Ásia-Pacífico (Osaka – Local). Para obter mais informações sobre regiões e endpoints do Amazon S3, consulte <a href="#">Regiões e endpoints</a> no AWS General Reference.</p> <p><b>Important</b></p> <p>Você pode usar a região Ásia-Pacífico (Osaka – Local) apenas em conjunto com a Região Ásia-Pacífico (Tóquio). Para solicitar acesso à região Ásia-Pacífico (Osaka – Local), entre em contato com o representante de vendas.</p>	12 de fevereiro de 2018

Alteração	Descrição	Data
Time stamp de criação de inventário do Amazon S3	O inventário do Amazon S3 agora inclui um time stamp da data e da hora de início da criação do relatório de inventário do Amazon S3. Você pode usar o time stamp para determinar alterações no armazenamento do Amazon S3 a partir da hora de início em que o relatório de inventário foi gerado.	16 de janeiro de 2018
Região da UE (Paris)	Agora, o Amazon S3 está disponível na região UE (Paris). Para obter mais informações sobre regiões e endpoints do Amazon S3, consulte <a href="#">Regiões e endpoints</a> no AWS General Reference.	18 de dezembro de 2017
Região da China (Ningxia)	Agora, o Amazon S3 está disponível na região China (Ningxia). Para obter mais informações sobre regiões e endpoints do Amazon S3, consulte <a href="#">Regiões e endpoints</a> no AWS General Reference.	11 de dezembro de 2017
Consulta de arquivos com SQL	Agora o Amazon S3 oferece suporte à consulta de arquivos de dados de Glacier com SQL. Para obter mais informações, consulte <a href="#">Consultar objetos arquivados (p. 263)</a> .	29 de novembro de 2017
Suporte para arquivos de inventário do Amazon S3 no formato ORC	Agora, o Amazon S3 é compatível com arquivos de saída de inventário no formato <a href="#">colunar de linhas otimizado do Apache (ORC)</a> e no formato de valores separados por vírgulas (CSV). Além disso, você já pode consultar o inventário do Amazon S3 usando o SQL padrão com o Amazon Athena, o Amazon Redshift Spectrum e outras ferramentas, como <a href="#">Presto</a> , <a href="#">Apache Hive</a> e <a href="#">Apache Spark</a> . Para obter mais informações, consulte <a href="#">Inventário do Amazon S3 (p. 273)</a> .	17 de novembro de 2017
Criptografia padrão para buckets do S3	A criptografia padrão do Amazon S3 fornece uma forma de configurar o comportamento de criptografia padrão para um bucket do S3. Você pode configurar a criptografia padrão em um bucket para que todos os objetos sejam criptografados quando forem armazenados nele. Os objetos são criptografados usando a criptografia do lado do servidor com as chaves gerenciadas pelo Amazon S3 (SSE-S3) ou as chaves gerenciadas pelo AWS KMS (SSE-KMS). Para obter mais informações, consulte <a href="#">Criptografia padrão do Amazon S3 para buckets do S3 (p. 68)</a> .	06 de novembro de 2017
Status da criptografia no inventário do Amazon S3	Agora, o Amazon S3 é compatível e inclui o status da criptografia no inventário do Amazon S3 para que você possa ver como seus objetos são criptografados em repouso para a auditoria de conformidade e outras finalidades. Você também pode configurar para criptografar o inventário do S3 com a criptografia do lado do servidor (SSE) ou o SSE-KMS para que todos os arquivos do inventário sejam criptografados apropriadamente. Para obter mais informações, consulte <a href="#">Inventário do Amazon S3 (p. 273)</a> .	06 de novembro de 2017

Alteração	Descrição	Data
Melhorias da replicação entre regiões (CRR)	<p>A replicação entre regiões agora oferece suporte para o seguinte:</p> <ul style="list-style-type: none"> <li>Em um cenário entre contas, você pode adicionar uma configuração da CRR a fim de alterar a propriedade da réplica para a conta da AWS que tem o bucket de destino. Para obter mais informações, consulte <a href="#">Configuração adicional da CRR: alteração do proprietário da réplica (p. 560)</a>.</li> <li>Por padrão, o Amazon S3 não replica os objetos em seu bucket de origem que são criados com a criptografia do lado do servidor que usa as chaves gerenciadas pelo AWS KMS. Agora, na sua configuração de CRR, você pode orientar o Amazon S3 a fim de replicar esses objetos. Para obter mais informações, consulte <a href="#">Outra configuração de CRR: replicar objetos criados com a criptografia do lado do servidor (SSE) usando chaves de criptografia gerenciadas pelo AWS KMS (p. 563)</a>.</li> </ul>	06 de novembro de 2017
Região da UE (Londres)	<p>Agora, o Amazon S3 está disponível na região UE (Londres). Para obter mais informações sobre regiões e endpoints do Amazon S3, consulte <a href="#">Regiões e endpoints</a> no AWS General Reference.</p>	13 de dezembro de 2016
Região do Canadá (Central)	<p>Agora, o Amazon S3 está disponível na região Canadá (Central). Para obter mais informações sobre regiões e endpoints do Amazon S3, consulte <a href="#">Regiões e endpoints</a> no AWS General Reference.</p>	8 de dezembro de 2016

Alteração	Descrição	Data
Marcação de objetos	<p>O Amazon S3 agora oferece suporte para marcação de objetos. Ela permite classificar o armazenamento. Os prefixos de nome de chave de objeto também permitem classificar o armazenamento, mas a marcação de objetos adiciona outra dimensão a isso.</p> <p>Há benefícios adicionados oferecidos pela marcação. Dentre elas estão:</p> <ul style="list-style-type: none"> <li>As tags de objeto permitem um controle de acesso rigoroso das permissões (por exemplo, você pode conceder permissões de usuário do IAM a objetos somente leitura com tags específicas).</li> <li>Controle fino para especificar a configuração de ciclo de vida. Você pode especificar tags para selecionar um subconjunto de objetos aos quais a regra de ciclo de vida se aplica.</li> <li>Se você tiver a replicação entre regiões (CRR) configurada, o Amazon S3 poderá replicar as tags. Você deve conceder a permissão necessária para a função do IAM criada para o Amazon S3 assumir que é preciso replicar objetos em seu nome.</li> <li>Você também pode personalizar métricas do CloudWatch e eventos do CloudTrail para exibir informações por filtros de tag específicos.</li> </ul> <p>Para obter mais informações, consulte <a href="#">Marcação de objetos (p. 114)</a>.</p>	29 de novembro de 2016
O ciclo de vida do S3 agora oferece suporte para filtro com base em tag	Agora, o Amazon S3 oferece suporte para filtragem com base em tag na configuração de ciclo de vida. Você já pode especificar a regra do ciclo de vida na qual pode determinar um prefixo de chave, uma ou mais tags de objeto ou uma combinação de ambos para selecionar um subconjunto de objetos ao qual a regra do ciclo de vida se aplica. Para obter mais informações, consulte <a href="#">Gerenciamento do ciclo de vida de objetos (p. 122)</a> .	29 de novembro de 2016
Métricas de solicitação do CloudWatch para buckets	O Amazon S3 agora oferece suporte para métricas do CloudWatch para solicitações feitas em buckets. Quando você habilita essas métricas para um bucket, elas são registradas em intervalos de 1 minuto. Você também pode configurar quais objetos em um bucket registrarão essas métricas de solicitação. Para obter mais informações, consulte <a href="#">Métricas de monitoramento com o Amazon CloudWatch (p. 598)</a> .	29 de novembro de 2016
Inventário do Amazon S3	<p>O Amazon S3 agora é compatível com inventário de armazenamento. O inventário do Amazon S3 fornece uma saída de arquivo sem formatação de seus objetos e dos metadados correspondentes diária ou semanalmente para um bucket do S3 ou um prefixo compartilhado (ou seja, objetos que têm nomes que começam com uma string em comum).</p> <p>Para obter mais informações, consulte <a href="#">Inventário do Amazon S3 (p. 273)</a>.</p>	29 de novembro de 2016

Alteração	Descrição	Data
Análise do Amazon S3 – Análise de classe de armazenamento	<p>O novo recurso de análise do Amazon S3 – análise de classe de armazenamento observa padrões de acesso de dados para ajudar você a determinar quando fazer a transição do armazenamento STANDARD acessado menos frequentemente para a classe de armazenamento STANDARD_IA (IA, para acesso raro). Depois que a análise de classe de armazenamento observa os padrões incomuns de acesso a um conjunto filtrado de dados em um período, você pode usar os resultados da análise para ajudá-lo a melhorar suas políticas de ciclo de vida. Esse recurso também inclui uma análise diária detalhada do uso de armazenamento no nível de bucket, prefixo ou tag especificado que você pode exportar para um bucket do S3.</p> <p>Para obter mais informações, consulte <a href="#">Análise do Amazon S3 – análise de classe de armazenamento (p. 267)</a> no Guia do desenvolvedor do Amazon Simple Storage Service.</p>	29 de novembro de 2016
Novas recuperações de dados expressas e em massa ao restaurar objetos arquivados no Glacier	O Amazon S3 agora oferece suporte para recuperações de dados expressas e em massa, além de recuperações padrão ao restaurar objetos arquivados no Glacier. Para obter mais informações, consulte <a href="#">Restaurar objetos arquivados (p. 259)</a> .	21 de novembro de 2016
Registro de objetos do CloudTrail	O CloudTrail oferece suporte ao registro de operações de API no nível do objeto do Amazon S3, como <code>GetObject</code> , <code>PutObject</code> e <code>DeleteObject</code> . Você pode configurar seletores de eventos para registrar operações de API no nível do objeto. Para obter mais informações, consulte <a href="#">Registro em log de chamadas à API do Amazon S3 usando o AWS CloudTrail (p. 606)</a> .	21 de novembro de 2016
Região do Leste dos EUA (Ohio)	Agora, o Amazon S3 está disponível na região Leste dos EUA (Ohio). Para obter mais informações sobre regiões e endpoints do Amazon S3, consulte <a href="#">Regiões e endpoints</a> no AWS General Reference.	17 de outubro de 2016
Suporte a IPv6 para Amazon S3 Transfer Acceleration	O Amazon S3 agora oferece suporte a Protocolo de Internet versão 6 (IPv6) para Amazon S3 Transfer Acceleration. Você pode conectar-se ao Amazon S3 via IPv6 usando a nova pilha dupla para o endpoint Transfer Acceleration. Para obter mais informações, consulte <a href="#">Conceitos básicos do Amazon S3 Transfer Acceleration (p. 76)</a> .	6 de outubro de 2016
Suporte a IPv6	O Amazon S3 agora oferece suporte para Protocolo de Internet versão 6 (IPv6). Você pode acessar o Amazon S3 via IPv6 usando endpoints de pilha dupla. Para obter mais informações, consulte <a href="#">Fazer solicitações ao Amazon S3 por meio do IPv6 (p. 12)</a> .	11 de agosto de 2016
Região Ásia Pacífico (Mumbai)	Agora, o Amazon S3 está disponível na região Ásia Pacífico (Mumbai). Para obter mais informações sobre regiões e endpoints do Amazon S3, consulte <a href="#">Regiões e endpoints</a> no AWS General Reference.	27 de junho de 2016

Alteração	Descrição	Data
Amazon S3 Transfer Acceleration	<p>O Amazon S3 Transfer Acceleration possibilita transferências de arquivos rápidas, fáceis e seguras em longas distâncias entre o seu cliente e um bucket do S3. O Transfer Acceleration aproveita os pontos de presença distribuídos globalmente do Amazon CloudFront.</p> <p>Para obter mais informações, consulte <a href="#">Amazon S3 Transfer Acceleration (p. 75)</a>.</p>	19 de abril de 2016
Suporte de ciclo de vida para remover os marcadores de exclusão de objeto expirado	<p>A ação <code>Expiration</code> de configuração de ciclo de vida agora permite que você instrua o Amazon S3 a remover os marcadores de exclusão de objeto expirados em um bucket com versões. Para obter mais informações, consulte <a href="#">Elementos para descrever ações de ciclo de vida (p. 133)</a>.</p>	16 de março de 2016
A configuração de ciclo de vida de bucket agora oferece suporte para anular multipart uploads incompletos	<p>A configuração de ciclo de vida de bucket agora oferece suporte para a ação <code>AbortIncompleteMultipartUpload</code>, que você pode usar para instruir o Amazon S3 a anular os multipart uploads não concluídos em um número específico de dias após serem iniciados. Quando um multipart upload torna-se qualificado para uma operação de anulação, o Amazon S3 exclui todas as partes carregadas e anula o multipart upload.</p> <p>Para obter informações conceituais, consulte os seguintes tópicos no Guia do desenvolvedor do Amazon Simple Storage Service:</p> <ul style="list-style-type: none"> <li>• <a href="#">Anular multipart uploads incompletos usando uma política de ciclo de vida de bucket (p. 183)</a></li> <li>• <a href="#">Elementos para descrever ações de ciclo de vida (p. 133)</a></li> </ul> <p>As seguintes operações de API foram atualizadas para oferecer suporte à nova ação:</p> <ul style="list-style-type: none"> <li>• <a href="#">Ciclo de vida de bucket PUT</a> – A configuração de XML agora permite que você especifique a ação <code>AbortIncompleteMultipartUpload</code> em uma regra de configuração de ciclo de vida.</li> <li>• <a href="#">Listar partes e Iniciar o multipart upload</a> – Essas duas operações de API agora retornarão dois cabeçalhos de resposta adicionais (<code>x-amz-abort-date</code> e <code>x-amz-abort-rule-id</code>) se o bucket tiver uma regra de ciclo de vida que especifique a ação <code>AbortIncompleteMultipartUpload</code>. Esses cabeçalhos de resposta indicam quando o multipart upload iniciado se tornará qualificado para a operação de anulação e qual regra de ciclo de vida é aplicável.</li> </ul>	16 de março de 2016
Região Ásia-Pacífico (Seul)	<p>Agora, o Amazon S3 está disponível na região Ásia-Pacífico (Seul). Para obter mais informações sobre regiões e endpoints do Amazon S3, consulte <a href="#">Regiões e endpoints</a> no AWS General Reference.</p>	6 de janeiro de 2016

Alteração	Descrição	Data
Nova chave de condição e uma alteração do multipart upload	<p>As políticas do IAM agora oferecem suporte para a chave de condição <code>s3:x-amz-storage-class</code> do Amazon S3. Para obter mais informações, consulte <a href="#">Especificação de condições em uma política (p. 335)</a>.</p> <p>Você não precisa mais ser o iniciador de um multipart upload para fazer upload de partes e terminar o upload. Para obter mais informações, consulte <a href="#">API de multipart upload e permissões (p. 186)</a>.</p>	14 de dezembro de 2015
Região Padrão dos EUA renomeada	A string de nome de região mudou de "Padrão dos EUA" para "Leste dos EUA (Norte da Virgínia)". É só uma atualização de nome de região, não há nenhuma alteração na funcionalidade.	11 de dezembro de 2015
Nova classe de armazenamento	<p>O Amazon S3 agora oferece uma nova classe de armazenamento, STANDARD_IA (IA, para acesso raro), para armazenar objetos. Essa classe de armazenamento é otimizada para dados armazenados por longo tempo e acessados com menos frequência. Para obter mais informações, consulte <a href="#">Classes de armazenamento (p. 107)</a>.</p> <p>As atualizações de recurso de configuração de ciclo de vida agora permitem fazer a transição de objetos para a classe de armazenamento STANDARD_IA. Para obter mais informações, consulte <a href="#">Gerenciamento do ciclo de vida de objetos (p. 122)</a>.</p> <p>Anteriormente, o recurso de replicação entre regiões usava a classe de armazenamento do objeto de origem para réplicas de objeto. Agora, ao configurar a replicação entre regiões, você pode especificar uma classe de armazenamento para a réplica de objeto criada no bucket de destino. Para obter mais informações, consulte <a href="#">Replicação entre regiões (p. 544)</a>.</p>	16 de setembro de 2015
Integração do AWS CloudTrail	A nova integração do AWS CloudTrail permite que você registre a atividade da API do Amazon S3 no bucket do S3. Você pode usar o CloudTrail para acompanhar criações ou exclusões de bucket do S3, modificações de controle de acesso ou alterações de política de ciclo de vida. Para obter mais informações, consulte <a href="#">Registro em log de chamadas à API do Amazon S3 usando o AWS CloudTrail (p. 606)</a> .	1 de setembro de 2015
Aumento do limite do bucket	O Amazon S3 agora oferece suporte para aumentos de limite de bucket. Por padrão, os clientes podem criar até 100 buckets na conta da AWS. Os clientes que precisarem de buckets adicionais poderão aumentar esse limite enviando um aumento de limite de serviço. Para obter informações sobre como aumentar o limite de bucket, vá para <a href="#">Limites de serviço da AWS</a> na Referência geral da AWS. Para obter mais informações, consulte <a href="#">Criação de um bucket (p. 54)</a> e <a href="#">Restrições e limitações do bucket (p. 59)</a> .	4 de agosto de 2015

Alteração	Descrição	Data
Atualização do modelo de consistência	O Amazon S3 agora oferece suporte para consistência de leitura após gravação para novos objetos adicionados ao Amazon S3 na região Leste dos EUA (Norte da Virgínia). Antes dessa atualização, todas as regiões menos a região Leste dos EUA (Norte da Virgínia) permitiam a consistência de leitura após gravação para novos objetos carregados no Amazon S3. Com esse aprimoramento, o Amazon S3 agora oferece suporte para consistência de leitura após gravação em todas as regiões para novos objetos adicionados ao Amazon S3. A consistência de leitura após gravação permite que você recupere objetos imediatamente após a criação no Amazon S3. Para obter mais informações, consulte <a href="#">Regiões (p. 4)</a> .	4 de agosto de 2015
Notificações de eventos	As notificações de evento do Amazon S3 foram atualizadas para adicionar notificações quando objetos são excluídos e para adicionar a filtragem em nomes de objeto com correspondência de prefixo e sufixo. Para obter mais informações, consulte <a href="#">Configurar notificações de evento do Amazon S3 (p. 522)</a> .	28 de julho de 2015
Integração do Amazon CloudWatch	A nova integração do Amazon CloudWatch permite que você monitore e defina alarmes de uso do Amazon S3 com métricas do CloudWatch para o Amazon S3. As métricas compatíveis incluem total de bytes para armazenamento padrão, total de bytes para armazenamento com menos redundância e o número total de objetos para um determinado bucket do S3. Para obter mais informações, consulte <a href="#">Métricas de monitoramento com o Amazon CloudWatch (p. 598)</a> .	28 de julho de 2015
Suporte para excluir e esvaziar buckets não vazios	O Amazon S3 agora oferece suporte para excluir e esvaziar buckets não vazios. Para obter mais informações, consulte <a href="#">Exclusão ou esvaziamento do bucket (p. 63)</a> .	16 de julho de 2015
Políticas de bucket para VPC endpoints Amazon	O Amazon S3 adicionou suporte para políticas de bucket para endpoints do Amazon Virtual Private Cloud (Amazon VPC). Você pode usar políticas de bucket do S3 para controlar o acesso a buckets de endpoints específicos do Amazon VPC ou VPCs específicas. Os VPC endpoints são fáceis de configurar, são extremamente confiáveis e fornecem uma conexão segura com o Amazon S3 sem exigir um gateway ou uma instância NAT. Para obter mais informações, consulte <a href="#">Exemplo de políticas de bucket para VPC endpoints para o Amazon S3 (p. 365)</a> .	29 de abril de 2015
Notificações de eventos	As notificações de evento do Amazon S3 foram atualizadas para oferecer suporte à alternância para permissões com base em recursos para funções do AWS Lambda. Para obter mais informações, consulte <a href="#">Configurar notificações de evento do Amazon S3 (p. 522)</a> .	9 de abril de 2015
Replicação entre regiões	O Amazon S3 agora oferece suporte para a replicação entre regiões. A replicação entre regiões é a cópia assíncrona automática de objetos em buckets, em diferentes regiões da AWS. Para obter mais informações, consulte <a href="#">Replicação entre regiões (p. 544)</a> .	24 de março de 2015

Alteração	Descrição	Data
Notificações de eventos	O Amazon S3 agora oferece suporte para novos tipos e destinos de evento em uma configuração de notificação de bucket. Antes dessa versão, o Amazon S3 era compatível somente com o tipo de evento s3:ReducedRedundancyLostObject e um tópico do Amazon SNS como o destino. Para obter mais informações sobre os novos tipos de evento, consulte <a href="#">Configurar notificações de evento do Amazon S3 (p. 522)</a> .	13 de novembro de 2014
Criptografia do lado do servidor com chaves fornecidas pelo cliente	<p>Criptografia no lado do servidor com AWS Key Management Service (AWS KMS)</p> <p>O Amazon S3 agora oferece suporte para criptografia do lado do servidor usando o AWS Key Management Service. Esse recurso permite que você gerencie a chave de envelope por meio do AWS KMS, e o Amazon S3 chama o AWS KMS para acessar a chave de envelope nas permissões definidas.</p> <p>Para obter mais informações sobre a criptografia do lado do servidor com o AWS KMS, consulte <a href="#">Proteger dados usando criptografia do lado do servidor com o AWS Key Management Service</a>.</p>	12 de novembro de 2014
Região UE (Frankfurt)	Agora, o Amazon S3 está disponível na região UE (Frankfurt).	23 de outubro de 2014
Criptografia do lado do servidor com chaves fornecidas pelo cliente	<p>O Amazon S3 agora oferece suporte para criptografia de servidor com chaves de criptografia fornecidas pelo cliente (SSE-C). A criptografia do lado do servidor permite que você solicite ao Amazon S3 para criptografar seus dados em repouso. Ao usar SSE-C, o Amazon S3 criptografa os objetos com chaves de criptografia personalizadas que você fornece. Como o Amazon S3 executa a criptografia, você obtém os benefícios de usar suas próprias chaves de criptografia sem o custo de gravação ou execução de seu próprio código de criptografia.</p> <p>Para obter mais informações sobre SSE-C, consulte <a href="#">Criptografia do servidor (com chaves de criptografia fornecidas pelo cliente)</a>.</p>	12 de junho de 2014
Suporte de ciclo de vida para versionamento	Antes dessa versão, a configuração de ciclo de vida era permitida somente em buckets sem versões. Agora você pode configurar o ciclo de vida em buckets sem versões e com o versionamento ativado. Para obter mais informações, consulte <a href="#">Gerenciamento do ciclo de vida de objetos (p. 122)</a> .	20 de maio de 2014
Tópicos de controle de acesso revisados	Documentação revisada de controle de acesso do Amazon S3. Para obter mais informações, consulte <a href="#">Gerenciamento de permissões de acesso aos recursos do Amazon S3 (p. 282)</a> .	15 de abril de 2014
Tópico de registro de acesso de servidor revisado	Documentação revisada de registro de acesso de servidor. Para obter mais informações, consulte <a href="#">Registro em log de acesso ao servidor Amazon S3 (p. 625)</a> .	26 de novembro de 2013

Alteração	Descrição	Data
Exemplos de SDK do .NET atualizados para a versão 2.0	Os exemplos de SDK do .NET neste guia agora são compatíveis com a versão 2.0.	26 de novembro de 2013
Suporte de SOAP via HTTP obsoleto	O suporte de SOAP via HTTP está obsoleto, mas continua disponível via HTTPS. Novos recursos do Amazon S3 não serão compatíveis com o SOAP. Recomendamos que você use a API REST ou os SDKs da AWS.	20 de setembro de 2013
Suporte para variáveis de política do IAM	<p>Agora, a linguagem de política de acesso do IAM oferece suporte para variáveis. Quando uma política é avaliada, todas as variáveis de política são substituídas por valores fornecidos por informações baseadas no contexto da sessão do usuário autenticado. Você pode usar variáveis de política para definir políticas de uso geral sem, explicitamente, listar todos os componentes da política. Para obter mais informações sobre variáveis de política, leia <a href="#">Visão geral de variáveis de política do IAM</a> no Guia do usuário do IAM.</p> <p>Para obter exemplos de variáveis de políticas no Amazon S3, consulte <a href="#">Exemplos de política de usuário (p. 367)</a>.</p>	3 de abril de 2013
Suporte do console para Pagamento pelo solicitante	Agora você pode configurar seu bucket para Pagamento pelo solicitante usando o console do Amazon S3. Para obter mais informações, consulte <a href="#">Configurar Pagamento pelo solicitante usando o console do Amazon S3 (p. 83)</a> .	31 de dezembro de 2012
Suporte de domínio raiz para hospedagem de sites	O Amazon S3 agora oferece suporte a hospedagem de sites estáticos no domínio raiz. Os visitantes podem acessar seu site do navegador sem especificar "www" no endereço da web (por exemplo, "example.com"). Muitos clientes já hospedam sites estáticos no Amazon S3 que podem ser acessados em um subdomínio "www" (por exemplo, "www.example.com"). Anteriormente, para oferecer suporte para acesso de domínio raiz, você precisava executar seu próprio servidor da web para as solicitações de domínio raiz de proxy de navegadores para seu site no Amazon S3. Executar um servidor da web para solicitações de proxy introduz custos adicionais, carga operacional e outro ponto de falha em potencial. Agora, você pode aproveitar a alta disponibilidade e a durabilidade do Amazon S3 para endereços "www" e de domínio raiz. Para obter mais informações, consulte <a href="#">Hospedagem de um site estático no Amazon S3 (p. 494)</a> .	27 de dezembro de 2012
Revisão do console	O console do Amazon S3 foi atualizado. Os tópicos de documentação que se referem ao console foram revisados conforme necessário.	14 de dezembro de 2012

Alteração	Descrição	Data
Suporte para arquivamento de dados no Glacier	<p>Agora, o Amazon S3 oferece suporte a uma opção de armazenamento que permite utilizar o serviço de armazenamento de baixo custo do Glacier para arquivamento de dados. Para arquivar objetos, você define regras de arquivamento e identifica objetos e um cronograma quando deseja que o Amazon S3 arquive esses objetos no Glacier. Você pode definir regras facilmente em um bucket usando o console do Amazon S3 ou usando programaticamente a API do Amazon S3 ou SDKs da AWS.</p> <p>Para obter mais informações, consulte <a href="#">Gerenciamento do ciclo de vida de objetos (p. 122)</a>.</p>	13 de novembro de 2012
Suporte para redirecionamentos de página de site	<p>Para um bucket que é configurado como um site, o Amazon S3 agora oferece suporte ao redirecionamento de uma solicitação de um objeto para outro objeto no mesmo bucket ou um URL externo. Para obter mais informações, consulte <a href="#">(Opcional) Configuração de um redirecionamento de uma página da web (p. 502)</a>.</p> <p>Para obter informações sobre a hospedagem de sites, consulte <a href="#">Hospedagem de um site estático no Amazon S3 (p. 494)</a>.</p>	4 de outubro de 2012
Suporte para compartilhamento de recursos de origem cruzada (CORS)	<p>O Amazon S3 oferece suporte para compartilhamento de recursos de origem cruzada (CORS). O CORS define uma maneira de os aplicativos web clientes carregados em um domínio poderem interagir com ou acessar recursos em outro domínio. Com suporte a CORS no Amazon S3, você pode criar aplicativos web cliente avançados no Amazon S3 e permitir seletivamente o acesso de domínio cruzado aos recursos do Amazon S3. Para obter mais informações, consulte <a href="#">Cross-Origin Resource Sharing (CORS, Compartilhamento de recursos de origem cruzada) (p. 156)</a>.</p>	31 de agosto de 2012
Suporte para tags de alocação de custos	<p>O Amazon S3 agora oferece suporte para tags de alocação de custos, o que permite identificar buckets do S3 para facilitar o acompanhamento de custos de projetos ou outros critérios. Para obter mais informações sobre o uso de tags para buckets, consulte <a href="#">Usar tags de alocação de custos para buckets do S3 (p. 99)</a>.</p>	21 de agosto de 2012

Alteração	Descrição	Data
Suporte para acesso à API protegido por MFA nas políticas de bucket	<p>O Amazon S3 agora oferece suporte para acesso ao MFA protegido por API, um recurso que pode impor a AWS Multi-Factor Authentication para um nível de segurança extra ao acessar os recursos do Amazon S3. É um recurso de segurança que exige que os usuários comprovem a posse física de um dispositivo MFA fornecendo um código válido de MFA. Para obter mais informações, consulte <a href="#">AWS Multi-Factor Authentication</a>. Agora você pode exigir a autenticação de MFA para todas as solicitações de acesso aos recursos do Amazon S3.</p> <p>Para aplicar a autenticação de MFA, o Amazon S3 agora oferece suporte para a chave <code>aws:MultiFactorAuthAge</code> em uma política de bucket. Para ver um exemplo de política de bucket, consulte <a href="#">Adição de uma política de bucket para exigir MFA (p. 362)</a>.</p>	10 de julho de 2012
Suporte para expiração de objeto	Você pode usar a expiração de objeto para programar a remoção automática de dados após um período configurado. Para definir a expiração de objeto, adicione a configuração de ciclo de vida a um bucket.	27 de dezembro de 2011
Novas regiões suportadas	O Amazon S3 agora oferece suporte para a região América do Sul (São Paulo). Para obter mais informações, consulte <a href="#">Acesso a um bucket (p. 56)</a> .	14 de dezembro de 2011
Exclusão de vários objetos	O Amazon S3 agora oferece suporte para a API de exclusão de vários objetos que permite excluir vários objetos em uma única solicitação. Com esse recurso, você pode remover um grande número de objetos do Amazon S3 mais rapidamente do que usando várias solicitações DELETE individuais. Para obter mais informações, consulte <a href="#">Excluir objetos (p. 237)</a> .	7 de dezembro de 2011
Novas regiões suportadas	O Amazon S3 agora oferece suporte para a região Oeste dos EUA (Oregon). Para obter mais informações, consulte <a href="#">Buckets e regiões (p. 56)</a> .	8 de novembro de 2011
Atualização da documentação	Correções de erros de documentação.	8 de novembro de 2011
Atualização da documentação	<p>Além de correções de erros de documentação, esta versão inclui os seguintes aprimoramentos:</p> <ul style="list-style-type: none"> <li>Novas seções de criptografia do lado do servidor usando o AWS SDK para PHP (consulte <a href="#">Especificação da criptografia de servidor usando o AWS SDK para PHP (p. 421)</a>) e o AWS SDK para Ruby (consulte <a href="#">Especificação da criptografia no lado do servidor usando o AWS SDK para Ruby (p. 423)</a>).</li> <li>Nova seção sobre como criar e testar amostras de Ruby (consulte <a href="#">Usar o AWS SDK para Ruby - versão 3 (p. 650)</a>).</li> </ul>	17 de outubro de 2011

Alteração	Descrição	Data
Suporte para criptografia do lado do servidor	<p>O Amazon S3 agora oferece suporte para criptografia do lado do servidor. Ela permite solicitar ao Amazon S3 para criptografar seus dados em repouso, ou seja, criptografar seus dados de objeto quando o Amazon S3 grava seus dados em discos nos datacenters. Além das atualizações de API REST, o AWS SDK for Java e o .NET oferecem a funcionalidade necessária para solicitar criptografia do lado do servidor. Você também pode solicitar a criptografia de servidor ao fazer upload de objetos usando o Console de Gerenciamento da AWS.</p> <p>Para saber mais sobre a criptografia de dados, acesse <a href="#">Usar criptografia de dados</a>.</p>	4 de outubro de 2011
Atualização da documentação	<p>Além de correções de erros de documentação, esta versão inclui os seguintes aprimoramentos:</p> <ul style="list-style-type: none"> <li>Adição de exemplos de Ruby e PHP à seção <a href="#">Fazer solicitações (p. 10)</a>.</li> <li>Adição de seções que descrevem como gerar e usar pre-signed URLs. Para obter mais informações, consulte <a href="#">Compartilhe um objeto (p. 172)</a> e <a href="#">Fazer upload de objetos usando pre-signed URLs (p. 214)</a>.</li> <li>Atualização de uma seção existente para apresentar o AWS Explorer para Eclipse e Visual Studio. Para obter mais informações, consulte <a href="#">Usar os AWS SDKs, a CLI e os Explorers (p. 639)</a>.</li> </ul>	22 de setembro de 2011
Suporte para enviar solicitações usando credenciais de segurança temporárias	<p>Além de usar suas credenciais de segurança de conta da AWS e de usuário do IAM para enviar solicitações autenticadas ao Amazon S3, agora você pode enviar solicitações usando credenciais de segurança temporárias obtidas no AWS Identity and Access Management (IAM). Você pode usar a API do AWS Security Token Service ou bibliotecas de wrapper do SDK da AWS para solicitar essas credenciais temporárias no IAM. Você pode solicitar essas credenciais de segurança temporárias para seu próprio uso ou para enviá-las para usuários federados e aplicativos. Esse recurso permite que você gerencie usuários fora da AWS e forneça a eles as credenciais de segurança temporárias para acessar os recursos da AWS.</p> <p>Para obter mais informações, consulte <a href="#">Fazer solicitações (p. 10)</a>.</p> <p>Para obter mais informações sobre suporte do IAM para credenciais de segurança temporárias, consulte <a href="#">Credenciais de segurança temporárias</a> no Guia do usuário do IAM.</p>	3 de agosto de 2011

Alteração	Descrição	Data
API de multipart upload estendida para permitir a cópia de objetos de até 5 TB	<p>Antes dessa versão, a API do Amazon S3 permitia a cópia de objetos de até 5 GB. Para permitir a cópia de objetos com mais de 5 GB, o Amazon S3 agora estende a API de multipart upload com uma nova operação, <a href="#">Upload Part (Copy)</a>. Você pode usar essa operação do multipart upload para copiar objetos com até 5 TB. Para obter mais informações, consulte <a href="#">Cópia de objetos (p. 219)</a>.</p> <p>Para obter informações conceituais sobre a API do multipart upload, consulte <a href="#">Upload de objetos usando a API de multipart upload (p. 181)</a>.</p>	21 de junho de 2011
Chamadas de API SOAP via HTTP desativadas	Para aumentar a segurança, as chamadas de API SOAP via HTTP são desativadas. As solicitações SOAP autenticadas e anônimas devem ser enviadas para o Amazon S3 usando SSL.	6 de junho de 2011
O IAM habilita a delegação entre contas	<p>Anteriormente, para acessar um recurso do Amazon S3, um usuário do IAM precisava de permissões da conta da AWS pai e do proprietário do recurso do Amazon S3. Com o acesso entre contas, o usuário do IAM agora só precisa da permissão da conta do proprietário. Isto é, se o proprietário de um recurso conceder acesso a uma conta da AWS, a conta da AWS agora poderá conceder aos usuários do IAM acesso a esses recursos.</p> <p>Para obter mais informações, consulte <a href="#">Criar uma função para delegar permissões a um usuário do IAM</a> no Guia do usuário do IAM.</p> <p>Para obter mais informações sobre como especificar principais em uma política de bucket, consulte <a href="#">Especificação de um principal em uma política (p. 329)</a>.</p>	6 de junho de 2011
Novo link	As informações de endpoint deste serviço agora estão localizadas na Referência geral da AWS. Para obter mais informações, consulte Regiões e endpoints em <a href="#">Referência geral da AWS</a> .	1 de março de 2011
Suporte para a hospedagem de sites estáticos no Amazon S3	O Amazon S3 introduz o suporte aprimorado para hospedagem de sites estáticos. Isso inclui suporte para documentos de índice e documentos de erros personalizados. Ao usar esses recursos, as solicitações para a raiz de seu bucket ou uma subpasta (por exemplo, <code>http://mywebsite.com/subfolder</code> ) retornam o documento de índice em vez da lista de objetos em seu bucket. Se um erro for encontrado, o Amazon S3 retornará sua mensagem de erro personalizada em vez de uma mensagem de erro do Amazon S3. Para obter mais informações, consulte <a href="#">Hospedagem de um site estático no Amazon S3 (p. 494)</a> .	17 de fevereiro de 2011
Suporte para API do cabeçalho de resposta	A API REST do objeto GET agora permite alterar os cabeçalhos de resposta da solicitação REST do objeto GET para cada solicitação. Isto é, você pode alterar metadados de objeto na resposta, sem modificar o objeto em si. Para obter mais informações, consulte <a href="#">Obtenção de objetos (p. 166)</a> .	14 de janeiro de 2011

Alteração	Descrição	Data
Suporte para objetos grandes	O Amazon S3 aumentou o tamanho máximo de um objeto que você pode armazenar em um bucket do S3 de 5 GB para 5 TB. Se estiver usando a API REST, você poderá fazer upload de objetos de até 5 GB em uma única operação PUT. Para objetos maiores, você deve usar a API REST do multipart upload para fazer upload de objetos em partes. Para obter mais informações, consulte <a href="#">Upload de objetos usando a API de multipart upload (p. 181)</a> .	9 de dezembro de 2010
Multipart upload	O multipart upload permite fazer uploads mais flexíveis e rapidamente no Amazon S3. Ele permite carregar um único objeto como um conjunto de partes. Para obter mais informações, consulte <a href="#">Upload de objetos usando a API de multipart upload (p. 181)</a> .	10 de novembro de 2010
Suporte de ID canônico em políticas de bucket	Agora você pode especificar IDs canônicos em políticas de bucket. Para obter mais informações, consulte <a href="#">Visão geral da linguagem da política de acesso (p. 326)</a>	17 de setembro de 2010
O Amazon S3 funciona com o IAM.	Esse serviço agora se integra ao AWS Identity and Access Management (IAM). Para obter mais informações, acesse <a href="#">Serviços da AWS que funcionam com o IAM</a> no Guia do usuário do IAM.	2 de setembro de 2010
Notificações	O recurso de notificações do Amazon S3 permite que você configure um bucket para que o Amazon S3 publique uma mensagem para um tópico do Amazon Simple Notification Service (Amazon SNS) quando o Amazon S3 detectar um evento de chave em um bucket. Para obter mais informações, consulte <a href="#">Configuração de notificação de eventos de bucket (p. 522)</a> .	14 de julho de 2010
Políticas de buckets	As políticas de bucket são um sistema de gerenciamento de acesso usado para definir permissões de acesso em buckets, objetos e conjuntos de objetos. Essa funcionalidade suplementa e, em muitos casos, substitui as listas de controle de acesso. Para obter mais informações, consulte <a href="#">Uso de políticas de bucket e políticas de usuário (p. 326)</a> .	6 de julho de 2010
Sintaxe de caminho disponível em todas as regiões	O Amazon S3 agora oferece suporte para a sintaxe de caminho para qualquer bucket na região clássica dos EUA ou se o bucket estiver na mesma região do endpoint da solicitação. Para obter mais informações, consulte <a href="#">Hospedagem virtual (p. 46)</a> .	9 de junho de 2010
Novo endpoint para UE (Irlanda)	O Amazon S3 agora fornece um endpoint para UE (Irlanda): <a href="http://s3-eu-west-1.amazonaws.com">http://s3-eu-west-1.amazonaws.com</a> .	9 de junho de 2010
Console	Agora é possível usar o Amazon S3 por meio do Console de gerenciamento da AWS. Você pode ler toda a funcionalidade do Amazon S3 no console no Guia do usuário do console do Amazon Simple Storage Service.	9 de junho de 2010

Alteração	Descrição	Data
Redundância reduzida	O Amazon S3 agora permite reduzir seus custos de armazenamento armazenando objetos no Amazon S3 com redundância reduzida. Para obter mais informações, consulte <a href="#">Reduced Redundancy Storage (RRS) (p. 6)</a> .	12 de maio de 2010
Novas regiões suportadas	O Amazon S3 agora oferece suporte para a região Ásia-Pacífico (Cingapura). Para obter mais informações, consulte <a href="#">Buckets e regiões (p. 56)</a> .	28 de abril de 2010
Versionamento de objeto	Essa versão apresenta o versionamento de objeto. Todos os objetos agora podem ter uma chave e uma versão. Se você habilitar o versionamento para um bucket, o Amazon S3 fornecerá a todos os objetos adicionados a um bucket um ID exclusivo de versão. Esse recurso permite que você se recupere de substituições e exclusões indesejadas. Para obter mais informações, consulte <a href="#">Versionamento (p. 8)</a> e <a href="#">Usar versionamento (p. 448)</a> .	8 de fevereiro de 2010
Novas regiões suportadas	O Amazon S3 agora oferece suporte para a região Oeste dos EUA (Norte da Califórnia). O novo endpoint para solicitações para essa região é s3-us-west-1.amazonaws.com. Para obter mais informações, consulte <a href="#">Buckets e regiões (p. 56)</a> .	2 de dezembro de 2009
AWS SDK para .NET	A AWS agora fornece bibliotecas, códigos de exemplo, tutoriais e outros recursos para os desenvolvedores de software que preferem criar aplicativos usando operações de API específicas da linguagem .NET em vez de REST ou SOAP. Essas bibliotecas oferecem as funções básicas (não incluídas nas APIs REST ou SOAP), como autenticação de solicitação, novas tentativas de solicitação e processamento de erros, para que você possa começar a usar com mais facilidade. Para obter mais informações sobre bibliotecas e recursos específicos da linguagem, consulte <a href="#">Usar os AWS SDKs, a CLI e os Explorers (p. 639)</a> .	11 de novembro de 2009

# AWS Glossary

For the latest AWS terminology, see the [AWS Glossary](#) in the AWS General Reference.